UMinho|2018

Rui Manuel Dias Ferreira Lima **Broadcast Cancellation in Unstructured Networks**

Universidade do Minho
Escola de Engenharia

Rui Manuel Dias Ferreira Lima

Broadcast Cancellation in Unstructured Networks

Programa de Doutoramento em Informática (MAP-i)
das Universidades do Minho, de Aveiro e do Porto

Universidade do Minho
universidade de aveiro
U.PORTO

fevereiro de 2018

Universidade do Minho
Escola de Engenharia

Rui Manuel Dias Ferreira Lima

Broadcast Cancellation in Unstructured Networks

Programa de Doutoramento em Informática (MAP-i)
das Universidades do Minho, de Aveiro e do Porto

Universidade do Minho
universidade de aveiro
U.PORTO

Trabalho realizado sob a orientação do
Professor Doutor Carlos Miguel Ferraz Baquero Moreno
e do
Professor Doutor Hugo Alexandre Tavares Miranda

fevereiro de 2018

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, 15 / 02 / 2018

Full Name: Rui Manuel Dias Ferreira de Lima

Signature: ______________________________

À Sílvia
À Leonor e ao Duarte

Agradecimentos

Quero endereçar os meus mais sinceros e especiais agradecimentos aos meus orientadores, Professor Doutor Carlos Miguel Ferraz Baquero Moreno (UM) e Professor Doutor Hugo Alexandre Tavares Miranda (FCUL), pela sua dedicação na orientação deste trabalho, pela persistência e motivação transmitidas durante todas as etapas, pelas videoconferências semanais, e em especial, naquelas alturas em que surgiram as maiores dificuldades, definindo sempre o rumo certo. Os seus sábios conselhos guiaram todo o desenvolvimento do trabalho, incansáveis em sugestões, comentários e, especialmente, na análise dos resultados e nas revisões da escrita. A principal mais-valia deste percurso resultou do facto de partilhar uma relação humana com ambos que ultrapassou o âmbito desta tese, excedendo em muito as minhas expectativas iniciais.

Sendo que a minha família é o motor da minha vida, esta caminhada foi uma partilha constante de todos e a quem eu quero agradecer todo o apoio e compreensão. Ao amor da minha vida, Sílvia, reconheço toda a sua atenção, dedicação, motivação inspiradora e suporte fundamental aos nossos filhos. Tenho a sorte de ter duas luzes que iluminam a minha vida; Leonor e Duarte, obrigado pela vossa paciência e alegria que sempre demonstraram. Nada disto seria possível, e, mesmo sendo, não teria qualquer significado para mim sem o vosso amor constante. Em particular, o da minha Mãe e o do meu Pai, com os quais estarei eternamente em dívida, pela sua entrega sempre incondicional e total. Sinto-me particularmente feliz por continuar a ter nos meus pais os meus maiores amigos de sempre.

Também só foi possível desenvolver este trabalho graças ao espírito de família demonstrado por todos quantos trabalharam comigo no Centro de Informática da Universidade Lusíada em Vila Nova de Famalicão, particularmente nestes

últimos sete anos durante os quais estiveram sempre prontos para assegurar as tarefas e partilhar momentos inesquecíveis. O excelente ambiente de trabalho foi proporcionado pelas extraordinárias pessoas que constituíram esta equipa; Carlos Ribeiro, Élio Costa, Fernando Silva, Ricardo Freitas e Sérgio Esteves, bem como todos os colegas que passaram pelo Centro de Informática da Universidade Lusíada, e que para sempre se mantêm no meu coração.

Adicionalmente, quero ainda agradecer às instituições que apoiaram o trabalho apresentado nesta tese:

- À Universidade Lusíada - Norte, *Campus* de Vila Nova de Famalicão, disponibilizando todas as condições necessárias para o desenvolvimento do trabalho de investigação apresentado nesta tese. A esta instituição tenho dedicado com empenho e paixão todas as minhas tarefas profissionais, constituindo uma segunda família à qual me orgulho de pertencer e onde encontrei pessoas fantásticas (funcionários, estudantes e docentes) que marcaram os 23 anos mais recentes da minha vida.
- Ao INESC e o HASLab - High Assurance Software Lab, pelo apoio logístico e comparticipação nas missões de divulgação científica.
- Ao Departamento de Informática da Universidade do Minho e ao MAP-i Programa Doutoral em Informática das universidades do Minho, Aveiro e Porto.

Braga, fevereiro de 2018
Rui Manuel Lima

Abstract

Technological advances on communications and information systems have led to the development of inexpensive electronic devices, powered by batteries, and with sensing capabilities. Nowadays, small devices are being ubiquitously embedded into the real world environment, with integrated wireless technologies communicating with their nearest neighbours, cooperating in a distributed system and increasing the support for the materialization of the Internet of Things (IoT) concept. To overcome the limitation of sending data beyond each sensor node's radio transmission range, the intermediate nodes work as relays, and broadcast is the main communication component for establishing multi-hop unstructured networks. However, state-of-the-art multi-hop broadcast communications are usually based on flooding mechanisms, that do not cope well with energy issues of battery powered devices. Our approach towards decreasing energy constraints is to improve broadcast efficiency by reducing the occurrence of unnecessary retransmissions.

This dissertation addresses several techniques for stopping an ongoing search by controlling the broadcast propagation, even without prior knowledge of the network topology. Our main contribution, named broadcast cancellation, provides a distributed approach to cancel multi-hop broadcasts. In particular, studying the cancellation issues and discussing solutions using chasing packets, stochastic data structures and gradient approaches for hybrid search mechanisms. The proposed algorithms were evaluated and compared with state-of-the-art search mechanisms, using simulation techniques. Experimental results show that broadcast cancellation outperforms the competitive alternatives, balancing the trade-off among latency and energy efficiency.

Resumo

Os avanços tecnológicos nos sistemas de informação e de comunicações têm impulsionado o desenvolvimento de pequenos dispositivos eletrónicos integrados, de custo reduzido, alimentados por baterias e munidos de vários tipos de sensores. Estes sensores inteligentes estão a ser embutidos de forma ubíqua no mundo real, comunicando através de redes sem fios diretamente com os seus vizinhos mais próximos, concretizando um sistema distribuído e materializando o conceito de *Internet of Things (IoT)*. Para ultrapassar os limites físicos impostos pela propagação do sinal de rádio, os próprios dispositivos funcionam como repetidores *multi-hop*, formando redes não-estruturadas e comunicando por difusão. No entanto, quase todos os mecanismos de difusão são baseados em algoritmos que, de uma forma não controlada, contactam exaustivamente todos os nós da rede, sem as devidas preocupações com os consumos energéticos. A nossa abordagem vem melhorar a eficiência energética dos mecanismos de difusão, reduzindo a ocorrência de retransmissões desnecessárias.

Esta dissertação aborda várias técnicas para travar uma pesquisa em curso, sem conhecimento prévio da topologia de rede, mas controlando o avanço da difusão. A principal contribuição, *"Cancelamento de Difusão"*, é um mecanismo distribuído para interromper a propagação da difusão em redes *multi-hop*. O trabalho investiga e debate soluções para o cancelamento baseadas em mecanismos de perseguição, estruturas de dados estocásticas e estratégias de pesquisas híbridas direcionadas por gradiente. Os algoritmos propostos foram avaliados e comparados com as alternativas mais recentes, recorrendo a técnicas de simulação, mostrando que o mecanismo de cancelamento de difusão é competitivo, equilibrando o balanceamento entre latência e eficiência energética.

Contents

List of Figures

List of Algorithms

List of Tables

Acronyms

ns-2	Network Simulator version 2.35

ABC	Adaptive Broadcast Cancellation
AODV	Ad-hoc On-demand Distance Vector
API	Application Programming Interface
AUSN	Autonomous Undersea System Network

BCIR	Broadcast Cancellation Initiated at Resource
BERS	Blocking Expanding Ring Search
BF	Bloom Filter
BFS	Breadth-First Search

CBF	Counting Bloom Filter
CDMA	Code Division Multiple Access

DSDV	Destination-Sequence Distance-Vector
DSR	Dynamic Source Routing

EDBF	Exponential Decay Bloom Filter
ERS	Expanding Ring Search

FDMA	Frequency Division Multiple Access
FRESH	FResher Encounter SearcH
GPS	Global Positioning System
GSM	Global System for Mobile Communications, *originally Groupe Spécial Mobile*
IDSQ	Information-Driven Sensor Querying
IoT	Internet of Things
IP	Internet Protocol
LAB	Location Aided Broadcast
LAR	Location Aided Routing
LBA	Limited Broadcasting Algorithm
LBF	Linear Bloom Filter
LEACH	Low Energy Adaptive Clustering Hierarchy
LHBA	Limited-Hop Broadcast Algorithm
LTE	Long Term Evolution
MAC	Media Access Control
MANET	Mobile Ad-hoc NETwork
MD5	Message-Digest algorithm 5
NAM	Network AniMator
NOAH	NO Ad-Hoc routing
OTcl	Object oriented extension of Tcl
P2P	Peer-to-Peer

RFID	Radio-Frequency IDentification
RGG	Random Geometric Graph
RSS	Received Signal Strength
SBF	Scalable Bloom Filters
SHA	Secure Hash Algorithm
SpecBF	Spectral Bloom Filter
TDMA	Time Division Multiple Access
TORA	Temporally Ordered Routing Algorithm
TTL	Time-To-Live
UMTS	Universal Mobile Telecommunications System
VANET	Vehicular Ad-hoc NETwork
WBAN	Wireless Body Area Network
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
WUSN	Wireless Underground Sensor Network
ZRP	Zone Routing Protocol

Chapter 1

Introduction and Overview

The intent of this chapter is to familiarize the reader with the thesis research area, and present an overview of the challenges to be addressed. Ongoing research activity related to wireless communications continues to be under intense development, denoting a demand for exploration of multiple unsolved problems. Our research target is to improve the efficiency of broadcast propagation in unstructured wireless networks, by reducing unnecessary retransmissions. Thus, increasing the lifetime of wireless battery powered devices, and expecting to improve a wide range of applications.

Communications technology evolution allowed for a significant increase in the quantity of information, its availability and in the simplicity of accessing it. Nowadays, networks are indispensable for personal relationships, work and leisure [71]. One of the biggest contributions was given by Internet-based communications systems, allowing people to overcome barriers imposed by physical distance, enabling remote access to information with just a simple click. From all the communications systems, mobile communications are becoming a new basic need for communities, which may be so essential as people transportation or utilities (*e.g.,* water, electricity) [173]. Recent years of the information society era were characterized by a variety of applications enabled by wireless technologies [32, 266]. Wireless communications have a strong impact on everyday life, allowing access to several information systems while managing user mobility in the physical world [182].

Recently the International Telecommunication Union (ICU)[1] estimated that

---

[1] `http://www.itu.int`

by the end of 2016 there were more than seven billion[2] global wireless communications subscriptions in mobile cellular networks, a number with an order of magnitude that is roughly 95% of the world population, providing a clear indication of the worldwide acceptance of mobile networks. Mobile phones are the greatest example of widespread use of wireless technologies [36]. However, a vast range of other wireless technologies such as *i*) satellite services, *ii*) radio frequency identification tags, *iii*) monitoring sensor networks, *iv*) bluetooth and *v*) Wi-Fi are also widespread, overpowering the impact of emerging wireless technologies [58].

Technology evolution has produced small size wireless devices that may be embodied into the physical items and support the visionary spirit of Mark Weiser for ubiquitous environments, making communication technology an invisible component in people lives [268]. Sensing devices may detect events or changes in the physical environment and actuators mechanisms may interact with the physical world, bridging both virtual and physical dimensions. A large number of solutions have been developed to interconnect almost all sensors/actuators devices to the Internet, thus making the Internet of Things (IoT) a reality. More than one hundred smart applications are available in the marketplace [208], contributing for a dynamic global information network with critical sense data (using very distinct sources), exposing the collected data available to everyone through the Internet, permitting later analysis and recommendation. However, application development is still one of the main barriers to a wide adoption of Wireless Sensor Networks (WSNs) technologies. Programming mobile wireless sensors in real-world deployments is typically carried out near to the operating system layer, therefore requiring the programmer to focus on low-level system issues [190].

Home automation and industrial monitoring are large application domains boosting the adoption of WSNs [112, 257]. The number of wireless devices available to assist human activities in everyday life is already quite substantial and it is not difficult to predict that it will continue to increase. Ubiquitous environments, using wireless sensor devices require a large amount of data to be propagated and communications are expected to be mainly supported by wireless sensors themselves.

Health sensor-based applications are becoming widespread. Wireless sensors

---

[2] 1 bilion $= 10^9$, as defined on the short scale

technologies have a strong impact in peoples lives, and in spite of some negative effects (*e.g.,* pollution, privacy loss), the positive effects are very significant. An example are the lives saved by providing healthcare diagnoses to distant villagers, where experts at remote locations can monitor patients medical signs and operate electronic healthcare systems. Events in the physical world may be easily detected by sensors, that can process the sensed data, detecting critical variations in its monitored activity and generate a corresponding output (*e.g.,* temperature, heart rate), to be stored locally or transmitted to other devices. As the equipment prices decreases, the demand for applications using remote wireless sensors increases. Wearable small devices with sensor capabilities are very popular gadgets among sportspersons, monitoring their own vital signs and uploading related data to central servers on the Internet [14].

Environmental issues are one of the top priorities all across the world. Gathering environmental data is crucial for detecting some critical indicator deviations, exploring the achieved sensing diversity, and exhibiting great potential for WSNs applications [183]. Monitoring environmental conditions for agriculture is one of the most explored areas by intelligent applications, where the WSNs play a key role for sensing and notification purposes. Applications for monitoring soil properties, water and toxic substances levels may explore the powerful capabilities exhibited by current mobile sensor nodes [233]. Operating wireless sensors below ground in Wireless Underground Sensor Networks (WUSNs) (*e.g.,* underground Mines) reduces the mess of cables often existing in underground sensing solutions based on wired networks. However, wireless communications through a dense medium such as soil or rock is significantly more challenging than through air. Replacing sensor batteries may be a hard task (sometimes impractical), due to difficulties in access (*e.g.,* unearthing). These factors combined in WUSNs, requires that communication protocols be redesigned to preserve energy [4, 124].

Sensor mobility is mainly provided by wireless and battery power technologies. Mobility may be a decisive requirement for several application scenarios. One of those examples are efficient disaster detection systems, that should operate without requiring any pre-existing infrastructure [122]. Among other situations, Mobile Ad-hoc NETworks (MANETs) may provide those crucial communications in self organized networks, offering an alternative to urban communication infrastructures that are expected to be damaged or non-operational [19]. Recently, MANETs have undergone major developments, proving that they are useful to implement effective alerting systems, reporting updated data associated with location information, assisting rescue teams in catastrophic scenarios, thus reducing the loss of life and property [16, 21]. Wireless sensors may act as mobile network nodes, also designated as **motes**. Nodes are small electronic platforms equipped with sensors, data processing (microcontroller), communication capability (radio transceiver), memory and autonomous power source (battery).

Nowadays, sensors have many additional features beyond basic physical detection capability [280]. However, wireless sensor nodes are made of small circuits with strict limitations for processing, storage and networking. The fact that multiple sensors have the ability to transmit data through wireless media is not sufficient to assure communication between them [91]. It should be noted that wireless media are unpredictable, resulting in several challenge issues from multiple adverse events such as *i*) signal attenuation, *ii*) propagation variations, and *iii*) interference; all of which may significantly change the link properties, affecting network coverage and connectivity [5].

Each radio frequency transmission may be received by multiple devices that share the same wireless standard, allowing the exchange of messages between wireless devices [63]. This characteristic of radio transmission may be used by any given devices for broadcasting data to other devices in proximity. Broadcast assumes the role of a main building block to enable communication, especially for neighbour discovering, global data collection, naming, addressing and route discovery [7]. In these scenarios, it is necessary to use wireless nodes as routers (which can relay messages), to extend communications out of the node scope, creating an enlarged communication multi-hop network [118]. A good property of this model is that it may be adapted to frequently changing or unknown network topologies.

As an example, consider a simple search mechanism based on a polling approach, that may be implemented by any given technology, which periodically injects queries to other nodes and waits for replies. Typically, those periodic reporting approaches ignore node conditions (*e.g.,* remaining energy) and lead to uncontrolled message propagation, compromising the system availability. There is a need to research for efficient query mechanisms that also attend to critical node resources and constraints, aiming to extend the system operational lifetime [148].

In many cases, energy consumption by distinct hardware in standby mode is negligible with respect to the energy consumed during the processing operations, and above all, the energy spent by the operation of communication protocols [18]. Experimental measurements have shown that in general the energy cost of transmitting a single bit of data is approximately the same as the one needed for processing a thousand operations in a generic sensor node [215, 225]. Several energy conservation schemes have been surveyed for WSNs exploring strategies that include energy-efficient routing protocols, node retransmission scheduling and topology control [172].

Energy consumed in the sensing subsystem depends on technological advances on low power consumption devices. In general, without considering changes on the hardware, energy saving techniques focus on two subsystems: *i*) the sensing subsystem, *i.e.* reducing the sampling frequency of expensive energy operations, and *ii*) the networking subsystem, *i.e.* reducing the required number of retransmissions (using compression and aggregation techniques for better energy-efficient communication protocols). Those techniques are one of the top research areas in WSNs and MANETs domains, where broadcast is the main communication primitive [9]. There has been a high demand towards research for efficient broadcast algorithms [231]. It should be noted that, without a careful design, energy-efficient protocols may perform much worse than uncontrolled protocols, *i.e.* the applied controls increases the overhead [292].

To highlight the energy issues, consider a simple monitoring application gathering several physical indicators with multiple distinct sensor nodes. To respond for a user need, it requires the location of a thermometer sensor. Generically, the application will trigger a search query, propagated by the nodes using broadcast. Query messages will be retransmitted by all the nodes. However they may be interrupted as soon as one of the thermometers reports a suitable match and before flooding the entire network. The need for a primitive to cancel the initial broadcast becomes clear, because the answer to the question is already known. We focus our research attention to new search mechanisms that may avoid irrelevant broadcast retransmissions, aiming to improve nodes energy consumption.

In face of such a wide range of applications previously presented, there are strong motivations to develop new solutions and improve the performance of wireless systems communication that are already extensively used and which have many unsolved issues [2]. The main motivation to proceed with this thesis research emanate from the practical need to extend wireless systems lifetime. The next section presents problem descriptions exposing some of the main issues shared by several wireless-based applications. The main challenge is to efficiently manage the limited power resources available in wireless nodes, introducing new mechanisms to control broadcast retransmissions. Our main strategy is to propose a new mechanism to limit broadcast, here referred to as cancellation, that collaborates to achieve energy-efficient solutions to problems like those detailed in the following section.

1.1 Problem Description and Statement

Autonomous powered sensors with wireless technologies have the ability to move or be freely moved across the physical world, enabling mobility while keeping communication facilities. No wire connection is needed between the sensors and the external environment, not even for power supply (supported by a battery) or for communication with other sensors (provided by radio frequency). However, direct radio communication between wireless sensors has some limitations such as *i*) distance, *ii*) baud rate, *iii*) moving speed and *iv*) pattern. We are assuming that normal utilization never exceeds those limits, avoiding permanent failures that prevent communications.

Connecting wireless nodes that are not static may be a hard task, mainly due to the strong impact that mobility may cause in network topology [121]. Our study is not focused on mobility. Namely we considered some topology stability during radio propagation, represented by synthetic topologies, that may have resulted from mobility, mimicking the characteristics of realistic scenarios, and increasing the scale of several known problems such as:

Query searching Wireless network topologies are dynamic, nodes and records have the capability to change their location. Thus, applications require regular query searching operations to address the outdated location problem.

Route discovery Considering multiple wireless nodes forming a network, nodes should be used to forward data between source and destination, knowing that routing tables may be quickly outdated.

Data dissemination How to inform each node of any given known property?

Query searching, route discovery, or data dissemination problems have well known solutions in static topologies [8, 37]. However, those solutions are unsuitable for real wireless devices deployments, where those problems are very hard to solve. Traditional solutions were designed for wired networks and cannot be efficiently applied in wireless networks deployments. To deal with the frequently expiration of any collected data, some solutions were developed addressing the above communications problems, and usually rely on broadcasting mechanisms.

In general, **broadcast** characterizes a one-to-many communication principle. Over the air broadcasting is a well known technology for radio and TV media content delivery. Broadcast may also describe the act of transmitting data from a device to other devices. When devices are wirelessly connected, a broadcast occurs whenever a given device starts a radio transmission to be received by wireless devices within the radio range. Multiple devices are considered to be *"neighbours"* when they can communicate directly (no relay) with each other. Our focus on broadcast is due to its importance when the network topology is unknown, and broadcast is the main building block for: *i*) maintaining routing tables [125], *ii*) node discovering [56] and *iii*) reputation management [209]. Moreover, in large scale wireless networks the transmission range of a node is much smaller than the size of the network. The solution to maintain connectivity is to use any given node to retransmit (relay) messages, in order to eventually reach all the nodes, functioning as a multi-hop network (IEEE 802.16j) [274].

The concept of **multi-hop** communication mechanism dates back to ancient Persia (≈ 500 BC), when the emperor strategically placed some of his soldiers on the hilltops, to orally repeat critical messages, achieving a faster information transmission system. The multi-hop mechanism improved early warning systems so much that it became an obvious advantage to achieve success in the battlefield. In 1860, a fast mail service known as *"Pony Express"* was created to deliver messages, newspapers and small packages between the Atlantic and Pacific coasts of the USA, on horseback and using a series of relay stations [237]. Similarly, wireless nodes may collaborate together in a distributed environment,

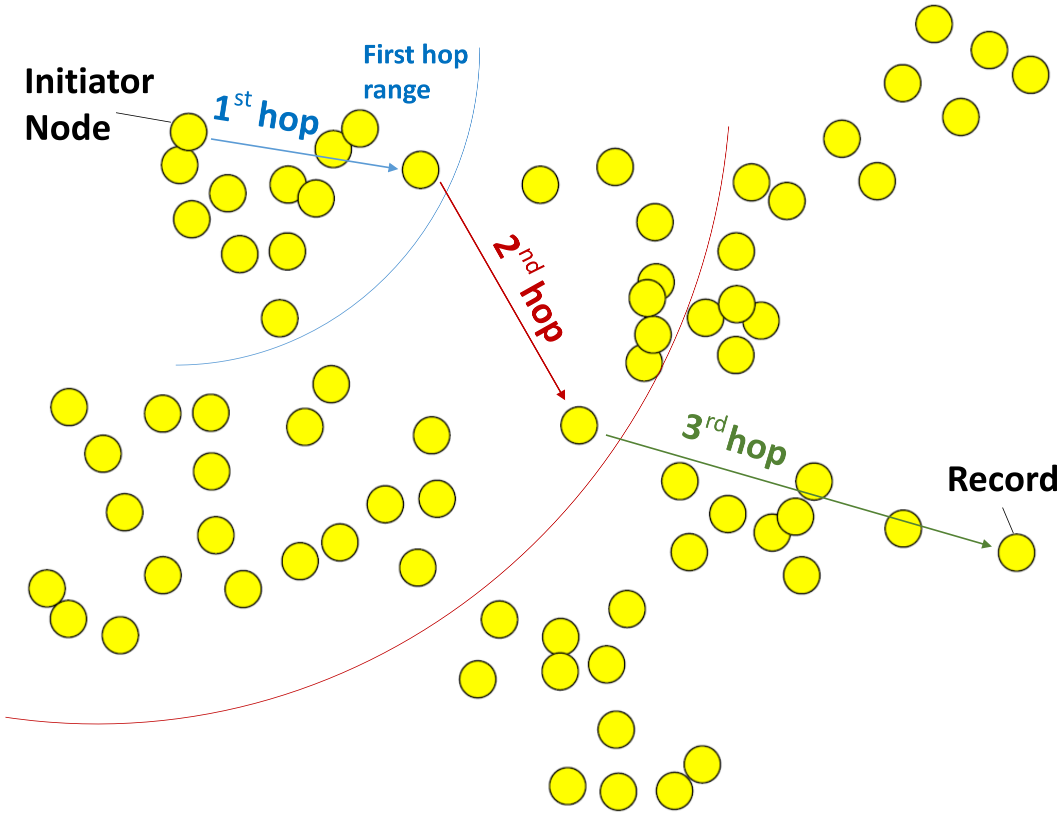


Figure 1.1: Emerged issues when forwarding messages in a multi-hop network.

using multi-hop wireless links among them, to efficiently retransmit messages. In a communication network, the **hop** is a counter attribute of a packet/message that is incremented by any device that retransmits it. Initially, when a new message is created the hop counter is set to `0`.

Figure 1.1 depicts a simple multi-hop scenario with 3 hops between the initiator node (representing the device that initiates communication) and the destination node (representing a device hosting the target record). The signal propagation range depends on several physical properties (*e.g.,* signal power, antenna gain, frequency, channel attenuation and noise, among others) and several nodes may be inside the same hop (imposed by signal propagation range). Assuming an optimal solution, it chooses one node to retransmit the message, coverage is extended with an additional hop and the process is repeated until the message eventually reaches the destination node. In the case depicted in Figure 1.1, after the 3rd hop the message is relayed to its destination.

Our motivation considers applications for Wireless Underground Sensor Networks (WUSNs), where the adverse environmental conditions in underground Mines are very challenging for wireless communications. Several issues arise when wireless sensors are connected via multi-hop links along several miles inside an operating Mine. Underground sensors are hard to access and maintain (*e.g.,* battery replacement). The Mine topologies exhibit long multi-hop routes and also changing in time, as the new tunnels are opened and the older ones are shut down [269].

Inefficiency in upper layers of the network stack has consequences on the lower layers. Instead of addressing the physical and data link layer issues, which depend mostly on the technology, we focus our attention in the efficiency issues of the broadcast communication primitive, which are transversal to wireless standard technologies.

To initiate our research, we consider a Mine scenario monitored by WUSNs. In the Mine, physical items such us shovels, excavators, pneumatic hammers and some other tools are marked with Radio-Frequency IDentification (RFID) tags. The RFIDs tags may be detected by sensor nodes and the corresponding records stored locally in sensor memory. To identify the main issues emerging from this complex system, we focus on a simple question:

> *Where is the Mine excavator?*

Considering that an excavator is lost inside the Mine, searching for that item corresponds to find which network node is hosting the correspondent RFID target record. This is generically addressed as a location problem [278]. Some generic **key issues** were established to focus our research:

Unstructured Multi-hop Networks Having sensors with relay communication capabilities dispenses the existence of a structured network to support communications. When a sensor node receives a query, it may reply with a successfully found message if it detects the object presence nearby. Otherwise, the sensor can retransmit the same query message to eventually reach other sensors, exploring the Mine topology and tracking the object location. The query/answer messages may be forwarded over a multi-hop network to eventually reach their respective destination, but there is no guarantee that messages will reach their target. A devised problem is to select the minimal set of nodes that relay the message and guarantee that it propagates to the entire network.

Broadcast Limitting When a message broadcast is initiated over a multi-hop network, every node will relay (not more than once) the same message. Considering that the queried object is near to the query origin, then after a few relays the object location will be found. Nevertheless, the broadcasting will continue exploring other locations that are irrelevant to the initial query. There is a need for a mechanism to stop the query broadcast process, once the object is found. We expect to propose efficient solutions for these kind of problems.

Topology Change Opening new Mine tunnels generates changes to the network topology, and the communication mechanism should be self-adaptable to new topologies. To successfully find an object location, communications must not rely on historic network information, which rapidly becomes outdated.

Response Delay Some delay may be accepted when monitoring physical quantities such as temperature and humidity, that do not have instant variations. However, query response delay must not be too long, since the location of objects will change as the mining work progresses.

Duplicate Avoidance In a multi-hop network, every node may relay a given message. This generates too many duplicates, creating problems related with battery depletion and bandwidth capability reduction. For node energy conservation it would make sense to avoid unnecessary retransmissions.

Previous problems raise important issues that may be found in similar scenarios where WSNs are also being applied [65]. Several applications are using WSNs in multiple domains, such as:

- Structural Monitoring (*e.g.,* bridges, dikes, towers, railings)
- Traffic Control (*e.g.,* congestion, alternative routes, emergency, speed)
- Health Care (*e.g.,* unobtrusive monitoring, heart attack, epilepsy, hospital, disease prevention, alert systems)

- Agriculture (*e.g.,* climate changes, pests, water resources, nutrient content)
- Monitoring (*e.g.,* environment, volcano activity, earthquake)
- Underground Mining (*e.g.,* location, air quality, explosion avoidance, rescue missions)

Previous applications require the recurrent execution of queries to get updated data, and most of them are flood-based algorithms. The data query is broadcast and eventually retransmitted by every node, which is very inefficient.

1.2 Subject Area: Broadcast Cancellation

Regardless the application scope, the main goal in search and discovery services is to find which network node is hosting the record we are looking for. Assuming a network with several nodes and multiple equivalent copies of the record data, the query mechanism must stop when any of the copies is found. This is a very challenging problem, since topology information is unknown (unstructured networks) before starting the search process.

The **main problem** to be addressed is investigate for alternatives to flood-based search mechanisms. To the extent of our knowledge, no efficient solution for this problem has been found and, in the absence of acceptable alternative solutions, many protocols still rely on broadcast communication mechanisms. In this dissertation the general research question is:

> *Is it possible to improve the performance of existent search mechanisms for unstructured wireless networks based on multi-hop broadcast propagation, by reducing the number of retransmissions, the latency, or a combination of both?*

Our strategy to address the previous question is to investigate for new solutions, taking advantage of the propagation speed, in order to cancel an ongoing broadcast. Different network topologies must be analysed to identify the best cancellation mechanisms for each case. The cancellation cost is proportional to the number of retransmissions until the first record copy is found. The number of retransmissions is one of the main metrics used during the evaluation phase, usually for estimating the communication algorithm energy efficiency.

Broadcast cancellation is the main innovative concept described in this dissertation, serving as a foundation for development of a new set of mechanisms to stop an ongoing broadcast. As described in the previous section, the focus is to limit the broadcast progression, without knowing any network topological information. There are no known clues about how distant other network devices are or even if there are any at all. Broadcast cancellation is inspired on space travel issues that are narrated in several SciFi allegories. Commonly in some of those titles, a given spaceship goes to deep space on a journey that lasts for some generations, to explore the unknown. As the time goes by, technology in the home planet evolved, and a new spaceship is created, which may travel faster than the original one. Supposing that the new spaceship goes in the same direction of the previous one, eventually it will outrun it, and two distinct generations have an encounter. This allegory inspired us to propose a cancellation mechanism for an ongoing broadcast, based on delaying the retransmissions of the initial broadcast [96].

The **cancellation** approach is similar to the strategy used by other energy-efficient search mechanisms, that forward chasing packets for limiting the broadcast propagation [10,11,203,217,219,246]. Details on cancellation are described in Chapters 3 and 5. Before presenting the thesis contributions addressing the subject area previously described, the next section is dedicated to the research methodology that lead our investigation.

1.3 Research Methodology

Computers and computer programs are *"artifacts"* that are man-made and usually not found freely in nature to perform experimental work [46]. However and analogously to traditional sciences, the key to walk step-by-step towards scientific success is to follow the scientific method. A clear research methodology plan is crucial for studying complex systems, such as software *"artifacts"*. One of those examples is the development of new algorithms for wireless networks, where predicting the full system behaviour may be a very hard task. Being aware of these challenges, the adopted research methodology is in accordance with the philosophical bases of the scientific method [70].

We benefit from observations made by others that follow the scientific method approach, when addressing similar problems [86, 125, 211, 220, 241]. The unreliability that characterises participants on WSNs and MANETs networks challenge the new proposed algorithms and distributed applications, to either provide best effort services or to have their state replicated among a large proportion of the nodes [9]. An indication that the stated problems still require further investigation is the amount of research activity in this so popular area, showing that major related issues are far from being solved [203, 204].

In order to save time, space, and funding required for a physical sensor deployment, the new algorithms were evaluated using computer **simulation** tools. Applying simulation techniques is a widely accepted approach in computer science, allowing experiments to be replicated in the same conditions, and independently of when they were executed. Treating each new simulation as a new experiment, studying its behaviour, analysing the obtained results, and conclude about its main characteristics [68]. Similar wireless thematic studies that use simulation techniques usually implement quantitative empiric methods with a strong component of statistical analysis [48,116,125,179,211,211]. Extra care was considered to not use any special model property that may bias the collected results.

Experiments replication may either be: *i*) complete, which presents scalability problems when state is large, or *ii*) partial, which increases the probability of inquire for some missing state component. In any case, node movement and failures may frequently outdate any topological information gathered by the nodes and therefore invalidate any attempt of optimising the state replication or retrieval based on long time cached information.

The first stage of the research methodology ascertains why this is an important research topic, motivated by several application examples that demand improvement of broadcast mechanisms, to increase system availability exploring energy-efficient solutions. The second stage is concerned with terminology issues, clarifying the main concepts (Chapter 2) and pointing the main directions to explore. In the final stages, the followed methodology uses several simulation tools, with distinct granularity levels, to collect data for later analysis and interpretation.

To get a set of **observations** with statistical significance, we repeat the experiments independently and several times. The strategy used to perform the experimental component was performed using two distinct approaches.

During the first approach, we decided to initiate our research by implementing a synchronous prototype simulator from scratch to compare the state-of-the-art algorithms with the new proposed ones, and producing distinct topology scenarios. To generate synthetic network topologies we used the Python NetworkX[3] package. We model wireless networks topologies using Random Geometric Graph (RGG), on which the nodes are associated with active radio stations, and edges reflect the wireless transmission range and, therefore, representing the network connectivity [105]. In this phase, it is generally assumed that all connectivity faults occur at the topology layer of abstraction. The simulator assumes that topology is stable during the query propagation, however for each simulation run a new topology of the same graph model is generated.

In the second approach, we used the Network Simulator version 2.35 (*ns-2*)[4], an event-driven simulation tool based on standardized communications models, with widely reviewed results and well accepted by the scientific community [57, 267]. The main idea is to test, calibrate and observe how the new algorithms operate in several radio communications models. The *ns-2* has a level of detail that allows to observe the lower layers influence on: *i*) signal propagation, *ii*) link quality, *iii*) collisions, *iv*) receiving threshold, among other parameters and thus corresponding to a more realistic testing scenarios.

Hypothesis Our main propose is to research efficient search mechanisms based on broadcast, studying the viability of improving their performance by introducing the cancellation hypothesis. We intent to limit the query progression in a multi-hop network, by setting several scenarios on which an additional delay is imposed at each hop, enabling a time window to provide the opportunity for a successful broadcast cancellation. During cancellation, an explicit message will be propagated as far as the boundary of the flooded nodes (without added delay). The cancellation will eventually stop the initial broadcast, avoiding broadcast to reach unnecessary nodes. It is anticipated that cancellation hypothesis will have an impact in algorithm time response and the number of retransmissions. As such, this has to be characterized and studied. Simulation tools will be used to evaluate the cancellation hypothesis under different wireless network topologies and its performance compared with other competing approaches. In order to plan

---

[3] `https://networkx.github.io`

[4] `http://nsnam.isi.edu/nsnam/index.php/Main_Page`

the research, we identified the main **issues** related with broadcast cancellation mechanism:

- How to stop an ongoing flooding without knowing the network topology?
- When using cancellation based on delayed retransmissions, what is the impact in the message propagation time?
- What is the cost of the cancellation process?
- In what conditions the cancellation approach is more energy-efficient than other solutions?
- Although in some situations WSNs topologies may be synthetically represented by a connected graph, there is no guarantee that the search will find the target record. How adverse are the impact of changes in topology and link properties in the cancellation process?

Several **predictions** may be made, by applying the cancellation hypothesis to a simple network path topology:

- If the record is near the starter node, cancellation will significantly reduce the energy cost.
- Since the cancellation approach will use different packet velocities, an expected delay in record discovery will be present.
- The expected reductions in the number of retransmissions will result in energy-efficient solutions, thus increasing the network sustainability.
- Cancellation has additional costs (retransmissions) that must be considered when comparing it with other search approaches.

Previous predictions suggest the need to balance and adjust the research tasks according to the retrieved results, adapting the progress given by experimentation indications [30]. We address the problems complexity by performing several experiments designed to establish a clear set of boundaries.

Several simulation experiments were repeated to ensure an algorithm performance **evaluation** with stochastic significance, and a proper result analysis with quantitative methods [136, 248]. The assessed parameters are: *i*) the retransmissions overhead, *ii*) packet drop rate, and *iii*) end-to-end time considering any added delay.

One of the most important aspects regarding the **validation** issues, is the fact that it is mandatory that all the results obtained be reproducible. We also consider that is mandatory to implement a prototype for each algorithm describing the purposed protocols, in order to effectively demonstrate the research results [285]. The developed algorithms are evaluated, both in terms of abstract properties, such as *i*) correctness, *ii*) time complexity, *iii*) communication complexity, *iv*) space complexity, and on *v*) how they operate in several distinct scenarios. Suitable tools include complexity theory, graph theory analysis, and the use of network simulators. The validation process is not exclusive based on quantitative methods. It also gives attention to accuracy and to the ability to make generalizations by applying qualitative methods. These two types of methodologies are no longer considered as belonging to different spheres, since they complement each other [248]. The collected statistical data from simulations may also be analysed using the ground theory approach, that matches the most relevant part of the data behaviour collected from the system [175]. The submitted papers contribute to improve the validation strategy, afterwards they were submitted for peer review [202].

1.4 Thesis Contributions

During the thesis research special attention was dedicated to the evaluation process. To compare the performance of the algorithms described in Chapter 3, we implemented them using a self developed simulator, named Synchronous Round-based Simulator (SRS).

The *ns-2* simulation tool was extended to support the performance evaluation of the new algorithms described in Chapter 5, using extensive and more realistic simulations experiments. The experiments conducted with the *ns-2* uses the *ns-allinone* distribution in version v.2.35[5], with additional models developed in `C++` and using Object oriented extension of Tcl (OTcl) for ad-hoc and sensor wireless

---

[5] `http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/`

networks testing scenarios[6]. The evaluation of the new algorithms benefits from *ns-2* built-in and well tested modules, supporting protocols for physical (PHY) and data link (MAC) layers.

The main thesis contributions are a new group of energy-efficient search algorithms, suitable for unstructured networks, based on the broadcast cancellation principle. The main algorithm **contributions** are:

BCIR A new algorithm addressing the broadcast cancellation problem, showing that is possible to reduce the number of retransmitted messages while preserving latency performance, named as **Broadcast Cancellation Initiated on Resource (BCIR)**. To reduce the latency, we also devise an algorithm admitting the cancellation at next hop, namely **BCIR***. However, **BCIR*** has a small negative impact on energy. These algorithms operate without coordination between nodes or prior knowledge of the network topology. We evaluate both algorithms using synthetic random geometric topologies networks and compare the results with existing expanding ring strategies, applied to the same search applications.

LBF A new probabilistic data structure, addressing the problem of setting data membership and assigning it with a numeric value, named as **Linear Bloom Filter (LBF)**. LBF has the scalability and efficiency characteristics exhibit by Bloom filters probabilistic techniques. The LBF accuracy may be calibrated, allowing to quantify an acceptable error, balancing the quantification level with the filter size. Evaluation characterizes the LBF estimator behaviour and compares its performance with deterministic data structures.

ABC A novel hybrid informed search algorithm addressing the broadcast cancellation problem, named as **Adaptive Broadcast Cancellation (ABC)**. The ABC significantly improves latency and simultaneously reduce the number of retransmissions by integrating a distributed learning mechanism. It uses LBFs for managing record membership in the node local vicinity. In ABC record location estimation is used for propagating queries faster towards the likely destination, diminishing the trade-off between energy and time. The ABC propagated messages piggyback's LBF data, taking advantage of the propagation mechanism to disseminate it, without increasing

---

[6] `http://www.di.fc.ul.pt/~hmiranda/pampa/`

the number of retransmitted messages.

This dissertation expresses the research contributions published in the following papers, arranged in chronological order:

i) Comparação de Mecanismos de Cancelamento de Difusão
Rui Lima, Carlos Baquero, e Hugo Miranda
INFORUM - Simpósio Nacional de Informática, Almada, 2012

This paper compares expanding ring mechanisms to control the broadcast propagation. It corresponds to a preliminary approach for implementing the correspondent algorithms. Evaluations uses high level programming and synthetic networks, for comparing the expanding ring with the cancellation approach. The main result is an indication that cancellation may reduce the number of required retransmissions to control the propagation of multi-hop broadcasts.

ii) Stopping ongoing broadcasts in large MANETs
Rui Lima, Carlos Baquero, and Hugo Miranda
European Workshop on AppRoaches to MObiquiTous Resilience (ARMOR) in EDCC,
ACM 10.1145/2222436.2222440, Sibiu - Romania, 2012

This paper compares the impact on the latency and power consumption of four competing approaches for flooding containment. The evaluation uses a self made simulation tool and random geometric graphs to model MANETs synthetic topologies. It compares the stopping algorithms performance in path and tree topologies. The main result is the confirmation that stopping broadcast optimizations may improve flooding implementations and evidence the trade-off between energy and latency.

iii) Broadcast cancellation in search mechanisms
Rui Lima, Carlos Baquero, and Hugo Miranda
28th Annual ACM Symposium on Applied Computing (SAC),
ACM 10.1145/2480362.2480467, Coimbra - Portugal, 2013

This paper introduces a strategy with which the cancellation mechanism is immediately initiated by the nodes where the record is found. The evaluation compares the proposed analytical model with simulation results, using synthetic networks. The main result obtained by Broadcast Cancellation Initiated at Resource (BCIR) algorithm is that it improves energy efficiency without compromising latency. The BCIR* algorithm reduces latency at the expenses of a negative impact in energy.

iv) FBL - Filtro de Bloom Linear
Rui Lima, Carlos Baquero, e Hugo Miranda
INFORUM - Simpósio Nacional de Informática, Covilhã, 2015

This paper presents a new Bloom filter variation that uses floating point values instead of boolean values, assigning a numeric value to each element in a set. The evaluation compares the Filtro de Bloom Linear (FBL) probabilistic response with a deterministic data structure, ensuring an estimator error perfectly bounded. The main result revealed FBL as an efficient memory probabilistic data structure, with high potential applicability for WSNs search algorithms.

v) Adaptive Broadcast Cancellation Query Mechanism for Unstructured Networks
Rui Lima, Carlos Baquero, and Hugo Miranda
Next Generation Mobile Applications, Services and Technologies (NGMAST), IEEE
10.1109/NGMAST.2015.41, Cambridge - UK, 2015

This paper proposes and Adaptive Broadcast Cancellation (ABC), that improves previous works using both BCIR and Linear Bloom Filter (LBF). ABC implements a distributed learning mechanism where knowledge acquired in previous queries guides the propagation towards the expected record location. An extension for *ns-2* network simulation was developed for implementing FLOOD, BCIR, BCIR* and Adaptive Broadcast Cancellation (ABC) algorithms, giving special focus to evaluation using more realistic scenarios. The main result showed that having an intuition of the expected record location may significantly reduce the end-to-end time, while still contributing to a reduction in energy cost for successive discovery queries.

During the research activity an additional work was developed, related with the main research topic.

vi) System Support for Urban Computing - An RFID Caching Hunting Game
Rui Lima
International Journal of Engineering and Industrial Management, 2012

vii) Expanded Social Circles (ESC) - Efficient query on social graphs
Rui Lima
MSKE – Managing Services in the Knowledge Economy, 2013

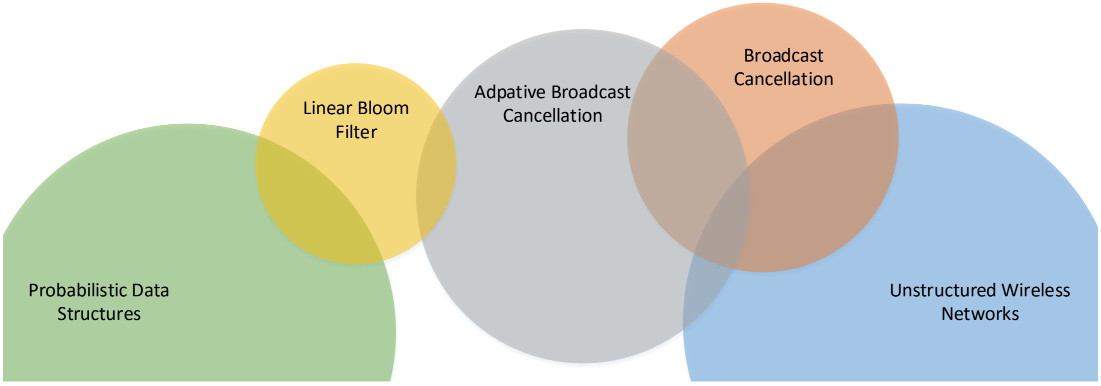


Figure 1.2: Dissertation outline and its affinity with background areas.

Source Code To facilitate the access to source code developed during the several stages of the research we share it using the GitHub[7], which is a web-based hosting service for a distributed version control system. The repository is hosted in `https://github.com/rmlima/BCUN` and the README.md file contains a summary description of the developed projects [158].

1.5 Dissertation Outline

The main background areas supporting this thesis research are: *i*) unstructured wireless networks, and *ii*) probabilistic data structures. They comprise the necessary knowledge background and expose the relationships with others related work.

The core structure of this dissertation comprises three chapters, *i.e.* Chapters 3-5, each one dedicated to the major addressed issues: *i*) proposing an uninformed broadcast cancellation mechanism, *ii*) extending the Bloom filter probabilistic data structure to support intuition on membership sets, and *iii*) an adaptive cancellation mechanism for accelerating the queries propagation towards to destination. Figure 1.2 depicts the dissertation outline framework, representing the affinity between core chapters scope and the background state-of-the-art.

The remainder of this dissertation is divided into five chapters as follows:

- Chapter 2 summarizes the fundamental concepts required to characterize the background research area, including broadcast primitives for unstruc-

---

[7] `https://github.com`

tured networks, record location, search mechanisms, Bloom filters, and early exploratory work (system model, main assumptions, flooding).

- Chapter 3 presents an efficient solution for a broadcast cancellation algorithm (BCIR), addressing the trade-off relationship between energy and time. It includes the algorithm design, implementation and the corresponding evaluation.

- Chapter 4 presents a stochastic data structure (LBF) useful for membership testing and simultaneously control assign a numeric value to stored items. It includes the algorithm design, implementation and corresponding evaluation.

- Chapter 5 presents a broadcast cancellation adaptive solution (ABC), by using an information hybrid approach where the previous queries contribute to accelerate the broadcast propagation towards the expected destination. It includes the algorithm design, implementation and corresponding evaluation.

- Chapter 6 summarizes the main results and discusses the achieved contributions. It presents relevant perspectives for future development foster the current work.

All the written material of this dissertation was compiled with the LaTeX 2_ε system [147]. For reference management we chose the open source application JabRef[8], using the BibTeX file format.

The structure of this document is in accordance with the dissertation writing guidelines as proposed in [31] and uses the Cambridge University[9] PhD/MPhil thesis LaTeX template. The final document format is complying with rules defined by Universidade do Minho (RT-32/2005).

---

[8] `http://jabref.sourceforge.net/`

[9] `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/ThesisStyle/`

Chapter 2

Background

This chapter reviews the background issues for broadcast propagation in unstructured multi-hop wireless networks. The focus is on the presentation of core mechanisms for efficient searching based on broadcast communication, considering energy preservation, a paramount concern when keeping alive battery powered wireless devices.

The remainder of this chapter is organized as follows: first, an overview on wireless technologies, focusing on unstructured networks issues; it summarizes recent work on search mechanisms for multi-hop unstructured networks based on broadcast communications and probabilistic data structures; the related work is discussed and compared using a simplified model, materialised in a Synchronous Round-based Simulator (SRS) [158]; at the end of this chapter, the exploratory work is described and issues emerging from uninformed flooding search experiments are discussed, motivating for further research as described in Chapters 3-5.

2.1 Overview on Wireless Technologies

Recent developments in wireless networking technologies overcome the requirement of a static and rigid network infrastructure (*e.g.,* central base station in a cellular network), to support communication between wireless devices. Some wireless devices may be set to support direct communication between each other by activating infra-structureless modes. One example is the **ad-hoc** communication mode, in which devices can self-organize themselves spontaneously to directly exchange messages or relaying received ones, operating in a distributed architecture, without any rigid infrastructure to support the communications [173].

Our research regards applications that monitor the location of physical items. Most of recent monitoring solutions adopt small wireless sensor nodes that are able to detect and measure external physical quantities [154]. Mobile wireless sensor nodes, also designated as **motes**, add mobility effectiveness to sensor technologies, enabling the implementation of smart cities initiatives [128]. The specifications for the physical layer and media access control for these low-cost, low-speed and low-rate ubiquitous wireless devices are maintained by the IEEE 802.15 working group. Mote examples include the Advanticsys motes (*e.g.,* XM1000, CM5000, CM4000, CM3000), panStamp NRG, Texas Instruments CC430F5137, Atmel ATxMega, Libelium Waspmote, Arduino based motes (NRF24L01), Telosb based motes, Intel motes, MicaZ motes and others [176, 212, 214]. Motes are able to connect with the outside world using a radio link, that may transmit within a typical range of 100m. Power consumption, size and cost are the main barriers to achieving longer distances.

Multiple criteria may be used to classify a computer communication network such as *i*) physical size, *ii*) functional relationship, *iii*) topology, *iv*) data rate, *v*) physical support, *vi*) type of application and *vii*) data transfer method, among others [195]. Although, the literature reveals some classification inconsistency, since the classes overlap. It is very difficult to assign a single class to a given network type, and at the same time maintaining it generic enough to not exclude some variants within the same research scope.

Wireless sensor nodes generate data that requires to be retransmitted wirelessly until it reaches its destination. The actual technological diversity offer a vast number of **wireless communication networks**, which may be categorized in two main classes:

Structured Wireless Networks The main common characteristic shared by structured wireless networks is the existence of powerful specialised hardware, *i.e.* communication infrastructures, to support the network communication and performing all the management activities. Communication between two mobile nodes in structured networks is supported by base stations (usually static) [85]. This model is the basis of popular wireless networks such as the Global System for Mobile Communications, *originally Groupe*

Spécial Mobile (GSM), Wireless Local Area Network (WLAN), and Wireless Mesh Network (WMN) in which base stations are respectively named cells, access points and sink nodes. Typically, their topology is known and it is a consequence of the chosen base stations placement strategy. Base stations are responsible for receiving clients connections and forwarding data to other base stations or gateway the traffic to other independent, *i.e.* third-party, networks (*e.g.,* Internet). Some examples of structured wireless networks are the very successfully mobile telephony systems such as Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX). It should be noted that WLAN are commonly known as Wi-Fi, which is a trademark of Wi-Fi Alliance.

Unstructured Wireless Networks Wireless nodes may self-organize to communicate with each other, extending the direct radio range and creating communication networks, also referred as infrastructure-less, mesh and hybrid networks [6]. Generically, unstructured networks have partial or no access to a supporting infrastructure, resulting in an unknown topology, where nodes are free to move. Some examples of unstructured wireless networks are MANETs and WSNs [280]. Both MANETs and WSNs are a consequence of the short range wireless communications technologies available in small, cheap and computational powerful devices [5].

Focus on discovery service issues, structured wireless networks dictate some organization to the network topology, which is explored by location services to improve the performance of those algorithms. When nodes are self-configuring and organize themselves in unstructured wireless networks, the location services require intensive search operations, due changes in network topology. Nodes cooperate to relaying query messages in a infrastructure-less network, thus expecting copious retransmissions. Given our concerning for unstructured networks issues, additional detail is described in Section 2.2.

Some of the high level protocols that are based on Internet Protocol (IP) may also be used in the WSNs domain, enabling the convergence between structured and unstructured networks [26].

Distinct network classes have distinguishable values for some attributes or properties, which later in this dissertation will be used as measurements indicators, regarding network topology and protocol performance. Following properties are studied for **comparing and characterizing** network topologies.

Connectivity One of the main properties of the network topology is connectivity. Network connectivity is the minimum number of nodes that split a network into two sub-networks. In other words, it is the minimum number of redundant paths between any pair of nodes that share neither the same edge nor a node [131]. It is necessary to have a connected network topology to ensure that the information sent by any given node has the opportunity to reach all the other network nodes within a finite number of hops. Otherwise, the loss of connectivity results in a partition of the initial network, causing connected components to be isolated from each other. Our scenarios consider only connected networks, *i.e.* granting that message failures may result from transient communication issues but not from any forced physical impossibility of connectivity [78].

Degree A metric inspired by graph theory is the degree node property, representing the number of nodes reached in a single node transmission, *i.e.* it is the number of nodes that receive a broadcast message directly. Lower degree nodes usually match to sparse network areas, while high degree nodes correspond to densely connected networks with intense localized traffic [282].

Diameter The maximum shortest path length between any pair of nodes is the network diameter [187], where the path is the number of hops necessary for any given message to reach from one node to another one.

To compare the search (find a given item location) algorithms performance, the following **metrics** are usually considered:

Latency A fundamental time metric is Latency, the time elapsing from the start of message transmission and the reception of the corresponding answer. Latency is measured at the node that initiated the procedure. To clarify this general definition, we measure latency as the time that the source node has to wait to receive a response from the target node, *i.e.* the time between the start of the broadcasting query message by the source node and the time when the source node receives a successful answer to the initial query, which is similar to the **round-trip delay** or **end-to-end** time. Latency is influenced by the network diameter (D). Therefore, latency is proportional to the number of steps required to accomplish the task. In many cases, this will be on the order of the size of the network [20].

Time Efficiency Most of wireless protocols aim to reduce latency [152]. Time efficiency decreases as the time required to obtain a successful answer for a query increases [220].

Retransmission Ratio For the same query message, the retransmission ratio is given by the ratio between the number of nodes that relay the query message (counting all retransmissions until the query process ends, *i.e.* $\#\mathsf{retransmissions}$) and the number of network nodes (N). The retransmission ratio (R) is given by formula: $\mathsf{R} = \frac{\#\mathsf{retransmissions}}{\mathsf{N}}$.

Energy Efficiency The retransmission ratio is an acceptable measure for energy efficiency. Energy-efficient algorithms are focused on minimizing the number of nodes that forward messages [244]. A lower retransmission ratio algorithm is analogous to a more energy-efficient algorithm [92]. Energy efficiency (also designated as overhead) decreases as the number of forwarders increases [113].

Considering the previous network classification (p. 24), details on structured wireless networks are out this thesis scope.

2.2 Unstructured Networks

This dissertation refers to unstructured networks as a collection of autonomous wireless nodes that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Therefore, management tasks must be performed by the nodes. Due to their mobility and scarce resources, the tendency is to replicate these tasks across a large proportion of the network participants.

Unstructured wireless networks may be seen as a particular case of Peer-to-Peer (P2P) unstructured overlay networks, in which topology is influenced by the geographical proximity of the network nodes [3]. This analogy is considering that node neighbours in the overlay network match with physical devices neighbours [167]. An intrinsic restriction for wireless networks is the fact that neighbour devices are established within its radio signal propagation range. P2P overlay networks domain shares multiple characteristics with the unstructured wireless networks domain, after all they are mainly examples of a self-organized distributed system, dealing with efficient routing and exploring optimal search mechanisms using information from nearby peers [129, 144]. Frequently, search mechanisms solutions explore gossiping dissemination algorithms to perform distributed queries while addressing scalability issues [89]. Managing the location of the replicated data may be used to avoid excessive data redundancy and keep search queries subject to a lower latency [180]. Efficient epidemic algorithms aim to spread information to every node without requiring each node to retransmit the initial message, *i.e.* any given node may have the relay function disabled (save energy) and still achieve a complete information dissemination [72, 84, 236]. Unfortunately for P2P networks, MANETs or WSNs, determining the perfect (or at least a good) set of re-transmitters is impossible in run-time as it requires updated topological information that is not available to the participants.

Unstructured networks may be established in different scenarios and are independent of any particular device, transmission technology or protocol. Motivations supporting unstructured networks are given by the expectation that in the near future there will be a high proliferation of small battery powered wireless devices paving the way for distributed applications, where devices self-organize themselves for communication purposes [42, 58]. Examples include environmental monitoring, tracking and personal health care. In addition, these applications create extra value by associating location information to the collected data, allowing some frequently asked questions to be answered, such as the location of physical items, also known as location discovery services [53, 90, 177].

Variety in unstructured wireless networks is no longer confined to MANETs and WSNs. It also includes other networks, such as Wireless Underground Sensor Networks (WUSNs), Vehicular Ad-hoc NETworks (VANETs), Autonomous Undersea System Networks (AUSNs), Wireless Personal Area Networks (WPANs), Wireless Body Area Networks (WBANs) and Wireless Mesh Networks (WMNs). All these types of unstructured networks are assumed to be built with **non het-**

erogeneous nodes, with compatible radio interfaces and comparable transmission ranges, enabling nodes to be operationally indistinguishable. Node diversity contributes to delegate on each node the decision to participate in the relaying process according to its location, local density, node degree or available resources [12, 75, 177, 264].

Some of the most appealing characteristics of **unstructured networks** are the flexibility to adapt to changes imposed by the external environment and the fact that its operation occurs in a distributed fashion, without a single point of failure. However, unstructured networks have an unpredictable topology, formed as wireless devices decide to forward messages and create a multi-hop network. Thus, resulting in frequently topology changes, with several issues associated to connectivity disruptions and network partitioning. Applications that need to keep updated routing tables for message handling rely (in most of the cases) on broadcasting discovery messages. The broadcasting service assumes a major role in the context of unstructured networks. Nevertheless, there are some gathering data applications exploring eavesdropping on unicast transmissions from the surrounding neighbours, instead of using dedicated broadcast messages, thus reducing the cost of broadcast traffic [221]. Extracting information from the gathered sensory data with a specified level of accuracy in a timely and power-efficient approach is still a challenge for monitoring applications [106].

The inclusion of wireless technology in a communication network, also results in various types of **security threats** such as Denial of Service, Attacks on Information in Transit, Sybil Attack, Blackhole/Sinkhole Attack, Hello Flood Attack and Wormhole Attack [206]. Security measures may be adopted to reduce the risk of attack by completely enabling the node anonymity [254]. Furthermore, where privacy-preserving is crucial, stronger protection may be achieved by making packets content unobservable [262]. Although some security schemes aim at reducing the risk of above mentioned threats, the cost-effectiveness and their impact in energy consumption still an ongoing research challenge. Nevertheless, detail on protection and security measures are out of this dissertation scope.

Concerning the main objective to preserve devices battery charge, we choose to focus our research in new efficient broadcasting mechanisms [102]. Next, the dissertation initiates the main issues discussion related to broadcast communication, flooding implementation, and their impact in network lifetime.

Algorithm 1: Flooding

```
begin
    msgList ← {};
    upon event RECEIVE(msg) do
        if msg ∉ msgList then
            PROCESS(msg);                    // Node locally process message
            msgList ← msgList ∪ {msg};
            SEND(msg);                                 // Broadcast message
        end
end
```

Broadcasting The best-effort communication mechanism to deliver a message to every node in a network is broadcasting. It is a core building block for higher level services development, widely used by protocols operating on unstructured networks [56]. Broadcast is a mechanism that circumvents the lack of a central reliable server, by forwarding a given query to every participant, thus ensuring that it will be delivered to a target node. One example that frequently relies on broadcast is a search application. Broadcast primitives may be used to search for records (*e.g.,* a target node, a record of a given user reputation or a device closest to a given location) that may be found somewhere in the network [252, 263]. In some adverse scenarios, broadcast may be the only way to ensure that a message reaches a destination at some unknown location [198, 277].

Flooding A simple and popular broadcast implementation follows the flooding algorithm [104, 125, 138, 211]. It is assumed that nodes are equipped at least with one radio interface and the broadcast service is available. In flooding, each node retransmits any newly received broadcast message (msg). The message is retransmitted to a broadcast destination address, so that there is a high probability that every node within transmission range will receive that message. Algorithm 1 shows a flooding implementation. The flooding overhead is maximum, it imposes N retransmissions to cover an entire network, with N being the number of node participants.

Flooding is simple, but also the most resource demanding algorithm for broadcast [56, 228]. Consider a multi-hop network topology scenario, as depicted in Figure 1.1 (p. 8). To provide a qualitative estimation of the flooding overhead, Figure 2.1 depicts the multiple redundant paths that result when the initiator node starts to search for a given record using the flooding algorithm. As may be

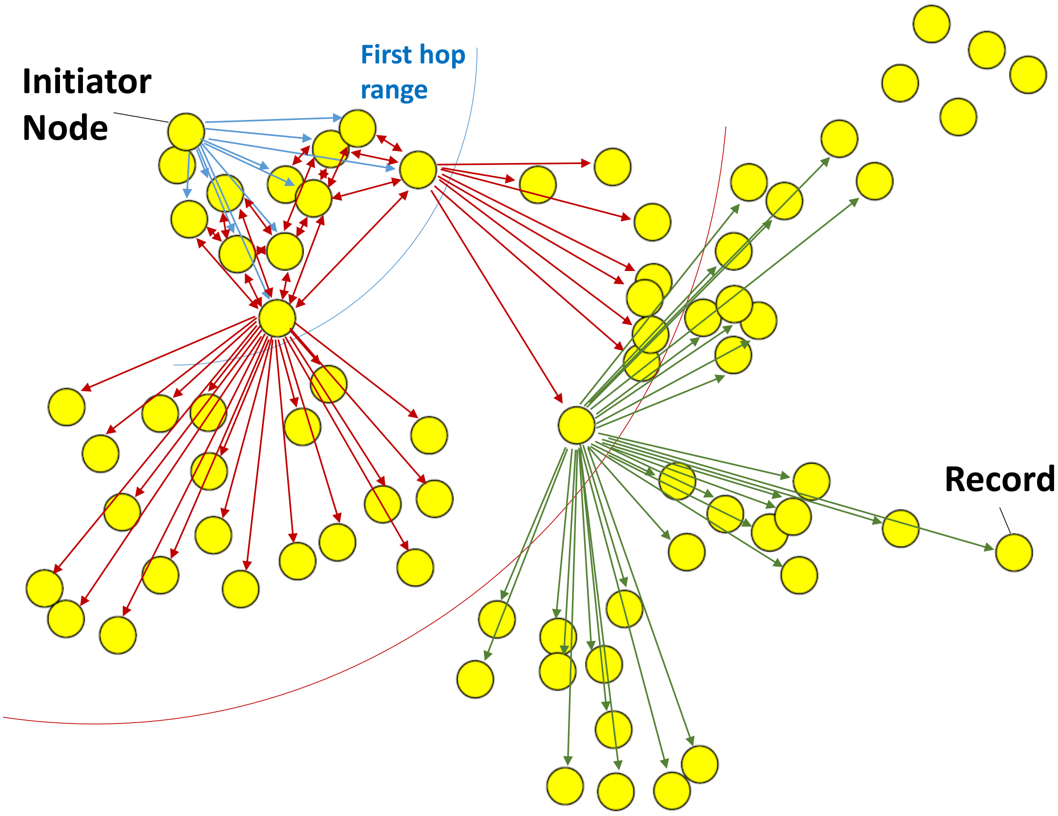


Figure 2.1: Energy inefficiency of flooding due to the existence of multiple redundant paths.

observed in Figure 2.1 using a flooding approach for searching a given record is very energy-inefficient, due to multiple redundant retransmissions that are unnecessary for that search task. Flooding has the advantage of improving communication reliability to signal obstruction by exploring alternatives paths. Nevertheless, a considerable amount of energy is wasted by multiple redundant transmissions.

Radio signal propagation is affected by several factors that contribute to its quality degradation. As an example, radio transmissions propagate over-the-air using a shared channel, creating the opportunity for signal overlapping [250,251]. These adverse factors have a negative impact on propagation of wireless signals with low-power radios, typically used in WSNs. A taxonomy for fundamental concepts of link quality estimation in WSNs and their performance analysis was already surveyed [23]. In situations where it is impossible to maintain continuous connectivity, a delay-tolerant broadcast channel is a competitive alternative to the regulated wireless broadcast channel. Considering a fair delay tolerance, that increments the time window to accommodate node neighbours retransmissions, facilitates communication between nodes [130].

Contention-free Media Access Control (MAC) protocols (attempt to) prevent contention during packet transmission. Access methods allow multiple wireless nodes to share the limited radio bandwidth. Examples of these protocols are: *i*) Frequency Division Multiple Access (FDMA), *ii*) Code Division Multiple Access (CDMA) and *iii*) Time Division Multiple Access (TDMA). FDMA divides the available frequency band into multiple individual channels, giving one channel to each node (*e.g.,* TV Channels Broadcast). With FDMA quickly arise scalability issues, considering the number of nodes used in WSNs applications. CDMA uses a spread spectrum technique, that converts the analog signal to digital and spreads it out over a wider band, that may be shared with other nodes using different codes (*e.g.,* 3G cellular network). TDMA is one of the dominant solutions to schedule transmissions in a fair and efficient manner for wireless multi-hop networks [238]. In TDMA, time is divided into frames. The number of time slots in each TDMA frame is called the frame length. A time slot has a unit time length required for a packet to be transmitted between adjacent nodes.

Despite previously described contention preventing techniques, when nodes are in close range, collisions may occur in case of simultaneous transmissions. Research for efficient MAC protocols for unstructured wireless networks is still under development, concerning energy conservation and providing good throughput [114, 291].

Broadcasting by flooding is usually very costly in terms of energy consumption by wireless devices. Several studies show that different approaches or algorithms that require broadcast primitives affect significantly the wireless devices energy consumption [40, 224]. In flooding, nodes retransmit any received new message which is very energy-inefficient, due to unnecessary retransmissions caused by the following events:

Redundancy When a message is retransmitted and it is already known by all the receiving nodes, it corresponds to a redundant retransmission, that contributes to reduce the algorithm energy efficiency. However, redundancy is useful to increase communication reliability and data accuracy.

Contention In a multi-hop network, nodes may retransmit any received message. Considering that the receiving nodes are near each other, some will compete for access to the communication channel. The access to the channel is prioritized by setting the inter-frame time. After waiting the inter-frame time, a node randomly selects a backoff time and decrements it by one. If another node begins transmitting before its timer has reached zero, the node defers access until the medium is available again, at which time it continues decrementing the timer from where it previously left off. Once the timer reaches zero, the node is allowed to transmit the frame. The timer value must be within the **contention window** values, defined for some traffic priority. Flooding will be responsible for intense channel access competition, leading to a large contention at the beginning of each time slot.

Collisions Some MAC protocols (of which IEEE 802.11 WLAN standard is a remarkable example) avoid certain collisions, imposing a transmission delay schedule. However, collisions are inevitable, such as those resulting from the hidden terminal effect, where the collisions occur near the receiver which may listen sources that are not in range of each other. When nodes transmit at the same time, and a collision occurs, no acknowledgement of the frame will be received. Next, those nodes will increment its retry counter and increase its contention window, following a binary exponential backoff algorithm, and up to a maximum contention window size.

Adverse communication scenarios appear when previously described events succeed each other, which often occurs during flooding algorithms. The combination of the previous events is also known as ***"broadcast storm"*** [256], leading the devices to waste battery in redundant retransmissions and delay messages delivery, due to contention management and collisions. The *"broadcast storm"* events may be very intense when wireless devices share dense radio areas and with multiple overlapping signals which bring communications almost impossible. The most common strategies available to reduce the impact of *"broadcast storm"* events explore mechanisms which constrain retransmissions to a fraction of network nodes, such as:

Probabilistic The decision to relay the query is based on a probabilistic (p) function locally computed. When all the nodes compute a $\mathsf{p} = 1$ value, it degrades to pure flooding [43, 59, 72, 90, 155].

Counter-based A counter is used for tracking the number of times a broadcast message is received at each node. In counter-based algorithms, whenever the counter is greater than the assigned threshold, rebroadcast is inhibited [7, 184, 276].

Distance-based A counter represents distance (hop) between nodes, and a distance-based algorithm will decide whether to relay the received message, based on a pre-determined hop threshold [210, 211, 289]. The threshold may be dynamically adapted to the network (*e.g.,* locally estimation of node density, received signal strength) [74, 156]. Typically in multi-hop networks, it is desirable that the next retransmitting node will be the most distant one, thus minimizing the number of retransmissions.

Location-based It is possible to estimate the covered area by each node when location is known, using a GPS transceiver or monitoring the received signal strength. The decision to relay the received message will be made only when the additional covered area is above a predefined threshold [119, 138, 196, 213, 227, 253]

Cluster-based The cluster-based extends the location-based approach using graph modelling techniques, and assuming that nodes send messages to advertise their presence. The exchanged messages are processed by each node to estimate its connectivity and to elect a cluster gateway, that will be responsible for message relaying operations [17, 38, 64, 97, 110, 143, 145, 232, 261].

Random Walk The simple process for visiting nodes in a graph G, is following some sequential random order. In random walk, point-to-point messages are relayed by randomly selected neighbours for a limited number of hops. In guided random walks, nodes exchange information about local resources to bias the random walk path according to a given relevant property (*e.g.,* remain battery, signal strength, time, location) [20, 22, 144, 168, 273, 283].

Epidemic All the nodes work in a collaborative process, where each node forwards the received messages for the first time, to a subset of the remaining nodes, mimicking the process of an epidemic spreading over a population [66, 123, 134, 151, 187].

One of the greatest disadvantages of using the previous strategies, is the potential decrease in the broadcast resilience, resulting from using less nodes to relay messages. Reducing the redundancy has a negative impact on the ability to face threats and faults, making more difficult to maintain the broadcast service availability.

From the perspective of networks made up of battery powered devices, all the previous strategies focus on saving energy and always include a technique to **avoid duplication** [170]. Duplication may occur when a node receives multiple copies of the same message, as each node receives messages independently, *i.e.* without being aware of other nodes relaying decisions. To allow discarding duplicate messages, each node may track all the listened messages [198]. A garbage-collect algorithm may purge obsolete entries, avoiding memory problems such as a growing history of message identifiers.

Flooding, as described in Algorithm 1 (p. 30) may be used by search mechanisms. However, its energy inefficiency makes it unsuitable for unstructured wireless networks [191, 217]. The main **flooding deficiencies** are:

Overlap / Implosion Considering that i nodes share the same observing region, all of them may decide to forward the message concurrently. As a result, neighbour nodes may receive duplicate messages. As an example, if node $\mathsf{node(A)}$ has neighbour nodes that are also neighbours of node $\mathsf{node(B)}$, the node $\mathsf{node(B)}$ receives multiple copies of the message sent by node $\mathsf{node(A)}$.

Resource Blindness The flooding algorithm does not take into account the available node resources (*e.g.,* energy, memory, cpu). This blindness forwarding typically drain energy of critical nodes which may severely disrupt the multi-hop network connectivity.

Despite flooding confirmed weaknesses and due to its simplicity, flooding continues to be used for discovering the available devices in a network, and reveal how devices are connected to each other. However, efficient search mechanisms should be used in MANETs [37] and WSNs [201].

Many research efforts are being dedicated to **routing** protocols for unstructured wireless networks, which may be categorized as:

Proactive / Table Driven Routing tables are updated through control packets in a proactive fashion [258]. Updates respond to changes in the network topology (*e.g.,* Destination-Sequence Distance-Vector (DSDV) [210], FResher Encounter SearcH (FRESH) [73]). Proactive protocols are periodically discovering routes between nodes, even when nodes suspend their communications.

Reactive / On-demand The route is established when needed (on-demand), by broadcasting route request messages through the network. A route discovery mechanism is reactive when a source wants to send a packet to the destination (*e.g.,* Dynamic Source Routing (DSR) [125], Ad-hoc On-demand Distance Vector (AODV) [211], Temporally Ordered Routing Algorithm (TORA) [205] and Location Aided Routing (LAR) [138]). Reactive protocols applied in MANETs and WSNs result in less communication overhead, although with an impact on latency, caused by the time consumed in route discovery before establishing the communication.

Hybrid Combines the advantages of proactive and reactive algorithms. The routing is initially established with some proactively prospected routes and then routing tables updated by new activated nodes through reactive broadcasting (*e.g.,* Zone Routing Protocol (ZRP) [104]).

Hierarchical / Cluster-based Depending on the nodes hierarchic level, they can perform distinct operations. The hierarchical protocols may improve energy efficiency, by electing the higher energy nodes to process and relay messages, while low energy nodes only perform sensing operations. (*e.g.,* Low Energy Adaptive Clustering Hierarchy (LEACH) [110]).

In what concerns to energy consumption, the reactive protocols are generically considered more energy-efficient than proactive ones. However, they operate without explicit energy concerns [37]. Hybrid protocols benefit from both approaches. Cluster-based protocols balance energy consumption and traffic load, expecting to improve the network lifetime.

Next section will discuss the main strategies used in cross layer search mechanisms, with focus on those that are based on the broadcast communication primitive [199, 265, 294].

2.3 Search Mechanisms

From the multiple services available in a wireless unstructured network, we focus our attention on search mechanism primitives, which are a location service key component, and also one of the most demanding tasks for WSN applications.

IoT presents a challenge in terms of search mechanisms, because devices and users need to find data gathered by wireless sensor nodes. IoT brings physical items into the Internet and search mechanisms require efficient solutions to discover an answer for a given query. Tagging physical items with RFID is one of the most popular solutions where physical items are mapped to locally stored records [208]. From herein the virtual representation in memory of a physical item will be designated as record. Thus, considering the underground Mine scenario, searching for any given physical item (*e.g.,* shovels, excavators, pneumatic hammers, etc) is the same as searching for any given node hosting a target **record**, *i.e.* a tuple like <TIME, NODE_ID, RFID, LOCAL>, assuming that the node location is close to the physical item location.

Generically, the main problem that search mechanisms for unstructured networks have to address is to find whether a given node, node(i), hod a target record $\mathsf{r_j}$ copy, requested by a particular query started by the initiator node, *i.e.* node(0). Considering that query matching answers are routed towards node(0) once instances of $\mathsf{r_j}$ are found, and that at least one instance is available in the network. The query search is complete when node(0) receives the first match, and thus may report it to a higher layer or an end user.

The location service must be aware of battery constraints of the devices, using energy-efficient distributed search mechanisms, wherefore any node may search the network looking for record whereabouts. The node resource constraints (*e.g.,* energy, memory, communication) promote the development of search mechanisms that improve critical resource usage by limiting non productive query message diffusion [204]. As an example, when searching for a given record, *i.e.* a physical item, there is no gain in continuing the query dissemination once the corresponding record that matches the query is found.

For motivation sake, we consider a WSN and a location-based application for tracking tagged item positions and using multi-hop communication across several miles. In such a setting, a query that searches the location of a given item should be interrupted once the object is found, instead of flooding the query through all the nodes in the network.

Search mechanisms may be classified as: *i*) uninformed, without using any particular domain knowledge, and *ii*) informed, using at least one heuristic (additional source of information) guiding the search process.

2.3.1 Uninformed Searching

Uninformed search mechanisms are also known as *"blind"* search mechanisms. Nodes decision to relay the query message is completely autonomous, without using any heuristic or considering any neighbour information. Uninformed searching uses only information available in the problem definition, such as *i*) initiator node (start state), *ii*) relaying nodes (generically, every node from the connected component), and *iii*) record to find (goal test). Most common uninformed search mechanisms are:

Flooding-Unlimited A search mechanism implemented with the well-known algorithm based in Breadth-First Search (BFS) [186], is usually designated as flooding-unlimited. The BFS explores all neighbouring nodes, traversing the corresponding topological graph, until all nodes have been searched. Flooding-unlimited is also designated as **flooding**, **classic-flooding** or **simple-flooding**.

Flooding-Limited Considering that saving energy may rely on limiting the flooding expansion, several solutions explore the flooding-limited for alternative energy-efficient approaches which are inspired in the Iterative Deepening Depth-First Search (IDDFS) [139] algorithm. The IDDFS is a depth-limited version of the well-known algorithm Depth-First Search (DFS). The IDDFS iteratively increase hierarchical depth limits until it finds its goal. Several solutions control the flooding expansion by estimating the distance (hop) to the initiator node, usually by decrementing a counter or the Time-To-Live (TTL) field, until stopping the search upon reaching a given limit [29, 48, 56, 109, 125, 240].

Chasing Packets To stop an expanding broadcast, some search mechanisms use additional fast forward control messages designated as chasing packets, that chase the initial query messages and block the ongoing search [10, 160, 203, 219].

Random Several solutions adopt stochastic approaches, that randomly choose nodes to forward the search query messages [235]. Random approaches include the following search mechanisms:

Random Walk The query is forward only by a limited number of nodes, thus following a random walk. It only generates a fixed amount of query messages at each hop [20].

Probabilistic Forwarding Each node generates a random number. The probabilistic forwarding approach only rebroadcasts the query message when the random number exceeds a predefined threshold [55, 184, 289].

Probabilistic Flooding The search query is forwarded only by a percentage ($\mathsf{p} < 1$) of the node neighbours that are probabilistically selected. When the rebroadcasting of a message is decided using an artificial intelligence classification scheme, the probabilistic flooding is also known as **machine-learning** [43, 67].

Gossip-based Using an epidemic mechanism (similar to a biological virus spreading), a gossip-based search mechanism elects any given node neighbours to forward the query, thus reducing the number of retransmissions [151, 272].

2.3.2 Informed Searching

Informed search mechanisms use heuristics for guiding a query propagation towards its predicted location. Heuristics do not guarantee that an optimal solution will be found. Instead, heuristics find an acceptable solution with smaller effort, when compared with uninformed-flooding approaches. However, nodes are required to provide a mechanism to acquire some knowledge from its neighbours or from previous searches, and use that to compute an index for quantifying the heuristic representation. Most common heuristics supporting informed search mechanisms are:

Distance Nodes compute the expected shortest distance, *i.e.* the minimum hops, to searched record [156, 210, 256].

Received Signal Strength (RSS) Sensors detect the strength level of the received signal for estimating radio propagation properties to the nearby neighbours, relaying only those transmissions under a minimal strength threshold [49, 245].

Location Aware The informed search mechanism assumes that each individual node is aware of the location of every other node, optimizing the searching for a given heuristic related with location [138,253]. The location heuristic is also known as **geographical** or **context-aware** [92,93,94,126,200,244,290].

Clustering Aggregation techniques may be used to reduce the number of retransmissions by establishing clusters of nodes and electing a gateway to communicate with other clusters. Also known as **hierarchical**, when the cluster gateway election uses hierarchical levels of significance based on network topology or group density [38, 64, 110, 232, 261].

Power-Aware Energy is a limited resource and power-aware search mechanisms are designed to preserve battery power of critical nodes [17,50,113,132,171, 179,193,271].

Gradient Nodes are able to predict the expected best direction towards the target record, similarly to a physical gradient concept [22, 100, 279].

Self-Sensing Nodes sense link state information (*e.g.,* RSS) collecting complementary data provided by their self-sensing capabilities (*e.g.,* distance, location), and use it for reducing the area to explore [73, 281].

Swarm-based Nodes mimics the role of pheromones in nature. Swarm-based approaches usually considers messages as *"artificial ants"*, being forwarded stochastically in a multi-hop basis, thus travelling in search for target destination [230].

Adaptive Nodes collaboratively tune the rate of their retransmissions, adapting the searching progression to the network dynamics, and usually capable to start the exploration without any domain knowledge [115, 161].

Distinct heuristics may be combined in a single search mechanism aiming to achieve better performance, such as in Low Energy Adaptive Clustering Hierarchy (LEACH) that is a self-sensing, power-aware, adaptive clustering protocol for structured WSNs [72, 110].

The most common strategy explored by search mechanisms for unstructured multi-hop networks, collects information given by nearby nodes, instead of storing a full state information from a changing topology that will be outdated and useless in future [113]. Other approaches improved the informed search mechanisms by exploring 1-hop or n-hop (same as n-Tier) relative node information, extending the range of data considered in the neighbourhood and associate it with the above heuristics [28, 91, 281].

Despite the available alternatives for pure flooding search mechanisms listed above, the Chapters 3 and 5 will present new efficient search algorithms, exploring strategies such as cancellation and adaptive gradient-based approaches.

2.4 Bloom Filters

Previous section explains why search mechanisms for WSNs usually rely on message broadcast to share knowledge (*e.g.,* distance, location, records) between nodes [73, 94, 100, 113, 179]. In large WSNs, data state information grows exponentially, making it almost impossible to store in scarce limited memory devices, such as sensor nodes. One scalable approach is to accept some data loss, and use probabilistic techniques inspired by the Bloom Filter (BF) data structure [33].

The BF has a restricted size, thus contributing to control the size of the broadcast messages and reducing the communication overhead, especially in multi-hop networks. One example is the new informed search mechanism described in Chapter 5 that uses a BF variation to reduce query average latency, expecting that knowledge from previous queries will improve the performance of new ones. These were the main reasons to dedicate this section to overview the BFs techniques, and in-line with the *Bloom Filter principle* [41]:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 2.2: Empty Bloom filter representation with $\mathsf{m} = 16$ cells.

> *"Whenever a list or set is used, and space is at a premium, consider using a* ***Bloom Filter (BF)****, if the effect of false positives may be mitigated."*

Bloom Filter A BF is a probabilistic data structure designed to cope with large amounts of data, by efficiently managing the available memory space. BF allows testing whether a given element is a member of a set. Informally, a set is a collection of data instances. BF and its variants have been widely used in multiple domains [34, 44, 87, 95, 100, 222, 255]. One of the first use of BF in the search domain happened during an efficient representation of dictionary words [33].

For describing the BF principles, consider a set of n elements, represented by $\mathsf{S} = \{\mathsf{elem}_1, \mathsf{elem}_2, \cdots, \mathsf{elem}_\mathsf{n}\}$. A BF allows to efficiently test if an element elem_X belongs to a set S. However, due its probabilistic nature, the answer assumes the expenses of a given false positive probability.

Bloom Filter Implementation The BF main parameters notation are: *i*) m - number of cells, *ii*) n - number of elements in set, and *iii*) k - number of independent $\mathsf{hash(x)}$ functions. Considering a BF with m cells, the simplest implementation for a BF data structure may be given by a size m bit array. An example of a BF representation is depicted in Figure 2.2, showing that each filter cell contains a single bit. BF initialization corresponds to setting all the cell bits to zero. Two core operations may be performed with a BF: *i*) insert, add an element to the filter, and *ii*) query, test the filter to search if any given element is in set.

The **insert** operation of an element into the BF, corresponds to set k cells to $\mathsf{1}$. The k cells should be deterministically set from the element, although uniformly distributed in the interval $[\mathsf{0}, \mathsf{m} - \mathsf{1}]$. The general case applies k independent $\mathsf{hash(x)}$ functions to the element to be inserted, mapping the $\mathsf{elem}_\mathsf{n} \in \mathsf{S}$ into k array indexes. Some of the most used hash algorithms are Message-Digest algorithm 5 (MD5) or Secure Hash Algorithm (SHA), which have a compliant uniformly distributed behaviour [111].

Considering an example with $\mathsf{n} = 2$ elements (*e.g.,* $\mathsf{elem_A}$, $\mathsf{elem_B}$) and $\mathsf{k} = 3$ independent $\mathsf{hash(x)}$ functions (*e.g.,* $\mathsf{hash_1(x)}$, $\mathsf{hash_2(x)}$, $\mathsf{hash_3(x)}$). The $\mathsf{hash_k(x)}$ functions must necessarily return an index of the array representing the BF, which in this example has $\mathsf{m} = 16$ cell positions.

In the example depicted in Figure 2.3 we use the MD5 cryptographic algorithm and considered the first 5 bits from $\mathsf{hash_1(x)}$, $\mathsf{hash_2(x)}$, and the first 6 bits from $\mathsf{hash_3(x)}$, granting an $\mathsf{m} = 16$ cell BF representation.

$$\begin{array}{l}\mathsf{hash_1(elem_A)} \;\mid\; \mathsf{hash_2(elem_A)} \;\mid\; \mathsf{hash_3(elem_A)} = \mathsf{0001010000000100}\\ \mathsf{hash_1(elem_B)} \;\mid\; \mathsf{hash_2(elem_B)} \;\mid\; \mathsf{hash_3(elem_B)} = \mathsf{0000010001000010}\\ \mathsf{Addition(OR)} = \mathsf{0001010001000110}\end{array}$$

Figure 2.3: Example of $\mathsf{hash_k(x)}$ function output, for elements $\mathsf{elem_A}$ and $\mathsf{elem_B}$.

To insert elements to the filter we use the binary disjunction operator $\mathsf{OR()}$ (addition) and the corresponding array indexes are set to 1. In the previous example the addiction result is depicted in Figure 2.3. Depending on the BF parameters $(\mathsf{m}, \mathsf{n}, \mathsf{k})$ some overlapping indexation may occur. Figure 2.4 depicts the BF array, after inserting $\mathsf{elem_A}$ and $\mathsf{elem_B}$. For each element the corresponding indexes are set to 1. In this case an overlapping occurs for $\mathsf{elem_A}$ and $\mathsf{elem_B}$ at the index 5 position.

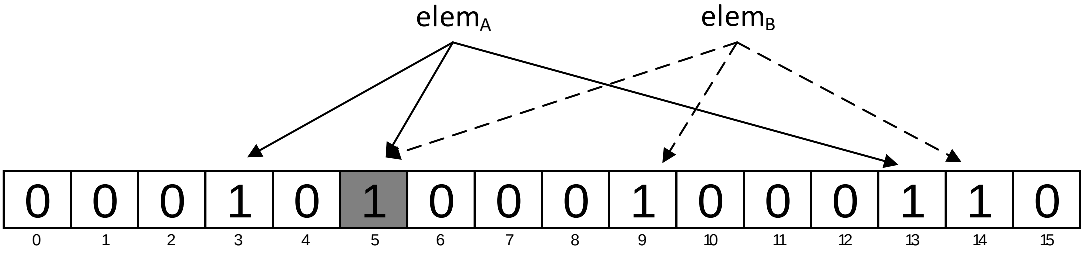


Figure 2.4: Inserting $\mathsf{elem_A}$ and $\mathsf{elem_B}$ into a set S, using a Bloom filter with ($\mathsf{n} = 2$, $\mathsf{k} = 3$, $\mathsf{m} = 16$).

The **query** operation tests if any given $\mathsf{elem_X}$ is a set member. A BF query inspects the content of k array indexes, obtained by $\mathsf{hash(x)}$ functions applied to the query input. In the example depicted in Figure 2.5, querying for $\mathsf{elem_X}$ presence in S, corresponds to check if all the k array cells are set to 1. The $\mathsf{hash(x)}$ functions output for $\mathsf{elem_X}$ is:

$$\mathsf{hash_1(elem_X)} \;\mid\; \mathsf{hash_2(elem_X)} \;\mid\; \mathsf{hash_3(elem_X)} = \mathsf{0001000001001000}$$

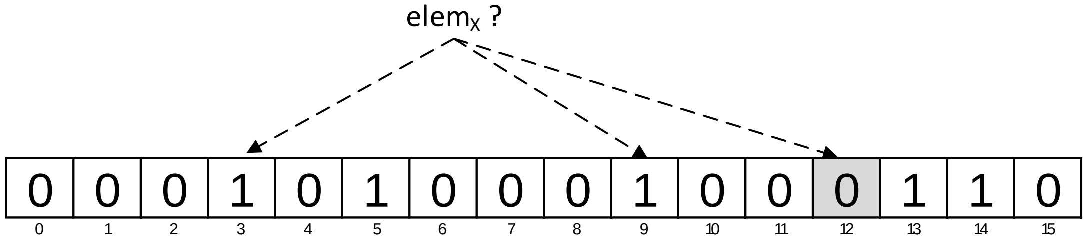


Figure 2.5: Query for $\mathsf{elem_X}$ membership in set S, using the Bloom filter in the conditions of Figure 2.4.

In this case (Figure 2.5), index $\mathsf{12}$ is zero, which is sufficient to obtain a negative response from the filter, *i.e.* $\mathsf{elem_X}$ is not in the set.

Assuming that a BF representation always consider a given data set correctly, the BF does not allow **false negative** responses, *i.e.* a negative query response indicating that an element is not in the set then it surely is not in the set. However, the BF probabilistic nature allows **false positives** responses, *i.e.* the query may wrongly indicate that an element is part of the set.

Assuming that $\mathsf{elem_Y}$ is not in set, Figure 2.6 illustrates a false positive event, resulting from query for a non inserted element, in this example $\mathsf{elem_Y}$. The $\mathsf{hash(x)}$ functions output for $\mathsf{elem_Y}$ is:

$$\mathsf{hash_1(elem_Y)} \;\mid\; \mathsf{hash_2(elem_Y)} \;\mid\; \mathsf{hash_3(elem_Y)} = \mathsf{0001000001000100}$$

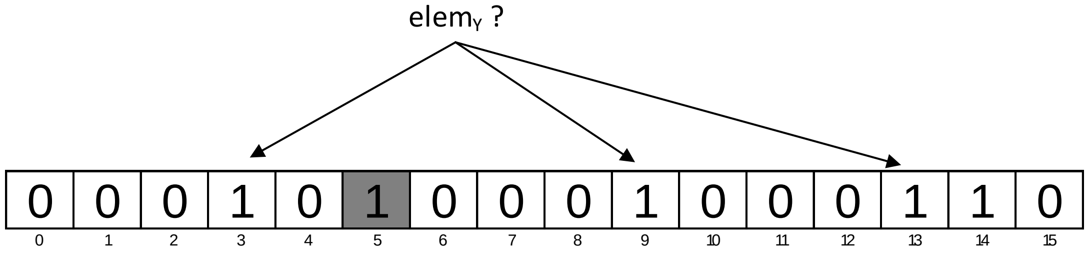


Figure 2.6: A false positive response, when query for a non inserted element $\mathsf{elem_Y}$.

BF false positive responses occurs when all the indexes match with indexes of another element inserted in the filter, *i.e.* all k array cells are set to $\mathsf{1}$.

Bloom Filter Response The accuracy of a BF response depends on: *i)* m, *ii)* k and *iii)* n. Assuming an uniform distribution, the probability to set one bit to $\mathsf{1}$ from a universe of m possibilities is $\mathsf{q} = \frac{1}{\mathsf{m}}$. The probability of having a $\mathsf{0}$ bit value pointed by some array index is the complementary probability, *i.e.* $1 - \mathsf{q} = 1 - \frac{1}{\mathsf{m}}$. Assuming that k $\mathsf{hash}(\mathsf{x})$ functions are independent, the probability of having all the k filter cells set with a $\mathsf{0}$ value is given by $(1 - \mathsf{q})^{\mathsf{k}}$. After inserting n independent elements, the probability of a given bit continues to be $\mathsf{0}$ is $(1 - \mathsf{q})^{\mathsf{n} \cdot \mathsf{k}}$. Combinatorial analysis provides an analytical approximation for p, where p is the probability of any given bit being $\mathsf{1}$, is given by Eq.(2.1) [255]:

$$\mathsf{p} = \left(1 - (1 - 1/\mathsf{m})^{\mathsf{n} \cdot \mathsf{k}}\right) \tag{2.1}$$

The Eq.(2.1) may be simplified and the probability of any given bit being $\mathsf{1}$ may be estimated by Eq.(2.2):

$$\mathsf{p} \approx \left(1 - \mathsf{e}^{-\mathsf{n} \cdot \mathsf{k}/\mathsf{m}}\right) \tag{2.2}$$

Reproduced from [255] Eq.(2.3), analytically expresses the number of $\mathsf{hash}(\mathsf{x})$ functions that minimize the false positive probability, *i.e.* minimize Eq.(2.2). For practical implementation reasons, k should be an integer (representing array positions). In Eq.(2.3) the ceiling operator $\lceil\ \rceil$ represents the smallest integer greater or equal to k.

$$\mathsf{k} = \left\lceil \frac{\mathsf{m}}{\mathsf{n}} \ln 2 \right\rceil \tag{2.3}$$

Considering the optimal k from Eq.(2.3), the probability of false positives f is given by Eq.(2.4):

$$\mathsf{f} \approx \left(\frac{1}{2}\right)^{\mathsf{k}} = \left(\left(\frac{1}{2}\right)^{\ln 2}\right)^{\mathsf{m}/\mathsf{n}} \tag{2.4}$$

The Eq.(2.4) predicts how the **calibration** parameters $\{\mathsf{m}, \mathsf{n}, \mathsf{k}\}$, interfere with BF false positive responses, and which may be resumed in the following practical rules [41]:

- False positive probability increases as more elements are added to BF.
- False positive probability decreases as the size of the BF increases.

- False positive probability decreases as the number of hash(x) functions increases.

The BF **fill ratio**, also known as **filter density**, is defined as the ratio between the number of cells set to 1 and the BF size. A BF with a fill ratio that asymptotically approaches to 100% denotes a saturated filter. A good indicator for filter **saturation** is a significant increasing in false positive filter responses [223]. Continuous addition of new elements to a saturated filter is completely inconsequential, as saturation renders the BF useless, due to the excessive false positives responses [15]. The balance between an efficient memory management and an acceptable answer accuracy has been the object of previous studies [35, 52, 227], which agree that the non zero bits should be approximately half of the filter size, *i.e.* a fill ratio $\approx 50\%$.

Both formulas from Eq.(2.3) and Eq.(2.4), allow to calibrate k and f. In practice filter size m calibration follows a simple rule that, when knowing the number of elements to be included in the filter, the filter size should be ten times greater, *i.e.* $\mathsf{m} \approx 10 \times \mathsf{n}$. An approach to estimate k initiates by establishing some bounds to the expected false positive probability, as an example $\mathsf{f} < 1\%$. Afterwards, considering Eq.(2.4) the optimal number of hash(x) functions prediction is $\mathsf{k} \approx 7$.

BF probabilistic properties are explored with more detail in Chapter 4, devising a new BF variation extending the traditional boolean BFs nature. The affinity between Bloom filters and search mechanisms for WSNs is posteriorly evidenced in Chapter 5, where a new search algorithm benefits from the efficiency of a new probabilistic data structure, which is included on every broadcast message for knowledge propagation.

2.5 Exploratory Work

The exploratory work corresponds to our initial efforts to obtain practical experience with current solutions for limiting broadcast in multi-hop networks. It was a fundamental step in the early stages of the research, allowing to gradually gain some experience and understand the related work delimitation.

Initially we take contact with the network simulation tools, as suggested in surveys [135, 226]. However, only some of those may be used for MANETs and

WSNs communication scenarios, and operating without using access points or base stations. This is the case of *ns-2* [79], GLOMOSIM [286], J-SIM [249], OPNET [234], OMNET++ [260], QUALNET[1] and TOSSIM [150], to name only a few. Unfortunately, there are significant performance differences between the network simulation tools, when comparing them for the same multi-hop MANETs routing protocols [267].

The *ns-3* discrete-event simulator is one the most popular network simulation tools, leveraging on innovative features from the C++ programming language and the computing power of the latest hardware. However, considering the lack of backward compatibility between *ns-3* and its predecessor *ns-2*, it was decided to continue performing simulations with the previous version, well consolidated at the early stages of this project [57, 135, 247].

Some of the most motivating reasons to use simulation are the reduction of the deployment time and cost of managing real equipment, enabling flexibility and scalability. We progress the investigation using simulation tools conscious that evaluation requires realistic features such as *i*) three-dimensional space, *ii*) obstacles, *iii*) link asymmetries, and *iv*) unpredictable fading, in order to approximate simulation conditions to experimental ones [140]. Through simulations, we concentrate efforts on the algorithm performance, and the implementation code is contained within a single logical component, which is clearly defined and accessible for refinement during development [47]. However, the exploratory work focus on the communication high level aspects, which may be considered orthogonal to lower layer issues such as propagation models, or MAC protocols.

For exploratory purposes with topological issues and query propagation, we create a self made simulator designated as Synchronous Round-based Simulator (SRS). We developed SRS for having full control on a simple customizable tool, useful for conducting preliminary tests with search algorithms and experiment distinct network topologies. The SRS allows to perform high level comparisons between uninformed multi-hop search mechanisms competing approaches [159]. Whenever new algorithms reveal promising features, we set off from SRS and implement them in the *ns-2* simulation tool, thus considering more realistic assumptions.

---

[1] `http://web.scalable-networks.com/content/qualnet`

2.5.1 System Model

The system model describes the requirements, functionality, assumptions and the design specifications that assist for Synchronous Round-based Simulator (SRS) development.

In SRS the network topology is modelled by undirected graphs G. Each graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is defined by a set of vertices V (representing network nodes) and a set of edges E between the vertices (representing bidirectional radio links) [282]. SRS assumes unpartitioned networks, *i.e.* every pair of distinct nodes in the network is connected through some path. In graph theory, an undirected graph with the previous characteristics is denoted as **connected component**, *i.e.* every vertex is reachable from any other vertex.

The basis of SRS is an untimed **synchronous** system model, where SRS may capture untimed-trigger events (*e.g.,* send/receive), considering the sequential relations deriving from the exchanged messages between network nodes, in synchronous communications rounds. The first node to transmit a message is the initiator node. A message is eventually received by every neighbour of the initiator node, which constitute the subsequent **round** members. When nodes from a given round pretend to retransmit the received message, the SRS assumes that all retransmissions may take place successfully, and a new set of nodes are scheduled for an additional subsequent round. Every node within each round, receives the most recent messages from all its neighbours. Each node may perform some local computation to decide when it will forward the received message to its neighbours [169]. Every node operates in a *"lock-step"* fashion, and for each round the **hop count** is incremented. Moreover, the hop count between source and destination corresponds to the minimum number of intermediate nodes that repeat the received message, for a particular path. This counter (hop-based) is an acceptable metric for **distance**, that is widely used in the evaluation of wireless networks [43, 188, 216, 239, 252].

The hop duration (hop elapsed time) depends on several parameters such as *i*) the number of nodes available on that hop which are trying to concurrently retransmit the received message, *ii*) distance between nodes, *iii*) signal propagation conditions, *iv*) the communication baud rate, and *v*) message size. In this synchronous model, SRS assumes the existence of a **hop time** slot ($\mathsf{T}_{\mathsf{slot}}$) capable of accommodating all physical and MAC constraints for one hop transmissions. The $\mathsf{T}_{\mathsf{slot}}$ may be dynamically allocated driven by the number of nodes that pretend to transmit during that hop. Although, for simplicity we consider it a constant parameter that is initially set for a given network throughput and topology.

Considering the previous assumptions, the minimum time required to search the most distant node hosting the target record in the network is $2 \times \mathsf{D} \times \mathsf{T}_{\mathsf{slot}}$ (seconds), *i.e.* proportional to the network diameter. This upper bound is given by the worst case scenario, where the initiator node and the hosting node are in the farthest opposite sides. A simple search mechanism require to traverse D hops to gather all the available information, *i.e.* the query will traverse all the network diameter, and other D hops to report the answer back to the initiator node.

Given the small time period in which each broadcast is expected to occur, it is assumed that node movement is not sufficiently fast to introduce relevant changes in network topology during the transmission. Therefore, node movement is not explicitly included in the system model. However, since values are averaged among randomized network topologies, the model includes some of the dynamics that are typical of node mobility.

The model assumes that nodes and links are perfect, respectively never failing or losing messages. Messages spend the same fixed amount of time to traverse every link. Consequently, implementation does not associate costs to the graph edges. When the timer expires for any given query, it is assumed that the broadcast reach every node.

There are some additional **assumptions** to consider in parallel with the system model previously described which are responsible for SRS main restrictions and limitations: *i*) nodes are independent and have unique identifiers, *i.e.* $\mathsf{node(ID)}$, *ii*) absence of topological information, *iii*) nodes only know local information, *i.e.* hosted records, *iv*) topology stability during a query propagation, and *v*) nodes are indistinguishable, *i.e.* nodes are identical, except the initiator node that is the broadcast source and responsible to initiate a new search.

The generic synchronous system model does not allow for randomization during the execution, although SRS introduces randomization effects into the simulation, by generating a pool of random network topologies, at the beginning of each simulation run. More detail about the SRS is described in the remaining sections of the current chapter.

2.5.2 Flooding-Unlimited Search

A simple search mechanism approach is known as flooding-unlimited, where a query message is broadcast and propagated by other nodes, exploring every node stored records in a multi-hop fashion seeking for target records through the network [169]. The exploratory work initiates the background study, including the flooding-unlimited implementation, as described by Algorithm 1 (p. 30), increasing our awareness with search mechanisms principles in multi-hop networks.

From the topological perspective, searching through a network is similar to searching through a graph G=(V,E) representing the network topology. SRS implements the uninformed flooding-unlimited search mechanism based on the BFS algorithm. The BFS explores neighbour nodes first, before moving to the next hop of neighbours. SRS does not consider any kind of information to control the query messages flux.

The SRS **uninformed** flooding-unlimited search mechanism implementation meets to the following specifications. At any point during execution, there are a set of nodes visited, initially just the initiator node, designated as $\mathsf{node(0)}$. As $\mathsf{node(0)}$ sends out a search message identified by (msgID) to every outgoing neighbours node, corresponding to a query propagation $\mathsf{1^{st}}$ hop. When an unvisited node receives a search message, it marks the query ($\mathsf{queryID}$) as processed and forwards a message copy to all of the outgoing neighbours. Duplication retransmissions are avoided by logging locally the tuple $< \mathsf{msgID}, \mathsf{queryID} >$, ensuring that a node only retransmits one message for a given query. The query message will eventually be delivered to every node belonging to graph G=(V,E) and target records will eventually be discovered.

Such a simple broadcast implementation is an uncontrolled mechanism, that stops by itself without any explicit **termination** condition, *i.e.* the broadcast ends when the retransmitted message reaches the most distant nodes and there are no more unvisited nodes.

Usually, to compare the performance of algorithms in synchronous distributed systems two major type of **metrics** are often considered, revisited in the context of wireless networks [258]:

Time Complexity For quantifying the amount of time required for algorithm execution we estimate the time complexity, which is the elapsed time during query propagation. Our estimation counts the number of hops occurred until the query processes terminates, which in our system model is proportional to time complexity. Notice that multiple nodes may be members of the same hop.

Communication Complexity For quantifying the communication complexity we estimate the communication overhead, which is the number of bits that need to be transmitted. Our estimation counts the number of exchanged (non-null) messages until a particular query stops being retransmitted.

Algorithms in distributed systems require computation and network resources, which demand may be estimated with previous metrics. One strategy to maximize an algorithm efficiency is minimising the required resource usages, considering both time and communication components.

Theoretical analysis for search mechanisms based in the BFS flooding algorithm reveal the following results: *i*) the time complexity is linear and given by $\mathsf{O}(\mathsf{D})$ rounds, and *ii*) the communication complexity is $\mathsf{O}(|\mathsf{E}|)$ that for dense graphs is given by $\Theta(\mathsf{n}^2)$, which is quadratic [169].

Assuming that for wireless devices the energy consumption is linear dependent on communication complexity, one strategy to improve an algorithm energy efficiency is reducing the number of exchanged messages. Although, there are several models to examine energy consumption for node send/receive network-layer packets. Usually, models assign a fixed cost associated with channel acquisition and an incremental cost proportional to the packet size [83,259]. Nevertheless, accurate models must consider additional costly protocol behaviours and link-layer issues that have a relevant impact for unstructured networks [83].

2.5.3 Synchronous Round-based Simulator (SRS)

This exploratory work section is concerned with practical experimentations and contains the details supporting the **Synchronous Round-based Simulator (SRS)** application prototype development.

The SRS prototype aggregates several independent open source tools, each one having a distinct interfaces. The prototype allows to assist the research progress and unify the access to several tools on a single simple interface. Users may visualize several network topologies and adjust algorithm main parameters, exploring for alternative solutions. The SRS is capable of giving a preview on available topological scenarios, setting basic parameters, easily triggering the selected search mechanisms and specifying the number of simulation runs. The SRS runs on a graphical Linux environment and requires the following tools:

BASH Command language interpreter is BASH, providing the application script programming language.

Python,GTK2 The main programming language is Python, used with GTK2 for SRS graphical implementation and for algorithm codification, such as Algorithm 1 (p. 30).

NetworkX Graph topologies are generated with NetworkX, representing synthetic network topologies.

matplotlib SRS displays graph topologies using the matplotlib, which is a 2D plotting library.

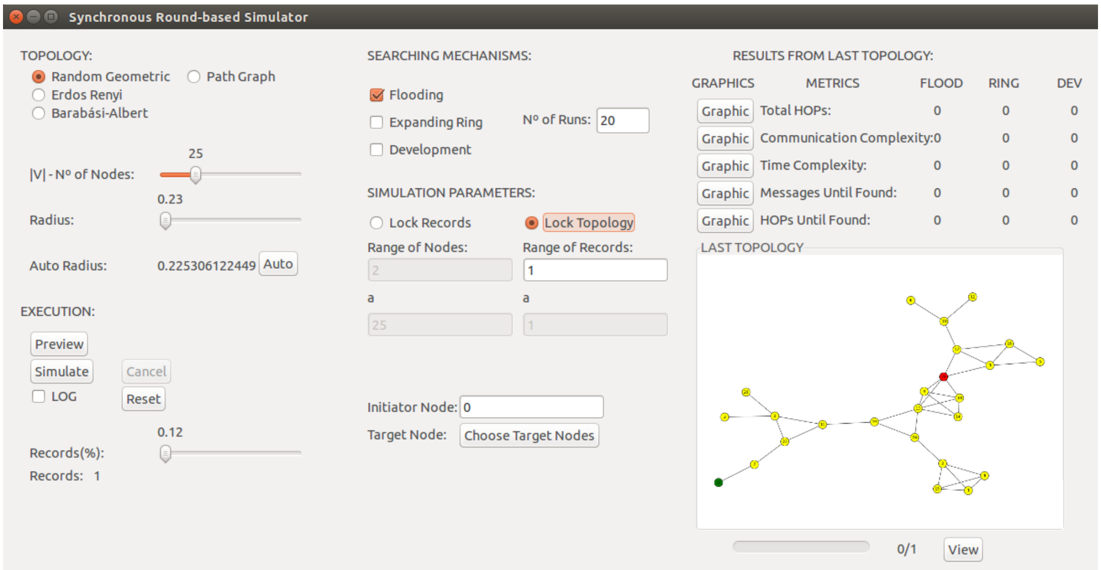
AWK For transforming the simulation output files into less complex log files, SRS uses AWK (developed by Aho, Weinberger and Kernighan) scripts, processing data rows and columns.

Gnuplot For facilitating the results analysis, SRS graphically represents the processed data using Gnuplot, which is a Command-line driven graphing utility.

The prototype functionalities are available in the SRS main screen, as depicted in Figure 2.7. The **SRS main components** are organized in group functionalities and described as follows:

Topology In the topology group, users can: *i*) select between four modelling graphs, Random Geometric, Erdős-Rényi, Barabási-Albert and Path Graph, and *ii*) set the number of network nodes, by choosing the number of graph vertices $|\mathsf{V}|$.

Execution The execution group, contains the Preview button that refreshes the selected graph topology displaying. The Simulate button initiates the simulation process. A text LOG file may be activated for error reporting and debugging. The Reset button clears all simulation data.



TOPOLOGY:
● Random Geometric ○ Path Graph
○ Erdos Renyi
○ Barabási-Albert

|V| - Nº of Nodes: 25
Radius: 0.23
Auto Radius: 0.225306122449

EXECUTION:
☐ LOG
Records(%): 0.12
Records: 1

SEARCHING MECHANISMS:
☑ Flooding
☐ Expanding Ring
☐ Development
Nº of Runs: 20

SIMULATION PARAMETERS:
○ Lock Records ● Lock Topology
Range of Nodes: 2 a 25
Range of Records: 1 a 1
Initiator Node: 0

RESULTS FROM LAST TOPOLOGY:

GRAPHICS	METRICS	FLOOD	RING	DEV
Graphic	Total HOPs:	0	0	0
Graphic	Communication Complexity:	0	0	0
Graphic	Time Complexity:	0	0	0
Graphic	Messages Until Found:	0	0	0
Graphic	HOPs Until Found:	0	0	0

0/1

Figure 2.7: Main screen interface of the SRS prototype.

Search Mechanisms In the search mechanisms group, users may select which algorithm will be considered for simulation, in order to collect data for statistical analysis and comparisons. Initially, only flooding was available. At later stage, the evaluated search mechanisms (Section 3.4) were added to the simulator, thus consolidating prototyping during the development phases. New search mechanisms can be added to SRS by an Application Programming Interface (API) written in Python.

Simulation Parameters Users may set parameters to control the simulation execution, namely the number of runs, *i.e.* distinct topologies of the same graph class, the record replicas rate, or the number of absolute nodes owning record data copies.

Results At the end of the simulation, the results data summary of each metric for the last topology is represented in a table view. The corresponding topology is also displayed using the *matplotlib* tool. The Graphic label buttons show the overall results for each metric and for the selected search mechanisms, corresponding to the average values computed for the selected number of simulation runs.

Each simulation consists of several runs, each one composed of multiple search queries. In each run a new topology is generated and records are randomly assigned to a fraction of the total number of nodes. Before activating the execution button, users may choose the query initiator node, select a search mechanism, or accept the default parameters, as depicted in Figure 2.7. The search task mimics a broadcast propagating queries through the network. As nodes forward the query message, the target record location may be discovered. When the initiator node hosts the target record, a new target is randomly generated.

Network radio connectivity is modelled by graph topology, not considering explicit models for packet loss or radio interference. Device neighbours respect the node degree graph assignment, *i.e.* neighbours are associated with one hop distant wireless devices, and the time required for those retransmissions never exceeds $\mathsf{T}_{\mathsf{slot}}$ (p. 49). All the depicted graphs show the initiator node in green and two randomly distributed nodes hosting the target records in red, to visually inform users about their location. It should be noted that those informations are unknown to the search algorithms, which are unaware of any topology knowledge.

Setting only one record per node, the number of nodes must be superior to the number of records ($\mathsf{N} > \mathsf{R}$). Adjusting N allow users to gain sensitivity to the maximum number of expected hops, depending on the selected topology. Incrementing R, increases the probability to find records closer to the initiator. The Lock Records option may be used to study the impact of adding nodes (N), while maintaining the same number of records (R). With this option, the first generated graph starts with a minimum of two nodes, then new graphs are generated by adding one more node at time, until the graph achieves the maximum number of nodes (N). In contrast, the Lock Topology option sets the number of nodes (N) invariant maintaining the same graph topology, while enabling to study the impact of incrementing the number of target record copies (R).

As depicted in Figure 2.7 SRS displays in the LAST TOPOLOGY, an image representing the graph topology generated in the final simulation run. It assists users adjusting topological parameters, showing a preview of the generated topologies.

Network Graph Models As described in the system model (Section 2.5.1), the SRS prototype represents distinct network topologies by implementing several synthetic graph models. Two notations were adopted for referencing nodes:

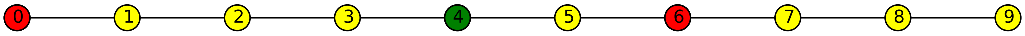


Figure 2.8: Path topology, with $\mathsf{N} = 10$ nodes, linearly connected by $\mathsf{N} - 1$ edges.

i) $\mathsf{node(ID)}$, represents a single node that is clearly identified in the graph (unique number), and *ii*) $\mathsf{node_{hop}}()$, represents a set of nodes that are at hop *jumps* (distance) from the initiator node. Applying the notation to the graph depicted in Figure 2.8, it results in the following examples: *i*) $\mathsf{node_0(4)}$ is the initiator node, and *ii*) $\mathsf{node_2}() = \{\mathsf{node_2(2)}, \mathsf{node_2(6)}\}$ represents nodes belonging to the 2^{th} hop.

Path Graph The simplest topology model is a path graph. Path graphs represents linear network topologies, as depicted in Figure 2.8 (with $\mathsf{N} = 10$ nodes, that is enough to perceive the network topology).

Random Graphs Combining graph and probabilistic theories result in several random graphs. Very often WSNs topologies may be modelled with random graphs. Random graphs are obtained by starting with a set of V isolated vertices and randomly add successive edges E between them [99].

The SRS prototype implements three distinct models of random graphs, namely: *i*) Erdős-Rényi, *ii*) Barabási-Albert and *iii*) Random Geometric Graph (RGG). Examples for these graph models are depicted in Figures 2.9-2.11 respectively. For better perception and readability all topologies were limited to $\mathsf{N} = 50$ nodes per graph.

Erdős-Rényi Assuming that the propagation of information path evolves like an epidemic flooding, the Erdős-Rényi graph (Figure 2.9) is a very successful model [77]. The Erdős-Rényi graph is also known as binomial graph. The graph is generated by connecting nodes randomly. Each edge is included in the graph with probability p and independently from every other edge [77]. From the SRS interface, users may switch to this topology and set the probability p for edge creation.

Barabási-Albert The human-made networks such as the Internet, or other systems that have power-law (or scale-free) degree distributions, may be mod-

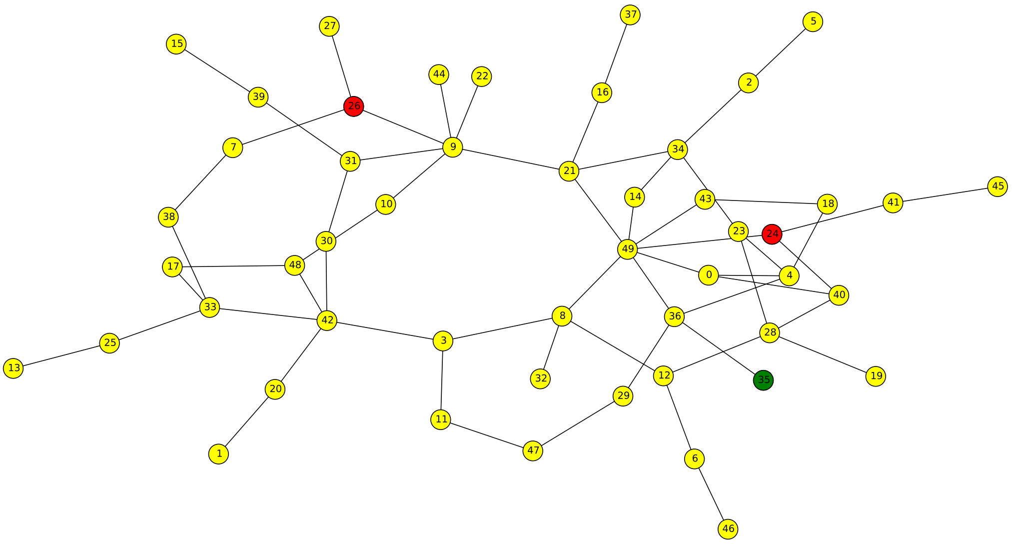


Figure 2.9: Erdős-Rényi topology, with $\mathsf{p} = \mathsf{0.04}$.

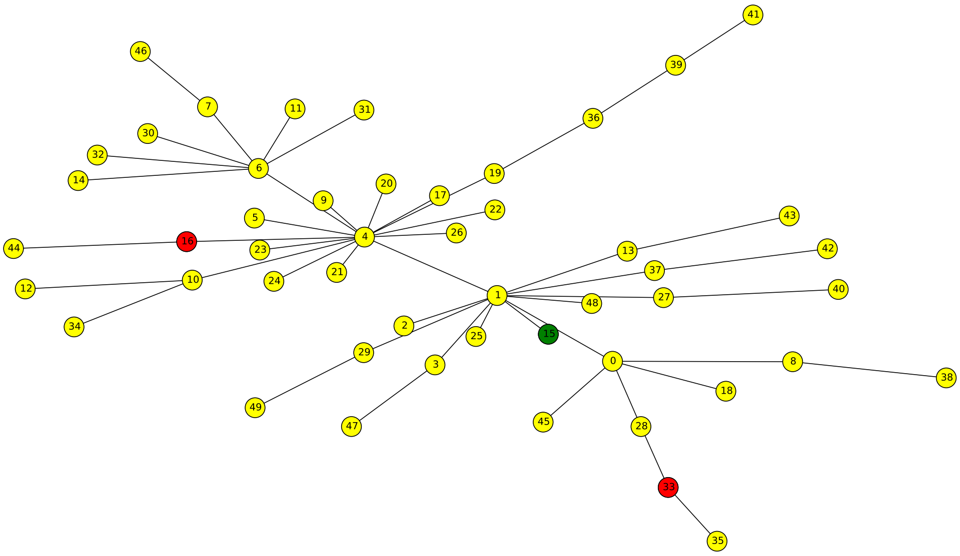
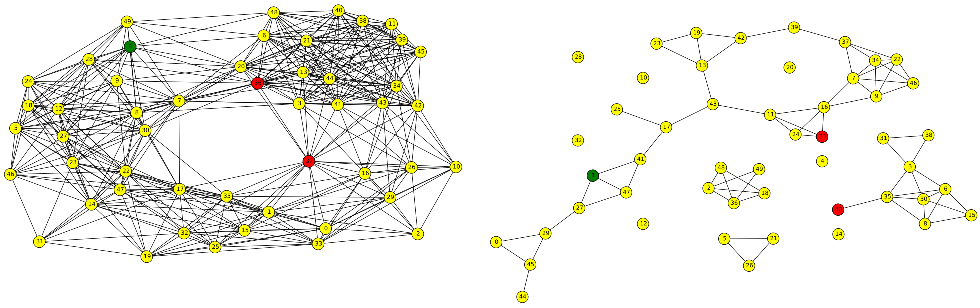


Figure 2.10: Barabási-Albert topology, with $\mathsf{m} = 2$.

elled by the Barabási-Albert approach, as depicted in Figure 2.10. It generates random networks using a preferential attachment mechanism, where nodes with higher degree are likely to receive new links [13]. From SRS interface, users must select the number of edge connections (m) to attach from a new node to existing nodes.



(a) RGG , with r = 0.517

(b) RGG , with r = 0.104

Figure 2.11: The r influence in the RGG=(V,E) connectivity.

Random Geometric Graph (RGG) The RGG model places nodes uniformly at random in the unit square, independently of its position [105]. The connection between nodes is defined by the threshold of the distance between them, *i.e.* radius (r). It is used for synthetically represent MANETs or WSNs topologies. The nodes are randomly deployed in a finite region. However and as far as we know, there is no deterministic solution to obtain the number of nodes that are required to cover a given region (*e.g.,* unit square) with a connected graph [51, 78].

From SRS interface, users may set the r parameter tuning graph connectivity, as depicted in Figure 2.11. When r is too high, this usually results in high degree graphs as depicted in Figure 2.11a, or when r is too low it will generate disconnected graphs as depicted in Figure 2.11b.

We focus our attention on synthetic random geometric topologies considering that r represents the radio propagation range between wireless devices, thus simplifying the MANET and WSNs topology model [243]. During our experimentations we observed the possibility to generate a connected RGG=(V,E) by choosing an acceptable value for r, depending on the number nodes (V) distributed in a limited area. Previous works about random geometric graphs, established some significant **connectivity thresholds** bounds [69, 207]. The RGGs have an asymptotic connectivity threshold value for r [76]. The following radius (r) upper and lower bounds (euclidean distance) are estimated considering the unit square deployment area.

The SRS uses the NetworkX function random_geometric_graph(N,r,2) to generate a RGG, that places N nodes uniformly at random in a two-dimensional square $[0,1]^2$ and connects any two nodes that distance at most r. The choice of r has direct consequences on the degree of each node and graph connectivity.

Upper Bound The upper bound ($\mathsf{r}_{\mathsf{upper}}$) predicts that there is a high probability that $\mathsf{RGG}=(\mathsf{V},\mathsf{E})$ is a connected graph, by choosing $\mathsf{r} > \mathsf{r}_{\mathsf{upper}}$. Using an approach based in a numeric result estimation, the $\mathsf{r}_{\mathsf{upper}}$ is given by Eq.(2.5) reproduced from [24]. An experiment using the upper bound is depicted in Figure 2.11a.

$$\mathsf{r}_{\mathsf{upper}} \geq 0.3043 \times \log(\mathsf{N}) \qquad \therefore \qquad (\mathsf{N} = 50 \Rightarrow \mathsf{r}_{\mathsf{upper}} \simeq 0.517) \tag{2.5}$$

Lower Bound The lower bound ($\mathsf{r}_{\mathsf{lower}}$) predicts that $\mathsf{r} < \mathsf{r}_{\mathsf{lower}}$ will result in a disconnected graph. An estimation for $\mathsf{r}_{\mathsf{lower}}$ is given by Eq.(2.6), that depends on the number of nodes N [39, 98]. An experiment using the lower bound is depicted in Figure 2.11b.

$$\mathsf{r}_{\mathsf{lower}} = \sqrt{\frac{\log(\mathsf{N})}{\pi \times \mathsf{N}}} \qquad \therefore \qquad (\mathsf{N} = 50 \Rightarrow \mathsf{r}_{\mathsf{lower}} \simeq 0.104) \tag{2.6}$$

Generically, previous results were confirmed by our experiments, moreover we acquired further sensibility with RGG topology issues.

The SRS tool is useful for understanding the impact of the radio range on the unstructured network connectivity. Wireless networks applications are designed to provide good connectivity with some redundancy [194]. However, it makes no sense to have too much physical redundancy in a multi-hop network, due to the waste of resources that occurs in scenarios with high node density [24].

The comparison depicted in Figure 2.11 reveals that scenarios using previous bounds ($\mathsf{r}_{\mathsf{upper}}$, $\mathsf{r}_{\mathsf{lower}}$) should be avoided, due to extreme connectivity properties. When r is near to upper bound, the generated topology could have many redundant links between nodes, as depicted in Figure 2.11a. Nevertheless, the connectivity fault is a consequence of choosing r near to the lower bound, as depicted in Figure 2.11b.

SRS-AUTO The solution implemented in SRS solves the problem of generating a connected random geometric graph topology, in the following way. SRS computes the lower bound connectivity estimator for r, given by Eq.(2.6) and uses it as a starting value to generate an RGG. When the generated graph is not connected, it repeatedly increments r by a constant step until a connected graph is found. This is a basic iterative (without pretending to be the most efficient solution) approach to find an estimation for r value, converging to a solution with an acceptable level of link redundancy.

To illustrate the previous iterative process that generates a connected RGG, an excerpt of the corresponding Python code is depicted in Listing 2.1, where r is incremented by $\mathsf{0.005}$.

Listing 2.1: Generating a connected RGG (SRS-AUTO)

```python
step=0.005
radius=math.sqrt(math.log(N)/(math.pi*N))
G=nx.random_geometric_graph(N,radius ,2)
while not nx.is_connected(G):
        radius=radius+step
        G=nx.random_geometric_graph(N,radius ,2)
```

We cannot ensure that all subsequent new topologies with the same radius (obtained in Listing 2.1) will be connected. Therefore, the automatic mode (AUTO) of the SRS, discards all the unconnected graphs and continues to generate new graph instances with the same parameters, until a complete set of connected topologies is obtained. We performed some experimental tests, in order to obtain $\mathsf{1\,000}$ RGG connected topologies with $\mathsf{N} = \mathsf{50}$ nodes. The connectivity results for those experiments are summarize in Table 2.1, evidencing the difficulties to obtain a connected RGG with an acceptable redundancy.

From Table 2.1 first line, all the experiments with $\mathsf{r_{upper}}$ result in connected graphs, at the expense of an excess of redundancy. We found some traces of connectivity when considering $\mathsf{r_{lower}}$, however the failures percentage is too high

Table 2.1: Testing the RGG connectivity for proposed radius (r) bounds.

Radius	**Iterations**	**Fails**	**Topology**	**Connectivity**
$\mathsf{r_{upper}} = \mathsf{0.517}$	1 000	0%	Figure 2.11a	Excess of Redundancy
$\mathsf{r_{lower}} = \mathsf{0.104}$	1 000	99.98%	Figure 2.11b	Traces of Connectivity
$\mathsf{r_{AUTO}} = \mathsf{0.203}$	9 582	89.56%	Figure 2.13	Acceptable Redundancy

(taking too many resources to converge). The SRS $\mathsf{r_{AUTO}}$ mode generates a total number of 9 582 RGGs, to obtain (iteratively) 1 000 connected graphs, corresponding to 89.56% failed attempts to generate a connected graph. The $\mathsf{r_{AUTO}}$ is the radius average value of the 1 000 connected RGGs.

The number of wireless links per node is directly related to the node degree $\deg(\mathsf{v_i})$, when considering a graph representing the network topology. Using a bigger step value (Listing 2.1) will accelerate the iterative graph generation and reduce fails percentage. However, it will have a negative impact in node degree, generating an excessive number of links per node. Table 2.1 denotes that it is possible to reduce the number of redundant links, and continue to ensure the graph connectivity.

RGGs are irregular graphs (not every node has the same degree) and it is not possible to specify a unique graph degree. Instead, we inspected the **maximum node degree** $\Delta(\mathsf{RGG})$ for each topology, as defined by Eq.(2.7):

$$\Delta(\mathsf{RGG}) = \max\{\deg(\mathsf{v_i})\} \therefore \forall \mathsf{i} \in \mathsf{N} \tag{2.7}$$

Figure 2.12 depicts the obtained results, checking the maximum node degree relative frequency, for different r values as proposed in Table 2.1. Thus, each plot line representing different radio communication ranges. Note that, topologies with $\mathsf{r_{lower}} = 0.104$ correspond to disconnected graphs, whereas both $\mathsf{r_{upper}} = 0.517$ and $\mathsf{r_{AUTO}} = 0.203$ consider exclusively connected graphs.

The SRS Auto mode is capable of generating connected graphs and maintain a lower $\Delta(\mathsf{RGG})$ (Figure 2.12). The Auto topologies have the most frequent maximum number of neighbours around $\Delta(\mathsf{RGG_{Auto}}) = 10$, which is a significant reduction compared with the $\Delta(\mathsf{RGG_{upper}}) = 14$ obtained by theoretical estimation.

Studying the impact of the radius parameter by comparing the consequences to the maximum node degree for each graph, enables to tune the r parameter, in order to build more realistic topology models.

One example of a RGG generated with the automatic r threshold (SRS-AUTO) is depicted in Figure 2.13. Thousands of similar $\mathsf{RGG}=(\mathsf{V},\mathsf{E})$ graphs have been generated to model synthetic topologies for wireless unstructured networks in Chapters 3 and 5.

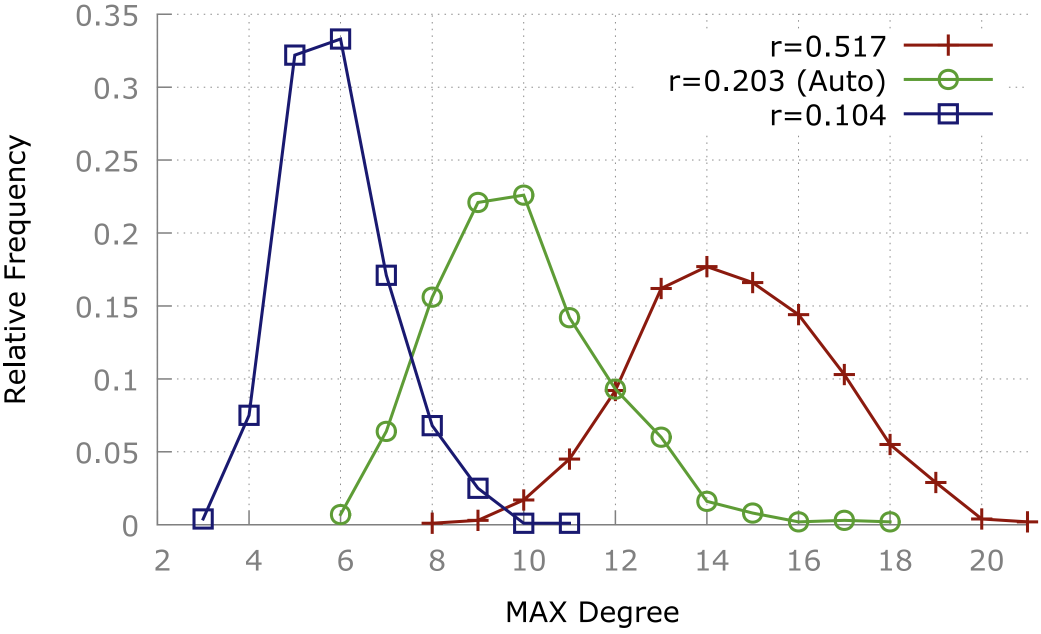


Figure 2.12: Radio communication range (r) influence in the maximum node degree, Δ(RGG).

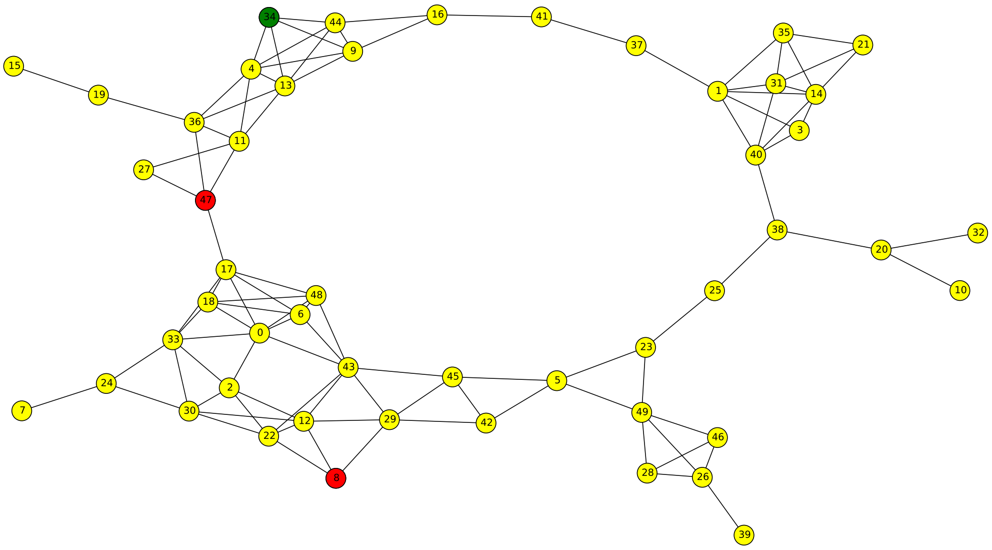


Figure 2.13: Random Geometric Graph topology, with an automatic r threshold.

2.5.4 Results: Multi-Hop Broadcast by Flooding

The SRS supports several search mechanisms. However, the first implementations just consider an uninformed multi-hop broadcast transmission mechanism, also designated as flooding-unlimited search.

Listing 2.2 depicts an excerpt representative of the flooding-unlimited codification. The code is written in Python and included in the SRS prototype, mimicking a multi-hop broadcast search mechanism, without message duplication. With this, SRS has a flooding-unlimited search mechanism for finding target records through a graph representing a given network topology.

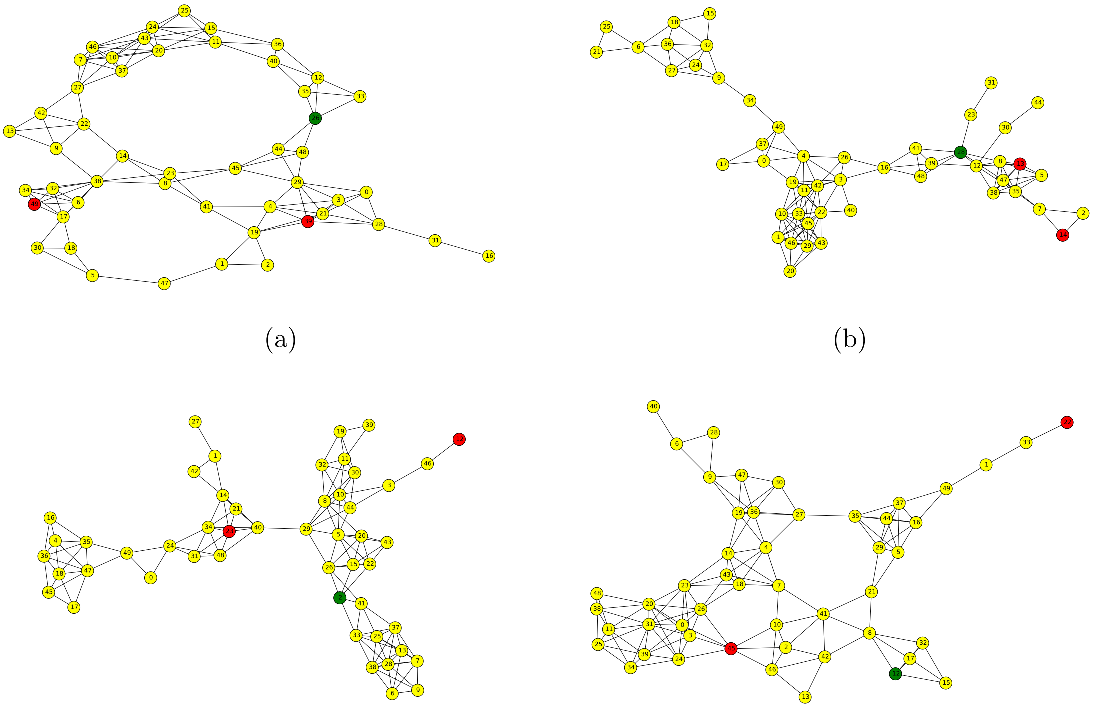
Listing 2.2: The SRS implementation for a multi-hop broadcast

```
actualround.append(initiator)
while (len(actualround)>0):
  nextround=[]
  for v in actualround:
    history.append(v)
    candidates=G.neighbors(v)
    trans+=1
    for temp in candidates:
      if ((temp not in history) and (temp not in nextround)
          and (temp not in actualround)):
        nextround.append(temp)
  nodeshop[hop]=len(actualround)
  hop+=1
  actualround=nextround
```

At the beginning of a simulation, SRS randomly chooses the initiator node and, using the synchronous round-based model, messages follow the graph connectivity visiting all the other nodes. A counter (`trans`) is included to monitor all the node transmissions. All lists (data structures), *i.e.* `actualround`, `nextround`, `history` are initialized empty, and counters *i.e.* `trans`, `hop` are set to zero.

The flooding-unlimited is complete when every node retransmit the broadcast message. At the end of each query, the SRS computes the number of nodes in the graph and verifies if it is equal to the counted transmissions (`trans`) during the flooding process, thus verifying no message fails and no duplications (for a connected network topology).

The SRS prototype is capable of obtaining very distinct topologies instances from the same topology model. Given the particular interest of RGGs in wireless unstructured networks, Figure 2.14 depicts some instances of distinct RGGs



(a) (b)

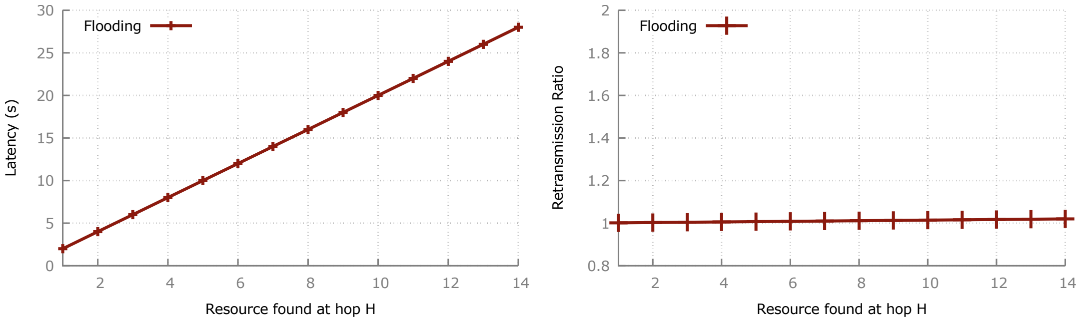
(c) (d)

Figure 2.14: The randomization effect introduced by SRS, using RGGs topologies with $\mathsf{r}_{\mathsf{AUTO}} = 0.203$.

topologies. Notice that, all the four topologies were generated using the same parameters $\mathsf{RGG}(\mathsf{N} = 50\ ;\ \mathsf{r} = 0.203)$.

The **randomization** effect has a significant impact on the graph topology, creating diversity for the same topology model, which is a technique used in other research domains that apply simulation [45]. Scenarios for simulation were configured with a high level of randomization in the network topologies, and consider the execution of several runs during the experimental tests, for obtaining statistically significant results.

At the end of the simulation, the collected data is normalized taking into account the number of rounds and repetition runs. Some indicators (*e.g.,* hop, time complexity and communication complexity) are visualized in the main screen. The Graphic button near each metric on the main SRS screen, as depicted in Figure 2.7 (p. 53), produces a detailed graphical representation showing average values collected from multiple executions.



(a) Time Complexity (b) Communication Complexity

Figure 2.15: Ratification of the expected energy efficiency of the flooding search mechanism. Average results obtained with SRS prototype.

Figure 2.15 depicts the flooding search mechanism average results for latency and retransmission ratio, obtained using the SRS prototype. Figure 2.15a confirms an obvious expectation, latency in flooding is linearly dependent on distance between the initiator and the record location. Figure 2.15b shows that the flooding retransmission ratio is practically constant, evidencing the energy inefficiency of the flooding search mechanism. In Figure 2.15b it is clear that $\mathsf{R} \gtrsim 1$, which means that the expected flooding exchanged messages for any given query are independent of the record proximity and messages are retransmitted by all the network nodes. Even so, the flooding retransmission ratio slightly increases with the distance at which the record is discovered, since the cost of sending the response to the initiator (using a minimal path route) is much smaller when compared to the cost of contacting all nodes.

Assuming that nodes are indistinguishable and operating in similar environment conditions, it is expectable that the network topology may affect significantly the number of transmissions per-hop. Figure 2.16 depicts the average number of contacted nodes per-hop, using the SRS prototype. High rates of **transmissions per-hop** in a shared communication channel (wireless scenarios) have more probability to produce higher collision rates. To measure the impact of different network models, SRS counts the average number of nodes that relay the broadcast message during some hop.

Figure 2.16 compares the evolution of the contacted nodes per-hop in the flooding mechanism for the topologies based on Erdős-Rényi, Barabási-Albert, Random Geometric (RGG) and Path synthetic graphs. These results were ob-

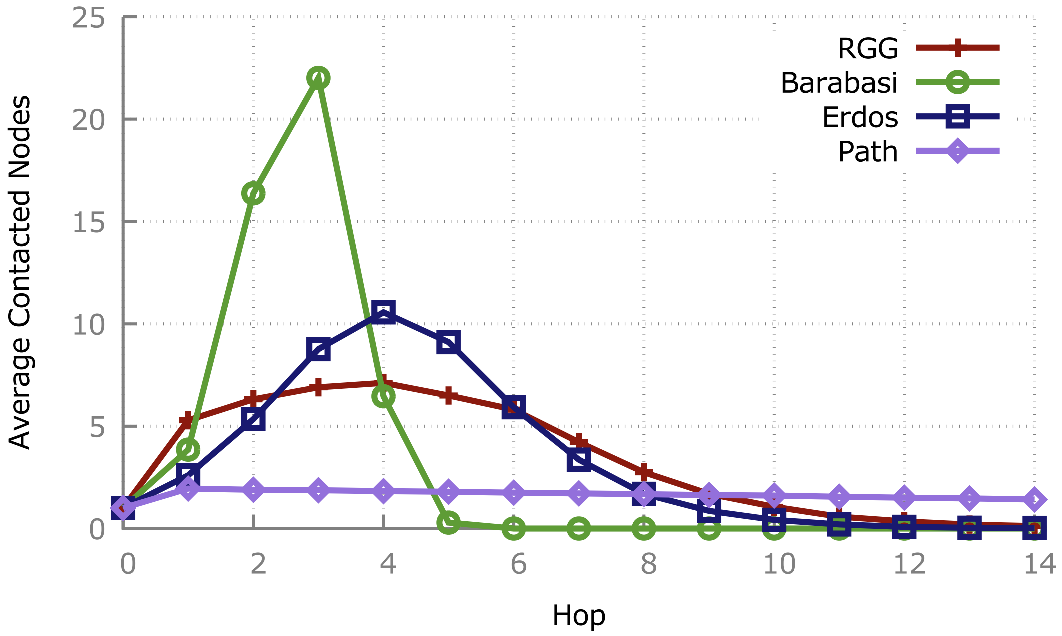


Figure 2.16: Comparing the average number of contacted nodes per-hop in a flooding mechanism, when traversing each topology model.

tained using the SRS tool, upon executing 1 000 simulation runs, for flooding-unlimited experiments.

Previous described experiments with connected graphs ensure that any detected connectivity fault will be a consequence of the radio communication mechanism and not of any deterministic topology fault dictated by the initial node location. It is expected that network connectivity disruption may occur more frequently on topologies with higher number of contacted nodes per-hop. However, the SRS prototype limitations do not allow to verify it, which require a simulation tool that can model radio communication details.

Considering that our primary motivation is to improve energy efficiency (with an acceptable latency) of search mechanisms for unstructured networks, we start by focusing our attention on uninformed search mechanisms and especially the flooding-unlimited algorithm.

The results obtained with the exploratory work confirms the flooding-unlimited inefficiency, as depicted in Figure 2.15b, which is an additional motivation to continue the research for new search mechanisms alternatives, that is paved in the following chapters. On continuation, Chapter 3 proposes a new uninformed flooding-limited search algorithm. Chapter 4 proposes a generic space-efficient data structure suitable to be included in the broadcast messages. Chapter 5 proposes a new informed adaptive search mechanism, using a space-efficient data structure and guiding the query propagation towards the record location.

Chapter 3

BCIR: Broadcast Cancellation Initiated at Resource

On previous chapters, we revisited relevant challenges and issues in the field of energy-efficient search mechanisms over unstructured wirelesses networks. The research community considers energy efficiency to be a major issue in designing network protocols to run on wireless devices with battery constraints, confirming the relevance to investigate energy-efficient broadcast propagations.

When one can determine that a query task has been fulfilled (*e.g.,* find a given target record) and there is no advantage in continuing the query dissemination, the broadcast propagation should ideally be stopped as soon as possible. However, cancelling an ongoing broadcast for wireless unstructured networks is a challenging task, and, by adding control packets to implement some mechanism, it may also increase the number of exchanged messages and instead of benefiting it may have a negative impact on energy efficiency.

In Chapter 2 we describe some approaches that limit the broadcast propagations such as *i*) hop-limited flooding (ERS and ERS-TTL) [56, 181], and *ii*) chasing packets [203, 218]. However, their performance may be improved by proposing new alternative approaches. Notice that previous mechanisms explored the balance between delay and energy expenditure, striving for efficient broadcast propagation strategies on networks with a significant number of nodes and distinct topologies that favoured the appearance of long paths (and diameter) between its nodes. Current chapter addresses flooding controlling issues for WSNs, and presents a new search mechanism designated as **Broadcast Cancellation**

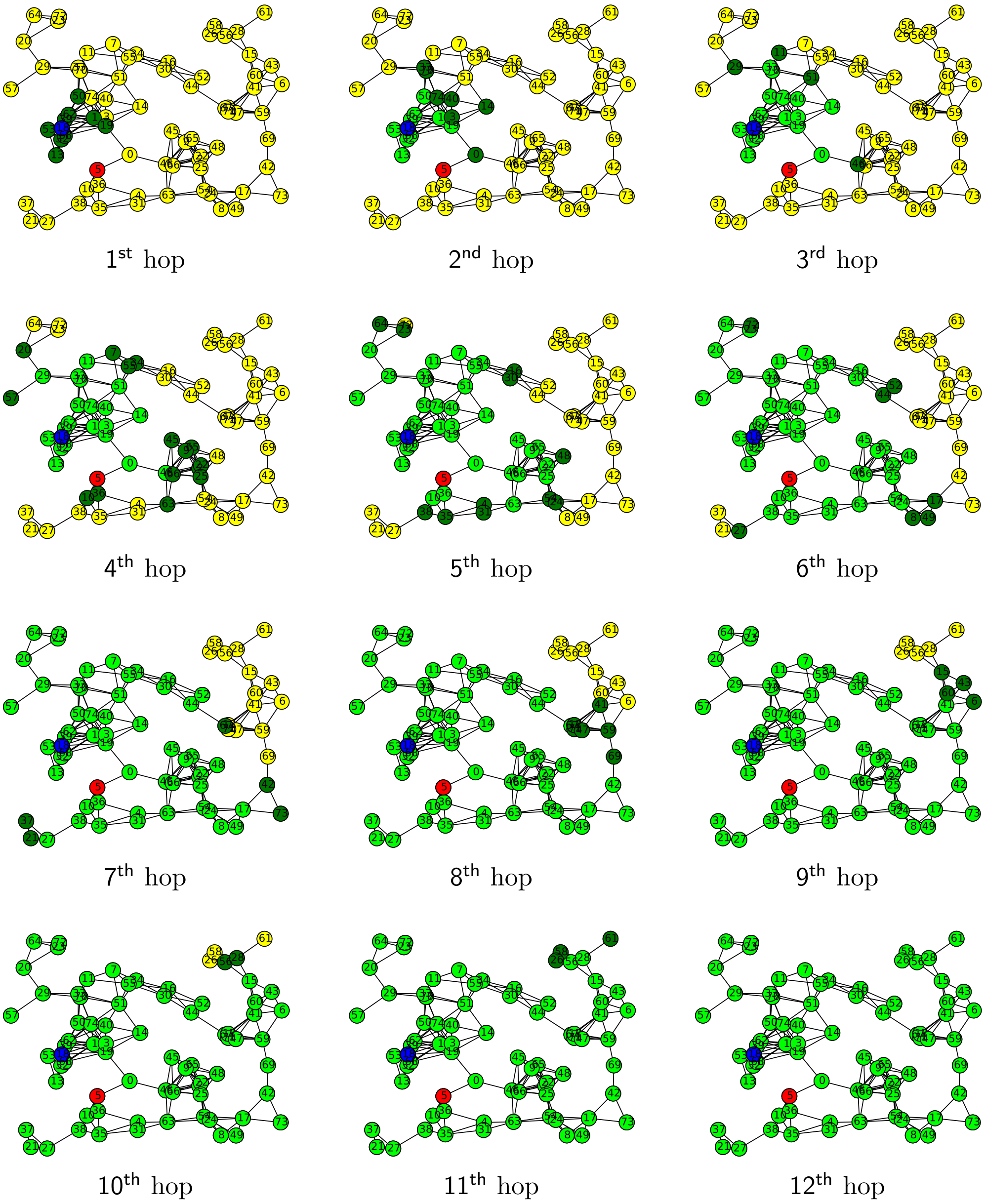
Initiated on Resource (BCIR).

The remainder of this chapter is organized as follows: first, the chapter starts with some challenges for stopping uninformed search mechanisms based on flooding; it reveals the motivation to conduct research into energy-efficient search mechanisms, highlighting WSNs constrains; some of the existing solutions are revisited in the related work, emphasising on techniques to stop ongoing broadcasts, and discusses the energy-time trade-off in the chasing packets approach; then present BCIR, describing the chosen research methodology, including a formulation of a simplified analytical model, and a comparison with its competitors based on metrics such as energy and time; next, it proceeds with the obtained simulation results, specifying the setup parameters enabling to repeat the BCIR experiments, and the result analysis taking into account values reported for each metric; a final discussion highlighting the advantages of using the broadcast cancellation technique based on chasing packets is presented at the end of the chapter.

3.1 Motivation

Unstructured multi-hop networks have a common main feature, the node transmission range is smaller than the network size, preventing direct radio communications between distant nodes. To cope with this limitation, network nodes must act as routers and relay the messages, to maintain network connectivity [37]. Furthermore, topology of the network changes over time, as a result of node mobility and radio signal propagation. Any collected topology information is quickly outdated, making the classic routing principles useless. Typically, unstructured wireless networks use broadcasting for data dissemination [12]. It is often possible to benefit from redundancy in the topology so that all nodes are reached, even if only some of them do actively transmit. Therefore, several mechanisms have been proposed to optimize the number of relay nodes that participate in a given broadcast [103, 256].

The idea of limiting the broadcast scope to save power has been considered by recent **energy-efficient** protocols [217, 288]. Previously reported work concerning search mechanisms denoted a tendency to substitute pure flood-based algorithms by more energy-efficient approaches [56]. To illustrate how flooding affects the retransmission ratio without neglecting the relationship with energy



1st hop 2nd hop 3rd hop

4th hop 5th hop 6th hop

7th hop 8th hop 9th hop

10th hop 11th hop 12th hop

Figure 3.1: Flooding-unlimited search mechanism.

concerns, we setup a simple scenario using the SRS simulation tool described in Chapter 2. The network topology is represented by a RGG=(V,E) graph with $V = 75$ nodes uniformly distributed at random in the unit square, and E radio links resulting from a range threshold set to $r = 0.16$ assuring a connected RGG=(V,E) graph.

Figure 3.1 depicts an example of the RGG=(V,E) topology and the sequence of snapshots represents node transmissions of the same query, resulting from the flooding evolution over time. The motivation to tackle the energy saving challenge is evident when looking at the footprint caused by a flooding algorithm. The initiator node, *i.e.* $\text{node}_0(18)$, is shown in blue and the node hosting the target record, *i.e.* node(5), in red (Figure 3.1). Nodes marked in yellow did not successfully receive any message (standby nodes). After receiving a given search message, nodes schedule themselves to forward that message, and retransmission nodes are marked in green. Dark green represents the most recently contacted nodes, *i.e.* nodes of the current hop, that receive a message but are waiting for the current slot time to expire before carrying out with the propagation.

The example given on Figure 3.1 shows a record which is near the initiator node being found in just 3 hops, although the flooding will continue reaching a maximum of 12 hops until it is terminated, with many unnecessary retransmissions. Considering the topology depicted in Figure 3.1 at the 3rd hop, the search message reaches the node(5) and the record is found on that node. However, the flood search message continues to be forwarded by the other nodes in the same hop until 12th hop, when there are no more nodes to forward the query. As shown above, flooding wastes a lot of energy in a significant number of unnecessary node retransmissions. This simple case shows (in practice) why flooding is a very energy-inefficient search strategy.

Using the SRS simulator, we observe the flooding caused by uncontrolled broadcast propagation, which resembles an expanding wave centred at the initiator node. This is the traditional representation of a flooding process: a single expanding *ring*, that grows with the hop distance until no more nodes are available for message retransmission.

Limiting the broadcasting area Work described in this chapter is motivated by the consideration that limiting the broadcasting area in a multi-hop network will save energy, by not involving all the nodes in the search process (if they may remain in standby mode). Given the interest in increasing the system lifetime, a set of algorithms were specifically developed to be compatible with MANETs resource limitations [37].

An intuitive strategy is geographically limit the broadcast to a target area.

For that, nodes initiating a query may include in the message header information specifying the target geographical boundary, implementing a basic mechanism to limit the broadcast. Any node receiving that message rebroadcasts it only if its own location coordinates are within the geographical boundary, and ignores it otherwise [54]. Location-limited algorithms (*e.g.,* LAR [138]) require that all nodes have continuous access to precise location information. Typically, each node knows its location through some Global Positioning System (GPS) receiver, and propagates this information to its peers, informing other nodes. It should be noted that these requirements limit the scope of applicability of these algorithms [253]. GPS receivers embedded in wireless devices may rapidly contribute to deplete the device battery when GPS module is active. Furthermore, GPS signal reception may be problematic even for outdoor locations, *i.e.* urban canyon problem, wherein built-up urban areas with dense blocks of structures (*e.g.,* skyscrapers) make it impossible for a GPS retrieve its position [278]. Location techniques come in a wide variety of radio access technologies such as GPS, GSM and WLAN. Nevertheless, for wireless-based technologies there are still a number of issues to be solved in order to obtain a better, more resilient and accurate location service [185].

Uninformed search mechanisms Despite the above problems, there are many reliable location services that improve the informed search mechanisms performance. Although, our focus on the current chapter will be on uninformed search mechanisms that do not require any location information, and may be used universally under all conditions, with special applicability interest for unstructured networks. Uninformed search mechanisms are divided in two classes:

Hop-limited Broadcast propagation may be limited by counting the number of hops, starting from the initiator node. Hop-limited mechanisms establishes a threshold distance limit, embedded in the broadcast message header, that is decremented on every retransmission. The limiting mechanism only permits node retransmissions of messages with positive counters. DSR [125] was one of the first search mechanisms to use the hop-limited, designed specifically for unstructured wireless networks. The DSR hop-limited component is also known as Expanding Ring Search (ERS) [48, 96, 240].

Chasing Packets A broadcast may be limited by disseminating faster blocking messages to stop an ongoing broadcast before its completion. This approach use chasing packets dissemination for stopping an ongoing broadcast [10]. The limiting mechanism only permits node retransmissions until it receives a chasing packet. One of the most successful example was introduced by Blocking Expanding Ring Search (BERS), also known as Blocking-ERS [203].

Limiting the broadcast with a chasing packets approach is achieved at the expense of adding fine tuned delays at each retransmission step (hop), and increasing the delay value as the search query propagates. Adding delay has the desired effect of opening a time window that provides the opportunity to broadcasting fast chasing packets to stop the original broadcast [48, 203].

Previous classes of uninformed search approaches should operate either without any pre-existing knowledge of the network environment or without considering any previous neighbour context information. Assessing these problems with previous assumptions is a very challenging task. Unfortunately, previous approaches have not been sufficiently explored, and farther alternative strategies may be developed. Both ERS and BERS suffer from energy inefficiency, which will be detail reviewed in the related work (Section 3.2).

On continuation, this chapter will introduce the **cancellation** approach, capable to outperforming current broadcast limiting approaches. Thus, expecting to benefit multiple services and applications that search for content over unstructured networks.

3.2 Related Work

MANETs and WSNs widely use broadcast algorithms for message dissemination [131]. A multi-hop broadcast communication principle sends a message to all network nodes, which is useful for searching purposes [93]. There is a tendency to implement broadcast communications in multi-hop networks using flooding algorithms, due to its implementation simplicity, *i.e.* all nodes retransmit a message when it is received for the first time. However, as observed in Section 3.1, the footprint caused by a flooding algorithm produces a significant number of unnecessary node retransmissions, as illustrated in Figure 3.1 (p. 69), wasting the battery charge, which is a limited resource for autonomous powered devices [216].

Algorithm 2: Hop-limited

```
begin
    msgList ← {};
    upon event RECEIVE($m_s$) do
        if $m_s \notin$ msgList then
            msgList ← msgList ∪ {$m_s$};
            PROCESS($m_s$);  // Node locally processing the received message
            if $m_s$.TTL > 0 then
                $m_s$.TTL ← $m_s$.TTL − 1;
                SEND($m_s$);                        // Broadcast message
            end
        end
end
```

On the related work (Sections 3.2.1 – 3.2.3), a detailed description is presented for hop-limited and chasing packets strategies, used for geographically limiting the broadcast primitive [96]. These mechanisms reduce the energy impact of a search query, controlling the area in which the search is performed and aiming to avoid flooding the entire network [104, 125, 138, 211]. A comparison of the impact in latency and energy consumption, resulting from broadcast containment policies, was presented in [159].

Chapter 2 presented the state-of-the-art background issues and a generic research area overview. On continuation, the related work is focused on what we consider the most representative approaches for controlling a multi-hop broadcast propagation, including a detailed description and practical experiments for ERS and BERS algorithms.

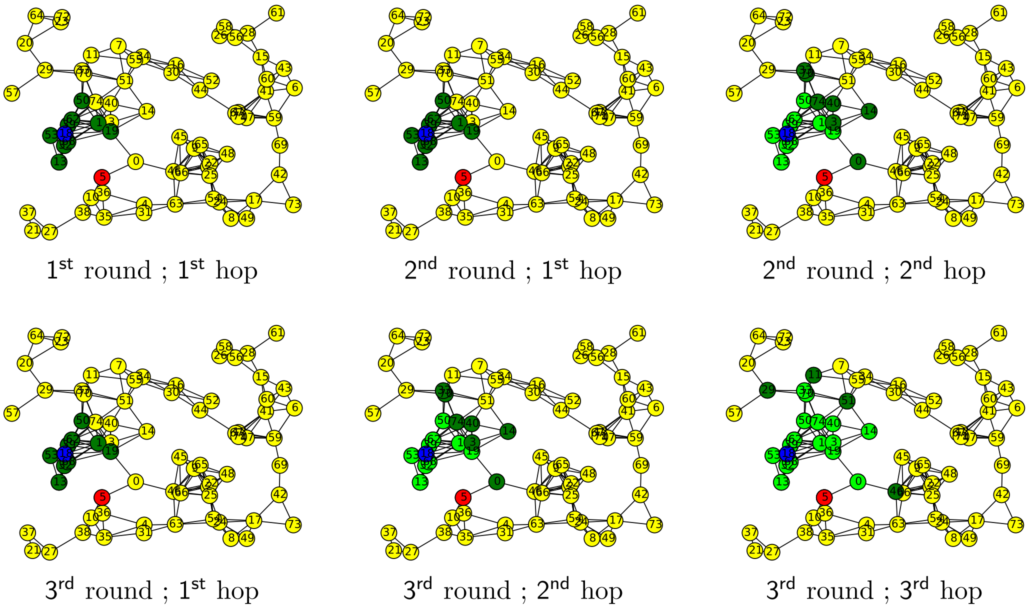
3.2.1 Hop-limited

The main idea put forward by the hop-limited strategy is to control the scope of query search, by counting how far the query should be forwarded from the initiator node, without using any kind of location information. A classical technique to limit broadcast query scope estimates distance using a hop counter for search message (m_s) retransmissions.

The Internet Protocol (IP) uses a mechanism to discard a packet that remains in network for long time, estimating time by counting the number of router retransmissions, using the Time-To-Live (TTL) packet field. Given the previous affinity, the hop-limit counter will be designated as TTL. Inline with DSR [125], and the **hop-limited** mechanism described in Algorithm 2. They assumed that the TTL initial value included in the search message ($\mathsf{m_s}$) header is properly set by the initiator node. Node relaying (message retransmission) is decided locally and independently on each node, according to Algorithm 2 description. Every time a node receives a valid $\mathsf{m_s}$, it will inspect the TTL value. If TTL is a positive non null value the node decrements the TTL field in $\mathsf{m_s}$, and rebroadcasts it. The node relay function will terminate when TTL reaches zero.

The **Expanding Ring Search (ERS)** is a search mechanism that works in rounds and exploring a limited area of the entire network. If the target record is not found until the round ends, then a new searching round is initiated exploring a larger area, as illustrated by the multiple snapshots depicted in Figure 3.2. This iterative mechanism is carried out until the initiator receives a successful answer.

To reproduce the ERS expanding *ring* progression, we set up a scenario with the same topology as defined in the flooding example and depicted in Figure 3.1 (p. 69), using the SRS simulation tool (Section 2.5.3). The ERS limits the search scope by controlling the search message $\mathsf{m_s}$ retransmissions, as described in Algorithm 2.


1st round ; 1st hop

2nd round ; 1st hop

2nd round ; 2nd hop

3rd round ; 1st hop

3rd round ; 2nd hop

3rd round ; 3rd hop

Figure 3.2: Expanding Ring Search (ERS).

ERS The initiator node, *i.e.* $\mathsf{node}(0)$, is responsible to start a new query, broadcasting a search message ($\mathsf{m_s}$), and setting the TTL field included in $\mathsf{m_s}$. The ERS search strategy uses a linearly incremented TTL sequence of values, *i.e.* $\mathsf{TTL} \in \{1, 2, 3, 4, ...\}$. The first round initiates with $\mathsf{TTL} = 1$, thus following Algorithm 2, the query is limited to 1 hop, *i.e.* (1^{st} round ; 1^{st} hop). If a record is found, the node hosting the record ($\mathsf{node_H(ID)}$) sends an acknowledgement message ($\mathsf{m_a}$) through an optimal point-to-point route back to $\mathsf{node}(0)$, informing the record location and the search stops at the end of the round.

In ERS, any node at h hops ($\mathsf{node_h(ID)}$) from the initiator node waits at least a minimum time of $\mathsf{t_h} = 2 \times \mathsf{h} \times \mathsf{T_{slot}}$ ($\mathsf{T_{slot}}$ as defined in p. 49), creating a time window for both messages $\mathsf{m_s}$ and $\mathsf{m_a}$ be propagated. In case of $\mathsf{m_a}$ is not received during the $\mathsf{t_h}$ interval, the $\mathsf{node}(0)$ assumes that the search attempt was not successful. The ERS initiate a new searching round, and the query is reissued by $\mathsf{node}(0)$ with a bigger hop limit (TTL), thus increasing the search area exploring additional nodes, *i.e.* a new bigger search *ring*.

Considering a weak assumption of: *at least one node hosts the target record*, then a ERS **termination** condition may be satisfied. The ERS stops generating extending search *rings* when the initiator node receives a discover acknowledgement ($\mathsf{m_a}$), informing that a record was discovered in the searched area covered by the last *ring*. Otherwise, the ERS eventual flood the entire network, until it expires when a maximum $\mathsf{t_{max_{hop}}}$ is reached. The $\mathsf{m_s}$ is delivered to all nodes and the critical records are eventually discovered.

In the example depicted in Figure 3.2, at the end of the 1^{st} round the initiator $\mathsf{node}(18)$ does not receive any $\mathsf{m_a}$, thus it will assume an unsuccessful attempt to find the record has occurred. Then, the initiator node sends the same query with a $\mathsf{TTL} = 2$, as depicted on the snapshots (2^{nd} round ; 1^{st} hop) and (2^{nd} round ; 2^{nd} hop) from Figure 3.2. Since during the 2^{nd} round no record was found either, the search must continue expanding for an additional round. On the 3^{rd} round the record is found and $\mathsf{m_a}$ is sent from $\mathsf{node_3(5)}$ to $\mathsf{node}(18)$, reporting the target record location at hop $H = 3$, and hosted by node $\mathsf{node_3(5)}$. The $\mathsf{m_a}$ serves also to notify $\mathsf{node}(18)$ for stopping the search process and $\mathsf{node}(18)$ will not initiate an hypothetical 4^{th} round.

When there are multiple record replicas in a network, the probability of finding the record located within a smaller number of hops increases, *i.e.* the record is close to the initiator node. The performance of ERS improves with additional record replicas. However, this reduction in the expected search energy cost comes at the expense of an increased energy cost per replication [141].

Unfortunately, ERS will only be energy-efficient when records are located on the firsts hops. The accumulation of transmissions on nodes close to the initiator node by successive searching rounds, repeats the cost of the previous attempt, since it must traverse the same links before it can reach to further nodes. In ERS the cost to increase the search area may have a considerable impact on the energy efficiency, possibly higher than a simple flooding approach [108].

ERS-TTL Other hop-limited broadcast techniques have much in common with ERS, aiming at reducing flooding expenses by using several improved strategies. Typically, tuning adjustments may be done by setting an alternative sequence for TTL values. One example is the **TTL-based ERS (ERS-TTL)** search mechanism, proposed in [56]. It assumes that network dimension (or network diameter D) is known, to decide what hop limit values it will consider. ERS-TTL uses a three-round searching scheme where TTL sequence values are $\mathsf{TTL} \in \{1, \lceil \frac{\mathsf{D}+1}{2} \rceil, \mathsf{D}\}$.

An alternative study using a dynamic programming formulation, revealed the sequences of TTL values which minimise the number of retransmissions [48]. Authors also show that for any given deterministic TTL sequence, there is a randomized version that has a lower worst-case expected search cost, concluding that generically randomized strategies are the best choice when location is unknown. However, they formulate the problem using a theoretical approach, based on dynamic programming techniques, and assuming that the probability distribution of the record location is known in advance.

In the work reported in [108], a performance analysis of ERS is provided. The analysis considers that the *initial* iteration consists of a hops ($\mathsf{TTL}_{\mathsf{initial}} = \mathsf{a}$) and, each time the search fails, it increments TTL by b, until the search exceeds the threshold value of L iterations, corresponding to a maximum TTL value of $\mathsf{TTL} = \mathsf{a} + (\mathsf{L} - 1)\mathsf{b}$. Authors consider a cost function that expresses the trade-off between latency and energy, concluding that for any given network topology, there is an optimal search threshold $\mathsf{L} \in [2 - 4]$. They also conclude that choosing $\mathsf{a} > 1 \wedge \mathsf{b} > 1$, the expected results are lower for both latency and retransmission ratio. Nevertheless, tuning a and b for an optimal solution requires topological information, which is a very demanding assumption, that we do not want to assume.

The ELastic Ring Search (ELRS) protocol proposed in [242] deviates from the optimal TTL sequences given by [48, 56, 108]. They argue that when considering frequent changes in the topology of the network, the complexity demanded by dynamic programming of the previous works requires powerful computation nodes, usually not available in MANETs or WSNs. ELRS proposes an elastic choice sequence for TTL values, resulting in a time diffused TTL-based search, with increasing and decreasing TTL values for each search. Knowing that computing optimal TTL sequences may be too demanding, the ELRS suggests that efficient sequences may be extracted using two simple heuristics: *i*) choose the same TTL value at regular intervals until the entire network is flooded, and *ii*) use a non-decreasing TTL sequence as defined by $\{\mathsf{TTL_i}, \cdots, \mathsf{TTL_j}\} \therefore \forall_{\mathsf{i<j}} : \mathsf{TTL_i} \leq \mathsf{TTL_j}$ in [242].

The two-sided expanding ring search (BiERS) proposed in [241] considers a search mechanism on which two nodes are simultaneously executing ERS for a path between each of them. All intermediate nodes that receive a query may send back a successful response to the initiator of the query, *i.e.* the path found is composed of the sub-path found by each node. BiERS explores the fact that searching cost is lower for shorter distances than for longer ones. However, they assume that the node hosting the record (destination) may start the query to the initiator, which is a strong assumption, considering this thesis assumptions.

Another approach, inspired by ant-colonies, uses agents for on-demand efficiently resource discovery [272]. The ant analogy is used to establish a route through the data stored inside HELLO messages (pheromone tracks) broadcast by the initiator node and destination. Ant-colony approaches are also used to gather information about the network state [197]. One of those examples is the Optimized-Ant protocol proposed in [230] which periodically broadcast HELLO messages from each node to all its neighbours and checks for *ant* arrivals with information about resource locations.

ERS was later explored by other discovering protocols that include AODV [211], ZRP [104], LAR [138] and is generically recognized as an energy-efficient search mechanism [48, 56]. Other energy-efficient flood-based mechanisms limit the expanding query using: *i*) geographical data (*e.g.,* Location Aided Broadcast (LAB) [253]), or *ii*) location information (*e.g., 1-hop* neighbour nodes [213]). Unfortunately, a comparison between the hop-limited and the location-limited (using GPS receivers) algorithms shows they exhibit comparable performance [54].

3.2.2 Chasing Packets

An alternative strategy to ERS for controlling a multi-hop broadcast using chasing packets was included in an algorithm known as Limited Broadcasting Algorithm (LBA) [96]. The main idea is to initiate a second broadcast, that chases the first one. The second broadcast is triggered by the initiator node once it receives a message informing a successful record discovery.

The chasing packet technique suggest that nodes add some delay before relaying (in a multi-hop network) a received search message, thus creating an opportunity for a chasing broadcast reach the original one and stop the query propagation. Chasing packets are also explored by BERS, which as far as our knowledge is concerned, its one of the best strategies, reviewed in detail on p. 79.

The technique used by LBA is based on the slot sharing policy [96]. It divides channel speed into slots, which are shared by search, acknowledgement and chasing messages. Whenever the initiator node starts a query, it broadcasts the search packet message with one-fourth channel speed, while the remaining is used for acknowledgement and chasing packet messages. When the record is found in any given node, it sends back a reply message (ack) on the second slot. Once the record has been found, the initiator node immediately broadcasts a chasing packet with a faster speed on the second slot. These chase packets terminate further propagation of the query broadcast.

To speed up the LBA stopping broadcast phase, a new protocol was proposed using marked messages and known as Limited-Hop Broadcast Algorithm (LHBA) [287]. LHBA limit the number of broadcasting hops and minimize the number of nodes that receive and rebroadcast unmarked message. In LHBA, a hop counter (k) is used to count the number of retransmissions. Every time a node receives an unmarked search message, it updates the received message by putting a mark on it, reporting the hosted records presence at $\mathsf{k} = \mathsf{H}$. The marked message continues to be rebroadcast for H hops. With that, the search message sent by the initiator node may be caught up and broadcast is forced to

Algorithm 3: Chasing packets

```
begin
    msgList ← {};
    forward ← false;
    upon event RECEIVE(msg) do
        if msg ∉ msgList then
            msgList ← msgList ∪ {msg};
            forward ← PROCESS(msg);      // Node locally decide to forward?
            if forward then
                if msg_type is search then
                    Delay ← delay(msg_hop);  // Adding delay to search message
                else
                    Delay ← 0;
                end
                SEND(msg, Delay);      // Schedule to future broadcast message
            end
        else
            DROP(msg);
        end
end
```

be terminated at the H hop. In LHBA the chasing packets are initiated by the target nodes, *i.e.* destination nodes.

Assuming an uniformed search mechanism and inline with the LBA approach, we describe the **chasing packets** mechanism with Algorithm 3. Nodes exchange messages (msg) of three distinct types: *i*) searching, *ii*) acknowledgement and *iii*) chasing. Triggered upon msg reception, node relaying will be decided locally (based on some heuristic specific of each algorithm) by calling function PROCESS(msg), as following Algorithm 3. Search messages are intentionally delayed and its retransmission is schedule to a future time by invoking function SEND(msg, Delay). Otherwise, the relay is made as fast as possible.

BERS In contrast with LBA and LHBA, the Blocking-ERS (BERS) was proposed to improve broadcast energy efficiency using a chasing packets approach, where the initiator node broadcast the query packet only once [203]. The responsibility for message rebroadcast beyond the first hop is shifted to relay nodes rather than the initiator node. The BERS protocol triggered the development of other variations of it, such as BERS* [218], BERS+ [11] and I-BERS [246]. All use the same strategy to stop a broadcast, making BERS protocol one of the most representative chasing packets search mechanisms.

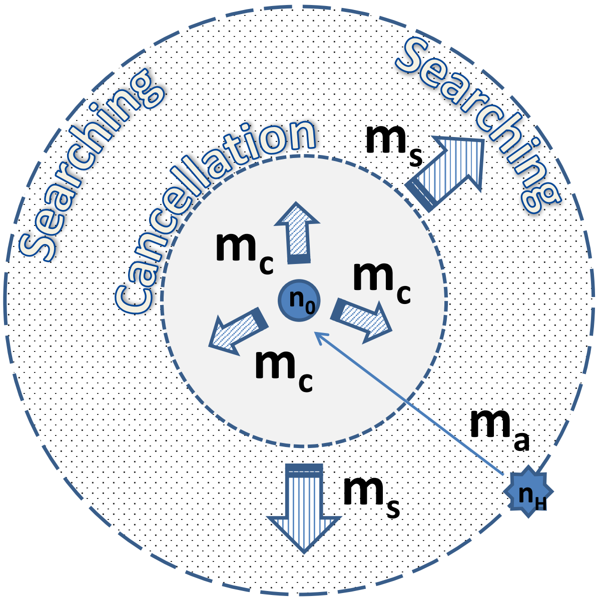


Figure 3.3: Blocking the ERS expansion (BERS), with cancellation messages sent from the initiator upon a successful record discovery.

The common characteristic of BERS search mechanism is the use of a new control message to stop the broadcasting process. BERS and BERS* search mechanisms are variations of the same **cancellation** approach, which is illustrated in Figure 3.3. The BERS search mechanism is started by $\mathsf{node}(0)$ with the transmission of a search message ($\mathsf{m_s}$). The intermediate nodes will decide to rebroadcast $\mathsf{m_s}$, after applying a parametric delay defined by $\mathsf{Delay} = 2 \times \mathsf{h} \times \mathsf{T_{slot}}$, where h is the hop distance from $\mathsf{node}(0)$ and $\mathsf{T_{slot}}$ (p. 49) is the elapse time interval to accommodate retransmissions from all nodes of the same hop. When a given node hosting the target record, *i.e.* $\mathsf{node_H(ID)}$, receives a $\mathsf{m_s}$ ($\mathsf{h} = \mathsf{H}$ is the discovery hop), it sends a point-to-point acknowledgement message ($\mathsf{m_a}$) back to the initiator, following the reverse route that is piggybacked in $\mathsf{m_s}$. Message $\mathsf{m_a}$ is relayed as fast as possible, without any deliberated delay. When $\mathsf{node}(0)$ receives $\mathsf{m_a}$, it starts the cancellation process by broadcasting the cancellation message ($\mathsf{m_c}$). The message $\mathsf{m_c}$ is also relayed without any deliberated delay by all the intermediate nodes that relayed slowly message $\mathsf{m_s}$. The ongoing broadcast terminates once $\mathsf{m_c}$ is received on nodes that are waiting the predefined delay, thus cancel their timer and any future retransmission.

Broadcast cancellation using chasing packets requires special attention to time

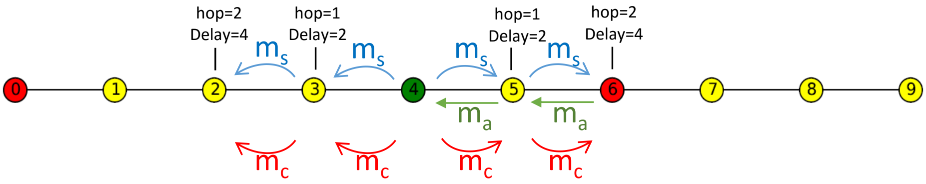


Figure 3.4: Minimal delay required to admit the opportunity to cancel an ongoing broadcast at the same hop where record discovery occurs.

restrictions. Adding delay will inevitably increase algorithm time response. However, using a too small delay, may prevent the cancellation efficiency. Depending on network topology, an optimal delay value will balance the need for low latency while minimizing the number of retransmissions, which incentives the related work consolidation as given below.

Cancellation at Same Hop Controlling the multi-hop broadcast propagation in BERS uses the chasing packets approach. Calibrating the deliberated time delay (Delay) has two major struggle issues, it should be maximum to wait for any ongoing cancellation messages and it should be minimum to reduce the impact in latency. Above all, the delay calibration should provide an effective cancellation, despite previous alerted issues.

Before any retransmission, nodes wait a minimum time depending on the distance, *i.e.* $\mathsf{Delay} \leftarrow \mathsf{f}(\mathsf{h})$, for any information about record discovery. When no information is received and upon expiring the added Delay, the algorithm assumes that no record was found and it should explore other location by rebroadcasting the received query. BERS latency performance is penalized by the creation of a time window required for allowing the cancellation mechanism to stop the query propagation in the same hop on which a target record was found.

Consider $\mathsf{t_a}$ as the time (number of synchronous transmission rounds) spent by $\mathsf{m_a}$ to be relayed from $\mathsf{node_H(ID)}$ to $\mathsf{node(0)}$, and $\mathsf{t_c}$ as the time spent by $\mathsf{m_c}$ to be propagated until hop H. Assuming that messages $\mathsf{m_a}$ and $\mathsf{m_c}$ have the same size and may be transmitted in a single round, both messages will spend the same time to reach to destination, namely: $\mathsf{t_a} = \mathsf{t_c} = \mathsf{h} \times \mathsf{T_{slot}}$. The added delay is given by $\mathsf{Delay} = 2 \times \mathsf{h} \times \mathsf{T_{slot}}$, imposed in BERS for $\mathsf{m_s}$ enables the time windows to allow the cancellation at the same hop where the record is found.

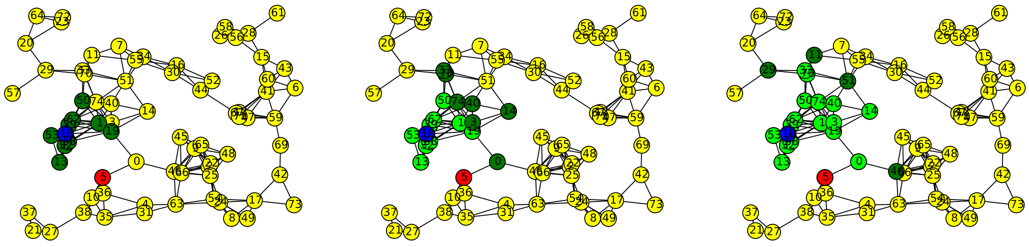
Using a simple Path topology and BERS algorithm, we describe a searching example, as represented in Figure 3.4. The search process is initiated by $\mathsf{node}(4)$ that starts the query. At first hop no record is found. The Delay time expires in nodes $\{\mathsf{node}_1(3), \mathsf{node}_1(5)\}$ and both retransmit the search message (m_s). The nodes from the second hop are $\{\mathsf{node}_2(2), \mathsf{node}_2(6)\}$. The record is found in $\mathsf{node}_2(6)$. At the same time, $\mathsf{node}_3(2)$ may not know that a record was found. The minimum time that $\mathsf{node}_3(2)$ has to wait before rebroadcasting is given by $\mathsf{Delay} = 2 \times \mathsf{h} \times \mathsf{T}_\mathsf{slot} = 4 \cdot \mathsf{T}_\mathsf{slot}$. Two slots for m_a to inform the initiator, plus two slots for m_c. Thus, $\mathsf{node}_3(2)$ receives a cancellation message before expiring the delay time.

To minimize energy consumption the cancellation should be effective at the same hop of the record discovery, avoiding additional retransmissions in subsequent hops. In this case, nodes belonging to the hop of discovery will not relay any m_s either, because they receive both m_s and m_c during the same slot time and the broadcast process terminates.

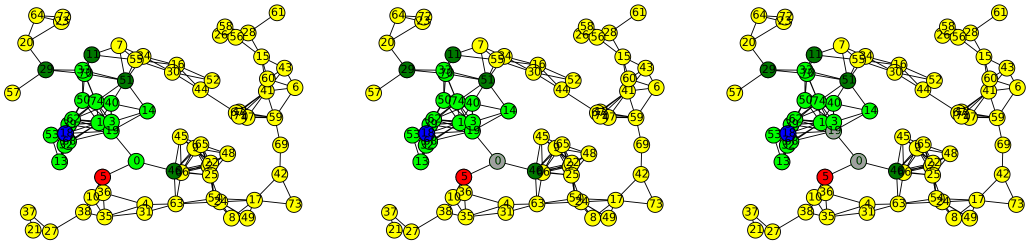
For comparing BERS with ERS and flooding, we consider the same scenario conditions, generated with the SRS (Section 2.5.3), as defined in the previous experiments (Figures 3.1−3.2). Figure 3.5 depicts relevant snapshots for BERS algorithm, illustrating its cancellation mechanism, where green nodes relayed m_s, gray nodes relayed m_a, and magenta nodes relayed m_c.

In the example depicted in Figure 3.5, the initiator $\mathsf{node}(18)$ starts the search by broadcasting m_s to its neighbours. The first snapshot corresponds to $\mathsf{hop} = 1$ where nodes after receiving m_s wait for the cancellation message before deciding to stop the rebroadcast. As no cancellation message is received they assume that the record is not found and rebroadcast m_s, corresponding to $\mathsf{hop} = 2$ of the search phase. At $\mathsf{hop} = 3$ the record is found and an acknowledgement (answer) m_a is sent back to $\mathsf{node}(18)$. Once m_a is propagated to $\mathsf{node}(18)$, it starts broadcasting the m_c (without added delay), which is propagated by the intermediate nodes as fast as possible. Message m_c is relayed only by the nodes that were active in the search phase, decrementing the hop counter embedded in the message, until it cancels the initial ongoing delayed broadcast, concluding the search.

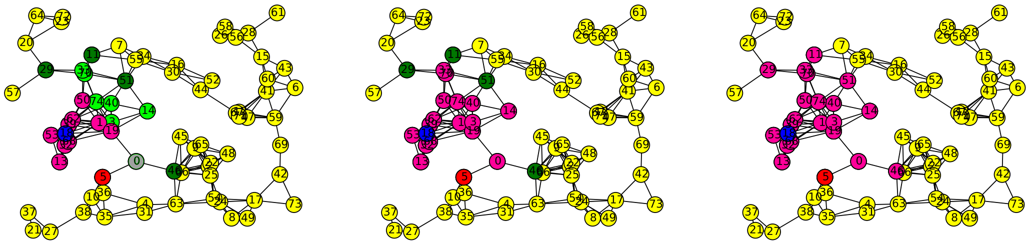
An estimation of the energy saved by BERS is given by the proportion of yellow nodes at the end of the search, as the example depicted in Figure 3.5, which were spared of any retransmission.



Searching ; 1st hop Searching ; 2nd hop Searching ; 3rd hop



ACK=1 ; 3rd hop ACK=2 ; 3rd hop ACK=3 ; 3rd hop



CancelHop=1 ; 3rd hop CancelHop=2 ; 3rd hop CancelHop=3 ; 3rd hop

Figure 3.5: Blocking Expanding Ring Search (BERS).

Aiming to improve the algorithm latency response, BERS authors proposed a new variation algorithm *enhanced-BERS* (also known as BERS*) [218]. BERS* uses the same BERS principles, differentiating only by halving the time delay applied to m_s retransmissions, thus corresponding to $Delay^\star = h \times T_{slot}$.

Cancellation at Next Hop Lowering the delay has consequences in energy efficiency, because there is no time to propagate both messages m_a and m_c within the $Delay^\star$ time.

Figure 3.6 depicts how BERS* operates on a Path topology, and compares it with example from Figure 3.4 (p. 81). Reducing delay to $Delay^\star = h \times T_{slot}$, will allow $node_2(2)$ relaying m_s for an additional hop, since the $Delay^\star$ will expire

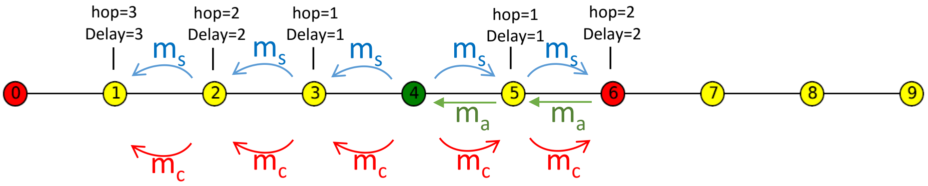


Figure 3.6: Reducing the delay to $\mathsf{Delay}^\star = \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$, reduce the latency, even though cancellation occurs in the next hop of the discovery.

before $\mathsf{node}_2(2)$ receives the cancellation message $\mathsf{m_c}$. However, during that time interval $\mathsf{m_a}$ reaches the initiator node, which starts a cancellation broadcast. Message $\mathsf{m_c}$ may be relayed to $\mathsf{node}_3(1)$ within its delay time, thus cancelling the initial broadcast in the next hop where record was found. It should be noted that $\mathsf{node}_2(6)$ hosts the target record and it does not relay $\mathsf{m_s}$ for that record, even if its $\mathsf{Delay}^\star$ has expired. Therefore $\mathsf{node}_2(6)$ does not require to relay $\mathsf{m_c}$ also.

In general, nodes belonging to hop H relay $\mathsf{m_s}$ for one more hop. The resulting behaviour is the continuation of $\mathsf{m_s}$ propagation for one additional hop given by $\mathsf{h} = \mathsf{H} + 1$. The BERS* cancellation mechanism stops one hop after the discovery, because only during the slot time $\mathsf{H}+1$ both $\mathsf{m_s}$ and $\mathsf{m_c}$ are received by the nodes at hop $\mathsf{H}+1$. Thus, resulting in a negative impact on energy efficiency since one more hop corresponds to more nodes involved in the broadcast propagation relatively to BERS. With this reduction in delay, BERS* will search faster than BERS, but with a price to pay for energy wasting.

The Combined Expanding Ring Search (CERS) [240] uses the same strategy for dynamic programming formulation introduced by [48]. It combines both hop-limited and chasing packets strategies for controlling the broadcast propagation in a way that guarantees an expected cost at least as low as the lowest of the ERS and BERS representative algorithms.

3.2.3 Discussion

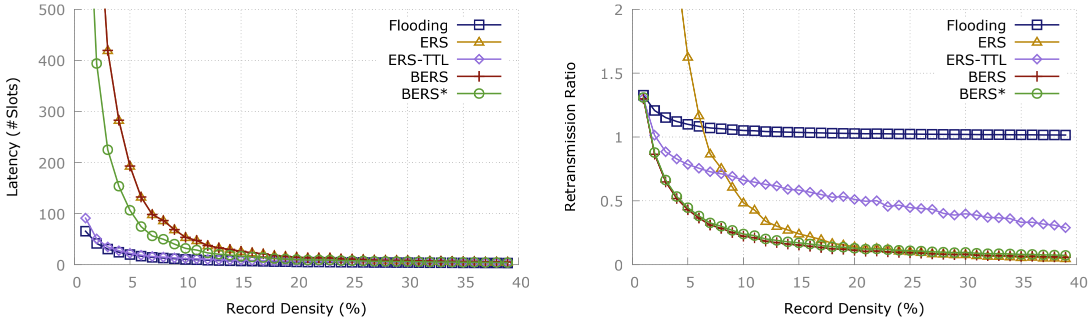
Even though cancellation has an energy cost, the chasing packets strategy avoids the ERS periodic rounds with increased TTL search messages, and so creating a power saving expectation, and thus to continue with the discussion of some experiments comparing ERS and BERS variations.

To consolidate our exploratory work, we have decided to compare the performance of distinct existing approaches that limit the broadcast propagation. We choose the following algorithms from the related work and implemented them in the SRS simulation tool:

- Hop-limited: *ERS* [125] with $\mathsf{TTL} \in \{1, 2, 3, 4, ...\}$, and *ERS-TLL* [56], with $\mathsf{TTL} \in \{1, \lceil \frac{\mathsf{D}+1}{2} \rceil, \mathsf{D}\}$.
- Chasing Packets: *BERS* [203] with $\mathsf{Delay} = 2 \times \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$, and *BERS** [218] with delay halved to $\mathsf{Delay} = \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$, corresponding to a cancellation at the next hop.

In our comparison scenarios all synthetic topologies (Path, Tree, RGG) have the same dimension, keeping the number of nodes a constant. The preliminary results consider the **average** values resulting from 2000 simulation runs for each network topology class. Each run consists of 100 nodes randomly deployed, respecting the topology model derived by limiting node connectivity by an euclidean distance limit, matching the radio signal range. Target record distribution is controlled by the *record density* parameter that regulates the number of record copies available, *i.e.* the number of records divided by the total number of nodes. The records are distributed uniformly at random through the nodes. Simulations consider a successful response to a query once the first record copy is found. The initiator node is randomly selected among the network nodes. The record copies are generated with ratios between 1% and 40%. Scenarios with record densities above 40% are not considered, because the probability of finding records nearby the initiator node is high enough, and all the algorithms (except flooding) have equivalent performance. The comparison considers two distinct metrics: *i*) Latency (time efficiency) and *ii*) Retransmission ratio (communication efficiency), considering their average values.

Figure 3.7 depicts the summary results obtained for **Path** topologies. Figure 3.7a shows that BERS* has better latency performance than BERS, and both BERS and BERS* exhibit quadratic evolutions in latency. A relevant result for low record density scenarios, where search times may cause the algorithm usefulness. Figure 3.7b shows that ERS may be very energy-inefficient for low record density, which corresponds to having the initiator node and the record on



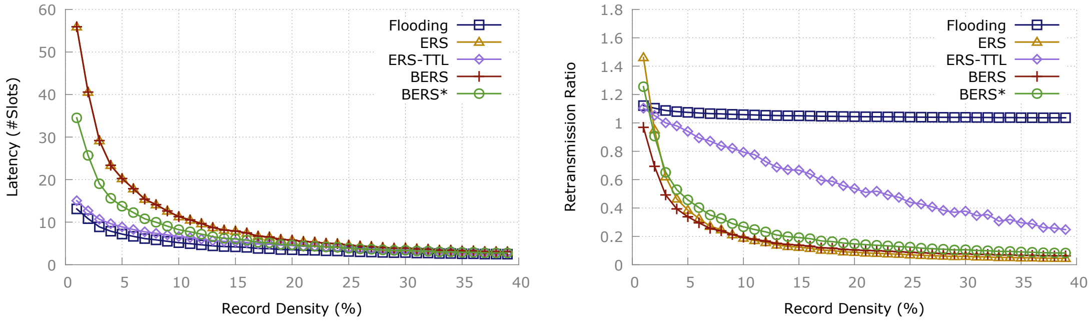
(a) Time efficiency (b) Communication overhead

Figure 3.7: Estimated latency and energy for Path topologies.

opposite sides of the path. The successive rounds of ERS have a very negative impact on communication overhead. Both BERS* and BERS have comparable energy performance. The chasing packets strategy is clearly the best mechanism to explore, when energy is a critical resource on a path graph.

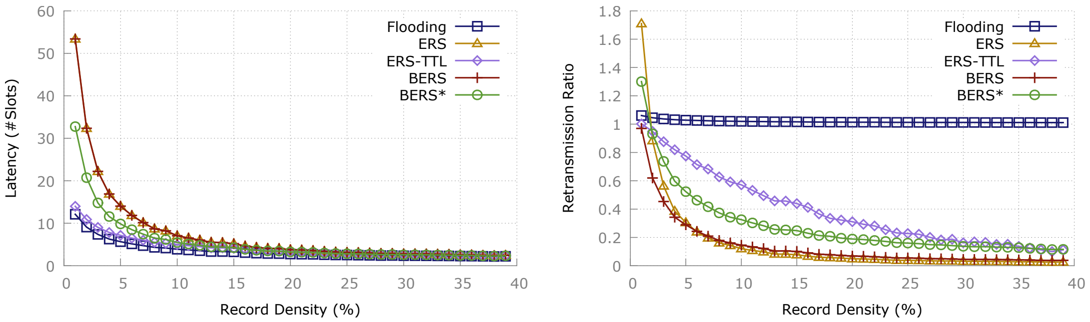
Figure 3.8 depicts the summary results obtained for **Tree** topologies. Figure 3.8a shows that the evolution of latency on Tree topologies is similar to Path topologies, with equivalent observed performance for ERS and BERS. Figure 3.8b shows that better energy performance of ERS-TTL is directly associated with the node geographical position and the Tree topology. In Tree topologies, the ERS-TTL schema is the most energy-efficient of the compared algorithms.

Figure 3.9 depicts summary results obtained for **RGG** topologies, which are generically accepted as WSNs or MANETs topology synthetic representations [133, 243]. The cost in latency is significantly smaller for chasing packets



(a) Time efficiency (b) Communication overhead

Figure 3.8: Estimated latency and energy for Tree topologies.



(a) Time efficiency

(b) Communication overhead

Figure 3.9: Estimated latency and energy for Random Geometric topologies.

algorithms, as may be observed in Figure 3.9a, except when compared with the ERS-TTL optimization. Figure 3.9b shows that chasing packets algorithms also have good energy performance.

Figures 3.7a – 3.9a show that flooding has achieved the best latency performance. Penalization in latency is the price to pay (by all the tested algorithms) to obtain better energy performances. It should be noted that ERS and BERS exhibit similar latency and therefore the results are overlapped in Figures 3.7a – 3.9a.

Figures 3.7b – 3.9b are estimations of the algorithms energy efficiency. The limited broadcast algorithms have a very significant reduction of the retransmission ratio, revealing good characteristics for the demanding WUSN requirements and difficulties for device battery replacements.

Results show that chasing packets approaches may improve the performance of hop-limited schemes, when the available record rate lies between 1% and 15%. It was also observed that, when the copies are farther than $\lceil(D+1)/2\rceil$ hops distance from the initiator, the chasing packets has better performance than ERS, although this property is not always visible when the copies are randomly distributed.

Choosing randomly the query initiator node or start the query always at the same node, *i.e.* $\mathsf{node}(0)$, it was observed that there were no changes in the algorithms average results, regardless of the option.

Not surprisingly, it may be observed that, as the number of record copies increases, the probability of finding a record nearby the initiator also increases, and ERS has better latency performance than BERS. It should be noted that, when records are available in the first hop, ERS has better energy performance than cancellation-based algorithms, since no successive rounds are required and ERS does not have any cancellation cost.

One of the main advantages of chasing packets over hop-limited algorithms is that, although both may lead to more than one broadcast, chasing mechanisms avoid using multiple secondary rounds to keep expanding the covered region. Knowing the network size, a requirement for ERS, may be a very hard task with a lot of battery power wasted. Solutions based on chasing packets mechanisms do not require previous topology information, thus also being more adequate for handling with topology dynamics.

Previous observations suggest that there are reasons to believe that controlled broadcast can improve the energy performance of flooding implementations, thus helping to extend the lifetime of battery power devices in multi-hop networks. Due to the complexity of controlling the broadcast propagation and despite the existence of some solutions, there is still an opportunity to contribute with new approaches, as the following one.

3.3 BCIR: Algorithm Description

In contrast with previous chasing packets approaches (BERS and BERS*), we propose a new mechanism designated as **Broadcast Cancellation Initiated at Resource (BCIR)** [160]. With the new BCIR algorithm, we aim at improving the performance of chasing packets search mechanisms, by exploring an alternative approach for this kind of protocols. Having explored the impact on latency caused by BERS and BERS* and after identifying some of its constraints, we introduce a new approach for the cancellation mechanism designed to improve the energy performance.

BCIR main idea is that cancellation starts as soon as possible, as depicted in Figure 3.10. There is no need for an independent acknowledgement message ($\mathsf{m_a}$), as used in BERS and BERS*. Instead in BCIR, the cancellation message ($\mathsf{m_c}$) is initially broadcast by the node hosting the record $\mathsf{node_H(ID)}$, avoiding the waiting time required for $\mathsf{m_a}$ rebroadcast and propagation. When $\mathsf{m_c}$ reaches the initiator node, it reports the target record location.

In BCIR no deliberate delay is considered for $\mathsf{m_c}$ dissemination which starts at

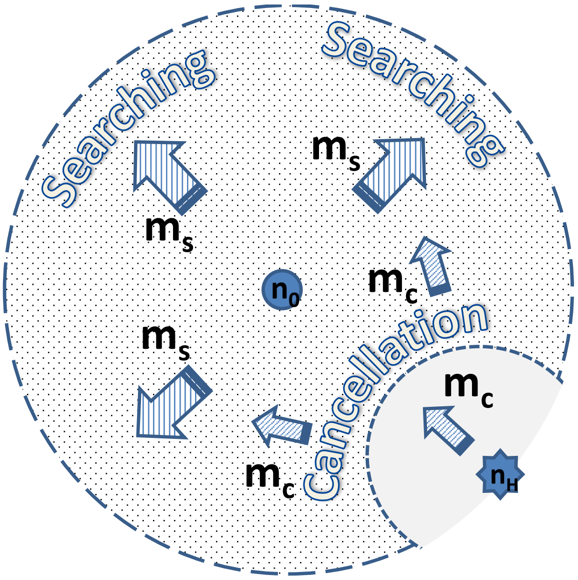


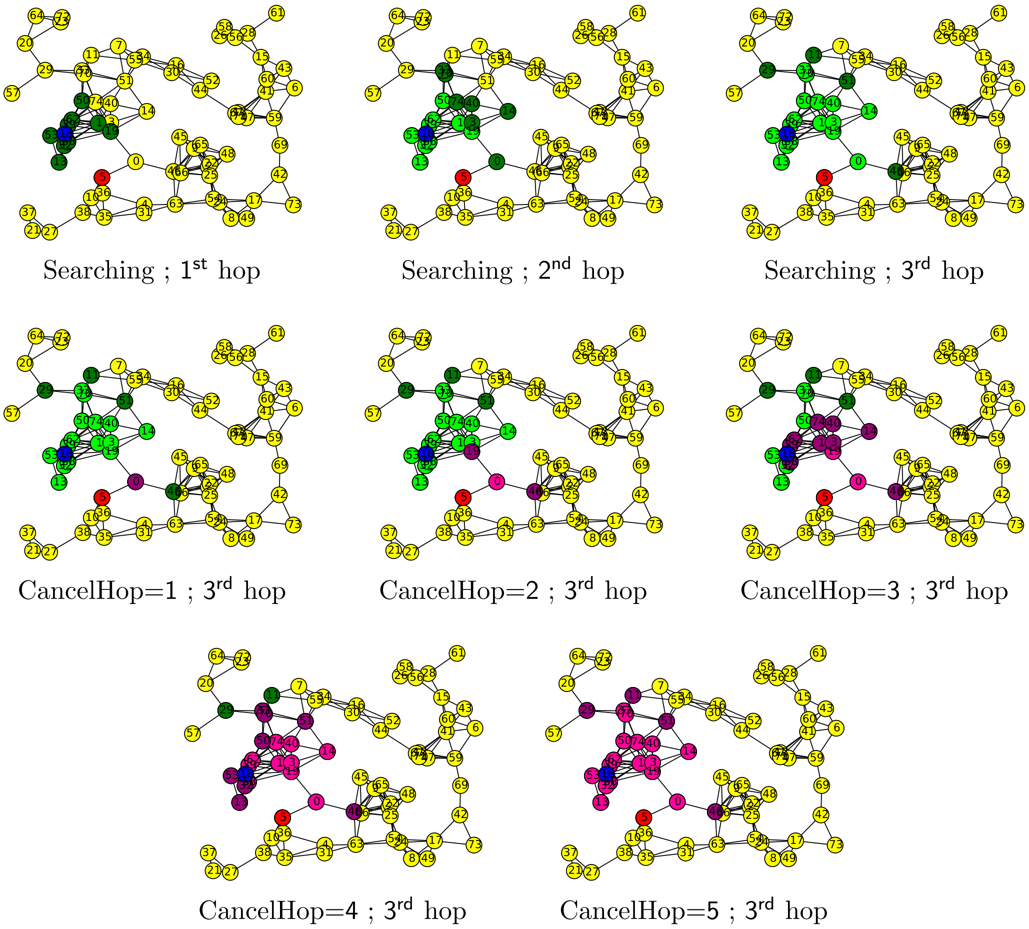
Figure 3.10: Cancellation mechanism starting as soon as possible.

the search *ring* periphery, as depicted in Figure 3.10. Analogously to BERS, two alternatives are also considered for choosing the $\mathsf{m_s}$ delay, which resulted in two variants, BCIR and BCIR*, exploring the trade-off between latency and energy consumption.

BCIR experimental tests will consider the same scenarios as the ones set in previous examples (Figures 3.1 and 3.5), ensuring that all search mechanisms run in the same topologies generated by SRS simulation tool (Section 2.5.3). Thus, making possible to compare distinct search mechanisms running in the same conditions. BCIR creates an expectation in the reduction of the energy consumption, due the absence of $\mathsf{m_a}$ retransmissions.

To explain how BCIR conducts a query propagation, Figure 3.11 depicts several snapshots representing node states for each hop over time. The search phase (represented by green nodes) is equal to BERS, as ilustrated in Figure 3.5 (p. 83), and its cancellation mechanism (represented by magenta nodes) starts immediately and at the same hop of the record discovery.

In the example depicted in Figure 3.11 the record is hosted at node $\mathsf{node_H(5)}$. The initiator node $\mathsf{node(18)}$ starts the query by broadcasting $\mathsf{m_s}$. Nodes receiving $\mathsf{m_s}$ (not owning the target record) wait some time (Delay) before rebroadcast $\mathsf{m_s}$ in a multi-hop network. When $\mathsf{m_s}$ reaches $\mathsf{node_3(5)}$ the record location is found and BCIR unchains cancellation (snapshot label CancelHop=$\mathsf{1}$; $\mathsf{3^{rd}}$ hop)

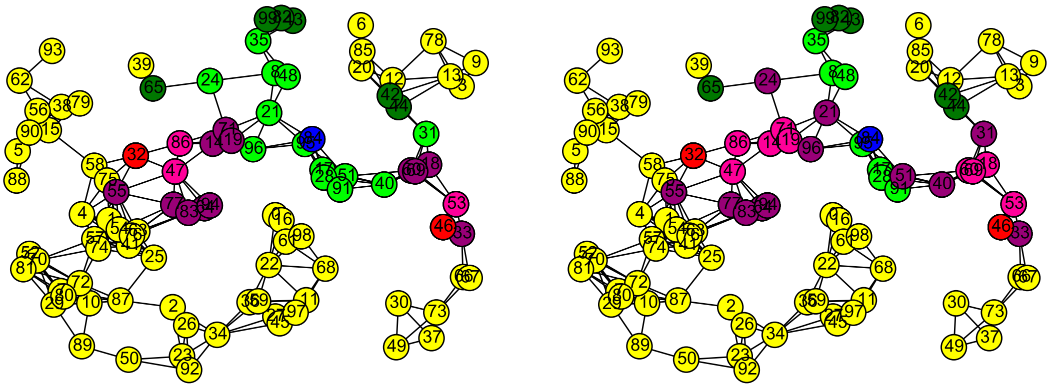


Searching ; 1st hop
Searching ; 2nd hop
Searching ; 3rd hop
CancelHop=1 ; 3rd hop
CancelHop=2 ; 3rd hop
CancelHop=3 ; 3rd hop
CancelHop=4 ; 3rd hop
CancelHop=5 ; 3rd hop

Figure 3.11: Broadcast Cancellation Initiated on Resource (BCIR).

by broadcasting $\mathsf{m_c}$, that contains record location information. Nodes receiving cancellation messages $\mathsf{m_c}$ are allowed to rebroadcasting it ($\mathsf{m_c}$), only if they repeated $\mathsf{m_s}$ from the same query (represented by magenta nodes). Messages $\mathsf{m_c}$ are relayed as fast as possible, until they catch the ongoing query broadcast, thus ending up the search query.

When multiple record copies are discovered before cancellation has time to disseminate its information, for example during the same hop (H), BCIR broadcast multiple cancellation messages initiated from distinct nodes (hosting the discovered records). Figure 3.12 depicts a sample of BCIR cancellation mechanism, that is concurrently initiated by two nodes hosting the target record copies and belonging to the same hop, *i.e.* $\mathsf{node_H(ID)} \in \{\mathsf{node_H(32)}, \mathsf{node_H(46)}\}$. Fig-



(a) 2nd cancellation hop

(b) 3rd cancellation hop

Figure 3.12: BCIR with multiple records arising.

ure 3.12 shows that BCIR can manage multiple cancellation *rings* which are in a expanding progress, that stops when they completely cancel the initial query.

3.3.1 Analytical Modelling

To proceed with BCIR analytical modelling, it should be noted that, the deliberate delay introduced during the search phase has a significant impact on BCIR performance. The analytical modelling ensures similar assumptions to the previous experiment scenarios and adopting the BERS approach, for a later fair comparison between them. Considering a synchronous model, from all possible variations, BCIR proposes two $\mathsf{Delay}_{\mathsf{proto}}$ alternatives, as described in Algorithm 3 (p. 79, and designated as:

BCIR To reduce the number of retransmissions, BCIR establishes a minimum delay of $\mathsf{Delay}_{\mathsf{BCIR}} = 2 \times \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$, enough for accommodating the cancellation propagation. Thus, impeding the search query spreading to next hop. The $\mathsf{Delay}_{\mathsf{BCIR}}$ enables BCIR to **stop broadcast within same hop** (p. 81), where the target record is found (H). Notice that, the added delay $\mathsf{Delay}_{\mathsf{BCIR}}$ is a function of the hop distance (h).

BCIR* To improve the algorithm latency, BCIR* reduces search added delay to half the one of BCIR. In BCIR* delay is given by $\mathsf{Delay}_{\mathsf{BCIR}^\star} = \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$. Thus, to control unnecessary retransmissions the $\mathsf{Delay}_{\mathsf{BCIR}^\star}$ ensures that BCIR* can **stop broadcast within next hop** (p. 83), where the target record is found ($\mathsf{H}+1$). Nevertheless, the number of contacted nodes is larger (a consequence of an additional hop), with a negative impact on energy efficiency.

Reducing $\mathsf{Delay_{proto}}$ (waiting time for $\mathsf{m_s}$ rebroadcast) significantly may preclude the cancellation process. The cancellation always starts after the search phase and may even never recover when the $\mathsf{Delay_{proto}}$ parametrization is set too high. Theoretical characterization of new cancellation mechanisms BCIR and BCIR*, proceeds with an analytical modelling based on the same approach described in BERS [218]. The analytical modelling may be used to predict the algorithms performance and compare it with other competitive algorithms, using the following time and energy efficiency metrics, which are influenced by the $\mathsf{Delay_{proto}}$ algorithm parametrization.

3.3.1.1 Time Efficiency

Time efficiency for search mechanisms may be quantified in terms of **latency**, *i.e.* $\mathsf{L_{proto}}$. Latency (p. 26) may be measured as the elapsed time between the start of the broadcasting search message ($\mathsf{m_s}$) by the initiator node, *i.e.* $\mathsf{node(0)}$, and the time at which $\mathsf{node(0)}$ receives a successful answer ($\mathsf{m_a}$ on BERS ; $\mathsf{m_c}$ on BCIR) to the initial query. Latency ($\mathsf{L_{proto}}$) has two components: *i*) search time ($\mathsf{t_{s_{proto}}}$) and *ii*) answer time ($\mathsf{t_a}$), *i.e.* $\mathsf{L_{proto}} = \mathsf{t_{s_{proto}}} + \mathsf{t_a}$.

We assume a multi-hop network scenario where latency may be expressed as a function of hop distance (h), given that radio signal propagation delay is negligible when compared to the intentionally added delay ($\mathsf{Delay_{proto}}$) introduced by the nodes. Assuming that acknowledgement messages ($\mathsf{m_a}$) for algorithms BERS, BERS* and cancellation messages ($\mathsf{m_c}$) for algorithms BCIR and BCIR* are propagated as quickly as possible (no added delay), *i.e.* the elapse time for $\mathsf{node(0)}$ to receive an answer, *i.e.* $\mathsf{t_a}$, is given by Eq.(3.1):

$$\mathsf{t_a} = \mathsf{h} \times \mathsf{T_{slot}} \tag{3.1}$$

Searching time $\mathsf{t_{s_{BCIR}}}$ component for BERS and BCIR is driven by node relaying events. Nodes at h hops distant from the initiator node ($\mathsf{node_h(ID)}$), delay message retransmissions by $\mathsf{Delay} = 2 \times \mathsf{h} \times \mathsf{T_{slot}}$.

The elapsed time interval between two consecutive hops has two components:

i) the waiting time at the previous hop ($\mathsf{Delay}_{\mathsf{h}-1}$), and *ii*) the time slot ($\mathsf{T}_{\mathsf{slot}}$) capable to accommodate all retransmissions concerning a particular hop, which is given by Eq.(3.2):

$$\mathsf{Delay}_{\mathsf{h}-1} + \mathsf{T}_{\mathsf{slot}} = 2 \times (\mathsf{h} - 1) \times \mathsf{T}_{\mathsf{slot}} + \mathsf{T}_{\mathsf{slot}} = (2\mathsf{h} - 1) \times \mathsf{T}_{\mathsf{slot}} \tag{3.2}$$

Generalizing, to an arbitrary number of n hops, the time instant when m_{s} reaches a node belonging to hop h, is given by Eq.(3.3):

$$\mathsf{t}_{\mathsf{s}_{\mathsf{BCIR}}} = \sum_{\mathsf{n}=1}^{\mathsf{h}} (2\mathsf{n} - 1) \times \mathsf{T}_{\mathsf{slot}} = \mathsf{h}^2 \times \mathsf{T}_{\mathsf{slot}} \tag{3.3}$$

Latency ($\mathsf{L}_{\mathsf{proto}}$) for BERS and BCIR mechanisms is the same, considering the two major components from Eq.(3.3) and Eq.(3.1), $\mathsf{L}_{\mathsf{proto}}$ is given by Eq.(3.4):

$$\mathsf{L}_{\mathsf{BERS}} = \mathsf{t}_{\mathsf{s}_{\mathsf{BCIR}}} + \mathsf{t}_{\mathsf{a}} = (\mathsf{h}^2 + \mathsf{h}) \times \mathsf{T}_{\mathsf{slot}} = \mathsf{L}_{\mathsf{BCIR}} \tag{3.4}$$

Next, we address BERS* and BCIR* latency $\mathsf{L}_{\mathsf{proto}}$ changes. For nodes $\mathsf{node}_{\mathsf{h}}(\mathsf{ID})$ (belonging to hop h) the search time component follows the added delay reduction, that is halved and given by $\mathsf{Delay} = \mathsf{h} \times \mathsf{T}_{\mathsf{slot}}$. Thus, the elapsed time interval between two consecutive hops is reduced and given by Eq.(3.5):

$$\mathsf{Delay}_{\mathsf{h}-1} + \mathsf{T}_{\mathsf{slot}} = (\mathsf{h} - 1) \times \mathsf{T}_{\mathsf{slot}} + \mathsf{T}_{\mathsf{slot}} = \mathsf{h} \times \mathsf{T}_{\mathsf{slot}} \tag{3.5}$$

Generalizing, to an arbitrary number of n hops, the time instant when m_{s} reaches a node belonging to hop h, is given by Eq.(3.6):

$$\mathsf{t}_{\mathsf{s}_{\mathsf{BCIR}^\star}} = \sum_{\mathsf{n}=1}^{\mathsf{h}} \mathsf{n} \times \mathsf{T}_{\mathsf{slot}} = \frac{1 + \mathsf{h}}{2} \times \mathsf{h} \times \mathsf{T}_{\mathsf{slot}} = \frac{\mathsf{h}^2 + \mathsf{h}}{2} \times \mathsf{T}_{\mathsf{slot}} \tag{3.6}$$

Latency ($\mathsf{L}_{\mathsf{proto}}$) for BERS* and BCIR* mechanisms is the same, considering the two major components from Eq.(3.6) and Eq.(3.1), $\mathsf{L}_{\mathsf{proto}}$ is given by Eq.(3.7):

$$\mathsf{L}_{\mathsf{BERS}^\star} = \mathsf{t}_{\mathsf{s}_{\mathsf{BCIR}^\star}} + \mathsf{t}_{\mathsf{a}} = \frac{\mathsf{h}^2 + 3\mathsf{h}}{2} \times \mathsf{T}_{\mathsf{slot}} = \mathsf{L}_{\mathsf{BCIR}^\star} \tag{3.7}$$

Figure 3.13 compares the analytical estimated latency for several cancellation-based search mechanisms, with respect to the distance between the initiator $\mathsf{node}(0)$ and the node hosting the record $\mathsf{node}_{\mathsf{H}}(\mathsf{ID})$. $\mathsf{H} = 0$ is the case when

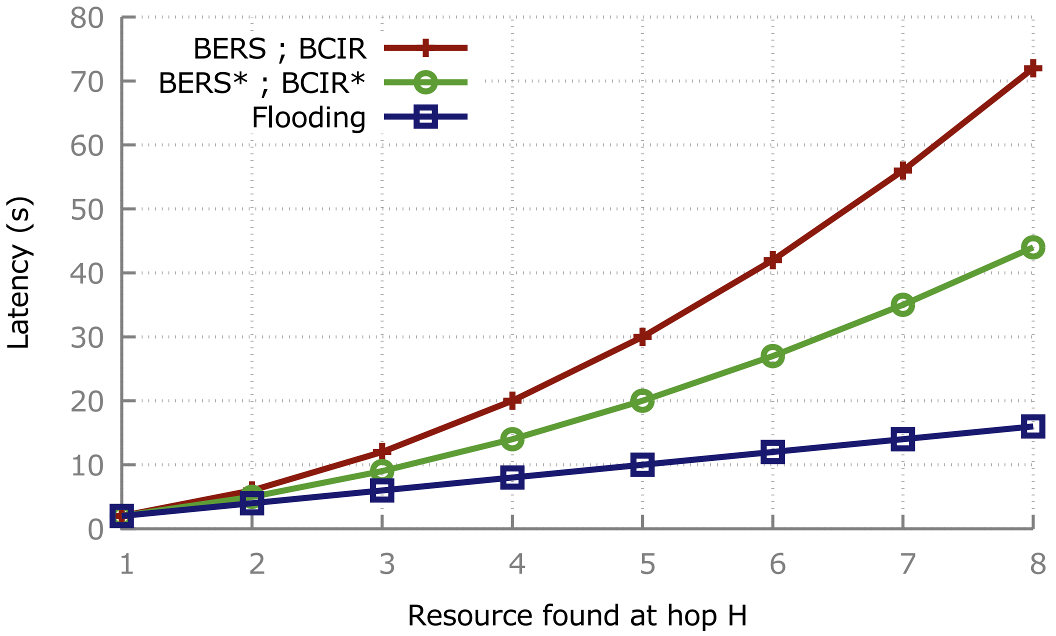


Figure 3.13: Latency ($\mathsf{L}_{\mathsf{proto}}$) analytical estimation.

$\mathsf{node}(0)$ hosts the target record and no query is needed to run, thus graph from Figure 3.13 starts at $\mathsf{H} = 1$. As H increases, more nodes will be contacted to find the record, *i.e.* a record may be found farther away from $\mathsf{node}(0)$.

The BERS* and BCIR* mechanisms improve their latency gains when compared with BERS and BCIR, as the number of hops to find a given record increases. Therefore, it is expected a significant latency improvement introduced by BCIR* for larger networks with low record density, when compared with BCIR.

3.3.1.2 Energy Efficiency

Energy efficiency for search mechanisms may be estimated in terms of **retransmission ratio** (p. 27), and measured by counting the number of retransmissions made until the broadcast is completely cancelled, *i.e.* $\#\mathsf{Retransmissions}$. Assuming that energy spent in each node is evenly distributed across all network nodes (N) and each retransmission consumes the same amount of energy, the algorithm energy efficiency is roughly proportional to the retransmission ratio, *i.e.* $R_{proto} = \frac{\#\mathsf{Retransmissions}}{\mathsf{N}}$.

Attending that WSNs are deployed to gather information on an geographical area and wireless nodes have a limited communication range, the size of the network, *i.e.* the number of devices to deploy N, depends on the size of the coverage area. A relationship represented as spatial **node density** given by $\sigma = \frac{N}{Area}$, which is the number of nodes per area unit.

Assuming an omnidirectional propagation in a multi-hop network, the radio coverage area corresponds to a circle of radius h. The number of nodes relaying messages during the query dissemination inside that area ($\mathsf{N_h}$), are estimated by Eq.(3.8):

$$\mathsf{N_h} = \sigma \times \pi \times \mathsf{h}^2 \tag{3.8}$$

Considering that the number of retransmissions is proportional to h^2, it is possible to devise an expression for the retransmission ratio related with the message dissemination process performed by the nodes within *ring* h, given by Eq.(3.9):

$$\mathsf{R_{proto}} = \frac{\mathsf{N_h}}{\mathsf{N}} = \frac{\sigma\pi\mathsf{h}^2}{\mathsf{N}} \tag{3.9}$$

Analytical approach for $\mathsf{R_{proto}}$ assumes the existence of a target record that may be found upon h hops, and considers three retransmission ratio components: *i*) search dissemination ratio ($\mathsf{r_s}$), *ii*) point-to-point acknowledgement ratio ($\mathsf{r_a}$) and *iii*) cancellation dissemination ratio ($\mathsf{r_c}$), given by Eq.(3.10):

$$\mathsf{R_{proto}} = \mathsf{r_s} + \mathsf{r_r} + \mathsf{r_c} \tag{3.10}$$

In BERS, the search dissemination is roughly limited to circular area with radius h, $\mathsf{m_a}$ is point-to-point retransmitted and the cancellation area overlaps the searched one, thus $\mathsf{R_{BERS}}$ is given by Eq.(3.11):

$$\mathsf{R_{BERS}} = \mathsf{r_s} + \mathsf{r_r} + \mathsf{r_c} = \frac{\sigma\pi\mathsf{h}^2}{\mathsf{N}} + \frac{\mathsf{h}}{\mathsf{N}} + \frac{\sigma\pi\mathsf{h}^2}{\mathsf{N}} = \frac{2\sigma\pi}{\mathsf{N}}\mathsf{h}^2 + \frac{1}{\mathsf{N}}\mathsf{h} \tag{3.11}$$

In BERS*, its cancellation mechanism admit one additional hop before stopping the search dissemination. The explored (search and cancellation) area is roughly limited to circular area with radius $\mathsf{h}+1$, nevertheless $\mathsf{m_a}$ is point-to-point retransmitted for h hops, thus $\mathsf{R_{BERS^\star}}$ is given by Eq.(3.12):

$$\mathsf{R_{BERS^\star}} = \mathsf{r_s} + \mathsf{r_r} + \mathsf{r_c} = \frac{2\sigma\pi}{\mathsf{N}}(\mathsf{h}+1)^2 + \frac{1}{\mathsf{N}}\mathsf{h} \tag{3.12}$$

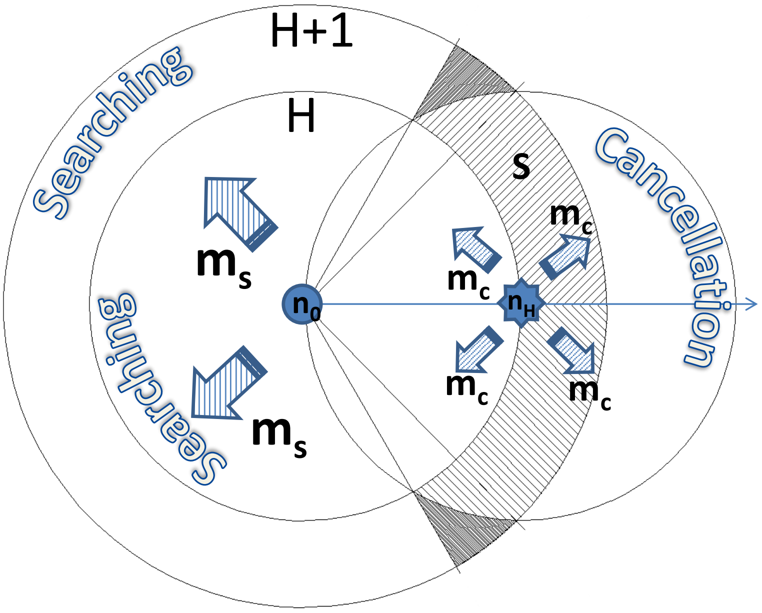


Figure 3.14: Early cancellation effect produced by BCIR⋆ search mechanism.

In BCIR, retransmission ratio is similar to BERS, except that it dispenses $\mathsf{m_a}$ retransmissions, *i.e.* $\mathsf{r_a} = 0$, thus $\mathsf{R_{BCIR}}$ is given by Eq.(3.13):

$$\mathsf{R_{BCIR}} = \mathsf{r_s} + \mathsf{r_c} = \frac{\sigma\pi \mathsf{h}^2}{\mathsf{N}} + \frac{\sigma\pi \mathsf{h}^2}{\mathsf{N}} = \frac{2\sigma\pi}{\mathsf{N}}\mathsf{h}^2 \tag{3.13}$$

In BCIR⋆, there are nodes belonging to searched hops that will not retransmit any query, which is a significant difference from previous algorithms and must be considered for analytical modelling. Upon a record discovery at a given node $\mathsf{node_H(ID)}$, BCIR⋆ initiates the cancellation mechanism, which will notify all $\mathsf{node_H(ID)}$ neighbour nodes. Nodes receiving a message $\mathsf{m_c}$ before the moment of retransmission of message $\mathsf{m_s}$, will not relay neither message $\mathsf{m_s}$ nor $\mathsf{m_c}$.

Figure 3.14 depicts a representation of BCIR⋆ search mechanism, showing the expected influence of cancellation initiated on resource in the nearby nodes. This early cancellation effect near $\mathsf{node_H(ID)}$, cannot be neglected when estimating the expected number of retransmissions.

The BCIR⋆ retransmission ratio (R) will not consider nodes belonging to region S depicted in Figure 3.14, that will not relay neither message $\mathsf{m_s}$ nor $\mathsf{m_c}$, because they will have already received at least one cancellation message. The retransmissions reduction at region S corresponds to an energy gain.

To simplify the analytical expression, instead of using the area of region S for calculation, as depicted in Figure 3.14, we consider the a smaller area region represented in cylindrical coordinates as given by Eq.(3.14), which reveals a linear dependency between the cancellation area (S) and the distance (H).

$$\mathsf{S} \gtrsim \int_{\mathsf{H}}^{\mathsf{H}+1} \int_{-\pi/3}^{\pi/3} \mathsf{r}\,\mathsf{dr}\,\mathsf{d}\phi = \frac{\pi}{3}(2\mathsf{H}+1) \tag{3.14}$$

The number of nodes corresponding to the cancellation area may be estimated by Eq.(3.15):

$$\mathsf{N_S} = \frac{1}{3}(\mathsf{N_{H+1}} - \mathsf{N_H}) = \frac{\sigma\pi}{3}(2\mathsf{H}+1) \simeq \mathsf{N_S} \tag{3.15}$$

Our analytical approach to estimate $\mathsf{R_{BCIR^\star}}$ is based on $\mathsf{R_{BCIR}}$ given in Eq.(3.13). In BCIR* the search explores a maximum of $\mathsf{H}+1$ hops and $\mathsf{R_{BCIR_{H+1}}} = \frac{2\sigma\pi}{\mathsf{N}}(\mathsf{H}+1)^2$. As $\mathsf{N_S}$ does not relay neither $\mathsf{m_s}$ nor $\mathsf{m_c}$ message. Therefore, it is necessary to remove these $2\mathsf{N_S}$ retransmissions to estimate $\mathsf{R_{BCIR^\star}}$, as given by Eq.(3.16):

$$\mathsf{R_{BCIR^\star}} \simeq \mathsf{R_{BCIR_{H+1}}} - 2\left(\frac{\mathsf{N_S}}{\mathsf{N}}\right) \tag{3.16}$$

Using Eq. (3.16) and (3.14), $\mathsf{R_{BCIR^\star}}$ estimation is given by Eq.(3.17):

$$\mathsf{R_{BCIR^\star}} \simeq \frac{2\sigma\pi}{3\mathsf{N}}(3\mathsf{H}^2 + 4\mathsf{H} + 2) \tag{3.17}$$

The expression for estimating the **energy gain** Δ_R of BCIR* over the BERS* is a result of combining Eqs.(3.12) and (3.17). Such gain Δ_R, quantify the difference between their respective retransmission ratio, as given by Eq.(3.18):

$$\Delta_\mathsf{R} = \mathsf{R_{BERS^\star}} - \mathsf{R_{BCIR^\star}} \lesssim \frac{2\sigma\pi}{3\mathsf{N}}(2\mathsf{H}+1) + \frac{1}{\mathsf{N}}\mathsf{H} \tag{3.18}$$

Searching by flooding has a trivial $\mathsf{R_{FLOOD}} = 1 + \mathsf{H}/\mathsf{N}$ analytical model, as revealed in practice by the carried out experiments (Section 2.5.4).

Measuring the retransmission ratio Both BERS* and BCIR* search mechanisms were developed to reduce latency. However, latency reduction gives rise for more energy consuming algorithms, as Figure 3.15 shows when comparing BERS* and BCIR* with BERS and BCIR.

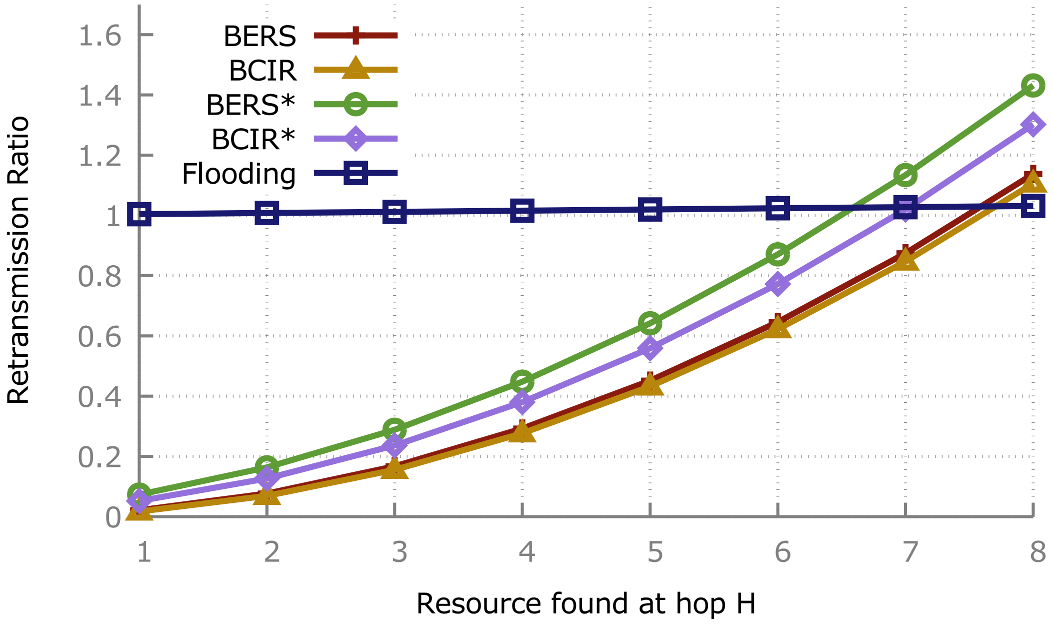


Figure 3.15: Retransmission ratio (R_{proto}) analytical estimation.

Figure 3.15 depicts analytical results, comparing the flooding baseline with R_{proto} for BERS, BERS*, BCIR and BCIR* search mechanisms, using Eqs.(3.11) – (3.13) and (3.17). The gain (difference) of energy between the BCIR* and BERS* is greater that the expected gain between BCIR and BERS, as suggested by results depicted in Figure 3.15.

Figure 3.15 shows that the BCIR* retransmission ratio is inferior to BERS*, suggesting that BCIR* will save energy over BERS*, even though both have the same latency performance, as results show in Figure 3.13 (p. 94).

3.3.2 On the Limits of Cancellation

An orthogonal issue to the cancellation problem is trying to determine the extent to which it is useful to apply a cancellation mechanism for saving energy. It is always advantageous to use a cancellation mechanism, when considering a theoretical scenario application for an infinite network size. However, for finite-dimensional networks, and assuming that the record is found near to the network limit, it is more energy-efficient to let the flooding end by itself, than to broadcast cancellation messages that will eventually cancel the ongoing search.

Any flooding broadcast will stop by itself thanks to the principle that no message is retransmitted twice by the same node. In limited-sized topologies there is a **threshold** ($\mathsf{H_{th}}$) on which cancellation mechanisms waste less energy than in simple flooding. The threshold will be found when the searched record is far away from the initiator and near to the network farthest limit. In these cases, the energy spent by a cancellation mechanism will be greater than the energy consumed by flooding (left to end by itself). The following modelling is a theoretical prediction of such threshold.

Considering $\mathsf{H_{max}}$ as the maximum number of hops, which depends on the network size and topology. Assuming that $\mathsf{H_{max}}$ is known and radio signal propagation is omnidirectional, the multi-hop network covered area may be roughly estimated as the circle area, given by $\pi \cdot \mathsf{H^2_{max}}$. Thus, the energy cost to contact all the nodes inside the circle will be proportional to $\mathsf{H^2_{max}}$.

The area covered by BCIR is two times the explored area (search plus cancellation), assuming that node density is uniform (an assumption that may be accepted for random topologies), the relationship between both areas may be used to reveal the $\mathsf{H_{th}}$ threshold limit, as given by Eq.(3.19):

$$2 \cdot \pi \cdot \mathsf{H^2_{th}} \leq \pi \cdot \mathsf{H_{max}}^2 \Rightarrow \mathsf{H_{th}} \leq \frac{\mathsf{H_{max}}}{\sqrt{2}} \tag{3.19}$$

For any limited-sized network there is a threshold distance, *i.e.* $\mathsf{H_{th}}$, from which it is cheaper to continue the ongoing flooding, by abandoning the cancellation mechanism. Eq.(3.19) suggests that cancellation mechanisms looses their energy efficiency for simple flooding, when records are located beyond the $\mathsf{H_{th}}$ threshold.

Figure 3.15 shows that BCIR is the last algorithm to cross the flooding baseline, suggesting that it is preferable to use BCIR when searching for farthest records. As predicted by $\mathsf{H_{th}}$ threshold limit in Eq.(3.19), flooding for $\mathsf{H} \geq 8$ has the lowest retransmission ratio from all the compared algorithms, as depicted in Figure 3.15, due to the finite network size scenarios used for experimentations.

3.4 Evaluation (BCIR)

To the best of our knowledge, BERS is the most energy-efficient uninformed search mechanism [218]. Experiments presented (Section 3.2.3) confirms BERS as the best search mechanism strategy, among the implemented algorithms, to reduce de average number of retransmissions. The methodology for BCIR evaluation compares BCIR with the most competitive search mechanisms, currently the BERS variations, both using chasing packets limited broadcast approaches.

Without compromising the generality of the evaluation, it was decided to manage a single record type (*e.g.,* RFID shovel). In our experiments, the record density parameter controls the number of record replicas distributed along the network and each node hosts only a single record copy. Target records are randomly allocated to nodes, avoiding clustering effects in sub-network areas [25]. From a practical point of view, it is relevant to evaluate record density influence in the search mechanisms based on chasing packets. It should be noted that, evaluation scenarios ignore memory and processing constraints, because they are irrelevant for energy and latency analysis.

Using simulation techniques, several experiments were performed comparing chasing packets algorithm variations (BERS, BERS*, BCIR and BCIR*) using the following metrics: *i*) latency and *ii*) retransmission ratio. Both must be considered as a function of record density that is specified for each scenario.

Although the simulation engine enables us to perform experimentations in different scenarios, using distinct types of topologies, on continuation, evaluation will be focus on using synthetic network topologies represented by Random Geometric Graphs (RGGs). We note, however, cancellation approach has a wider area of application, such as overlay networks over the Internet, which require other topology models that be considered relevant for that settings.

3.4.1 Experiment Setup

For comparing purposes with alternative algorithms (BERS and BERS*), the new set of algorithms (BCIR and BCIR*) were implemented and included in the SRS simulation tool (Section 2.5.3).

Node deployment is compliant with **Random Geometric Graphs (RGG)**, which synthetically represent MANETs topologies [243]. RGG connects two nodes if and only if the distance between them lies within a given range, *i.e.* smaller than a certain threshold radius r. Range limiting is consistent with the fact that radio signal decays as the distance between the sender and the receiver increases, due to a combination of attenuation and fading effects.

For evaluation purposes, experimental tests generated $\simeq \mathsf{170\,000}$ RGG net-

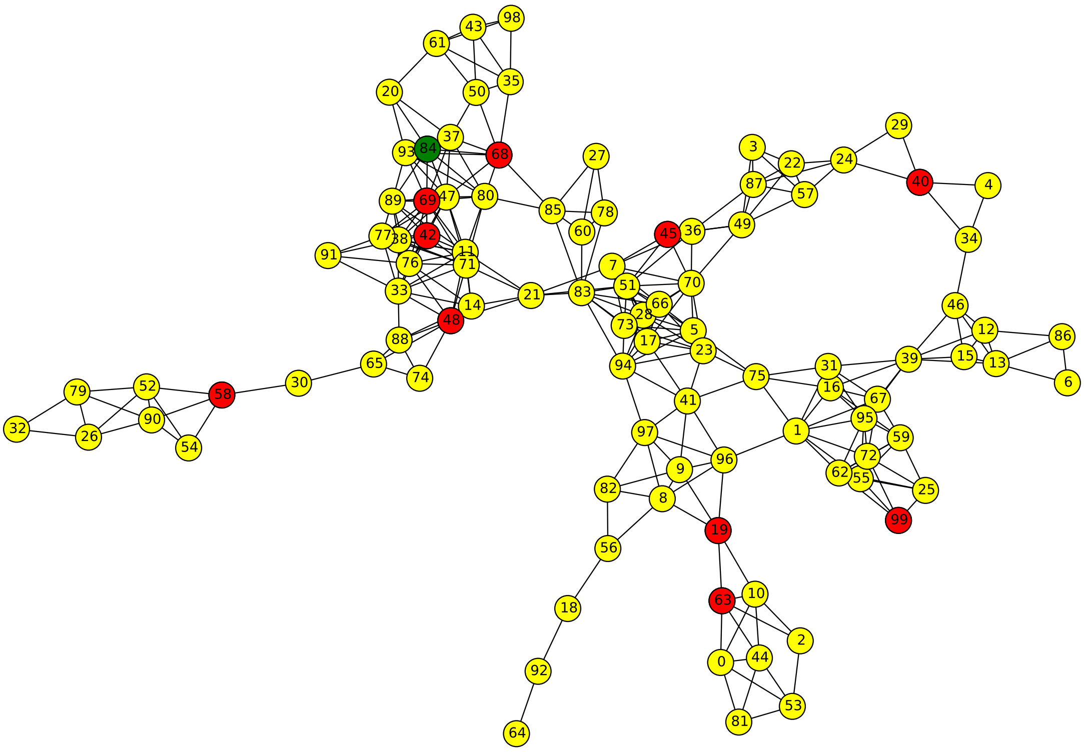


Figure 3.16: A connected RGG generated by SRS simulation tool, representing a synthetic network topology sample.

work topologies, each one with 100 nodes, *i.e.* RGG=(100,E). The connectivity was ensured for each RGG topology, using the SRS Auto mode, which automatically adjust radius (r) parameter conveniently, as described in Chapter 2 (p. 58).

Figure 3.16 depicts a generated RGG synthetic network topology sample, illustrating the ones used during evaluation.

Topology depicted in Figure 3.16 has one hundred nodes ($N = 100$) randomly distributed on a two-dimensional unit square ($[0, 1]^2$), with a threshold connectivity set by a radius of $r = 0.16$, record density is set to 10% (hosting nodes depicted in red) and the initiator node is node(84) (initiator node depicted in green). Nodes are completely independent entities and no relationship exists between the node(ID) and any other property (such as location).

Experiment setup During experiments, record density is ranging between 1% and 85%. For each record density percentage, 2 000 search queries are performed by each algorithm. In each query, a new RGG topology is generated, records are randomly assigned to nodes and the initiator node is randomly chosen from the nodes that do not host a copy of the record. All of the algorithms are independently executed and submitted to the same scenarios, ensuring a fair comparison. Each algorithm searches for a record in **synchronous rounds** by contacting only its direct neighbours, mimicking the multi-hop behaviour of the network. During each algorithm simulation run, the evaluation metrics are measured and collected. For each metric, at the end of the simulation, its average value is computed and stored for later result analysis. No warm-up time is considered since the SRS simulator has a single state for each discrete time interval and queueing operation are instantaneous.

Typically, in MANETs and WSNs the wireless sensors are set to duty cycling operating, where nodes are configured to enter in a sleep mode periodically, reducing energy consumption. Assuming a sufficiently small duration for each query dissemination, it is conceivable that the network topology remains stable during the search process.

The duty cycle time is tuned to meet the application requirements and the expected changes in the surrounding environment. Therefore, during evaluation testes it was considered a new topology for each new query search.

3.4.2 Result Analysis

To conduct the result analysis, we will assess the algorithm performance for time and energy, characterizing how they evolve, as the number of existing record replicas in the network increases. Even after setting the record density in the network, the random topologies impose a randomization effect on all the measurements.

Time efficiency Evaluation assessed latency, *i.e.* $\mathsf{L}_{\mathsf{proto}} = \#\mathsf{slots} \times \mathsf{T}_{\mathsf{slot}}$ (p. 26) for time efficiency purposes, measuring it by counting the number of time slots $\#\mathsf{slots}$ and considering that $\mathsf{T}_{\mathsf{slot}} = 1$s. One major negative impact in search algorithms caused by using chasing packets approaches, is a decrease in time efficiency. Figure 3.17 compares latency ($\mathsf{L}_{\mathsf{proto}}$) of BERS, BERS*, BCIR, BCIR* and Flooding algorithms, for several record density scenarios.

Figure 3.17 shows that as predicted in the theoretical model, the added delay required for algorithms using chasing packets has a significant impact on the algorithm latency response time. Figure 3.17 also confirms that latency for both

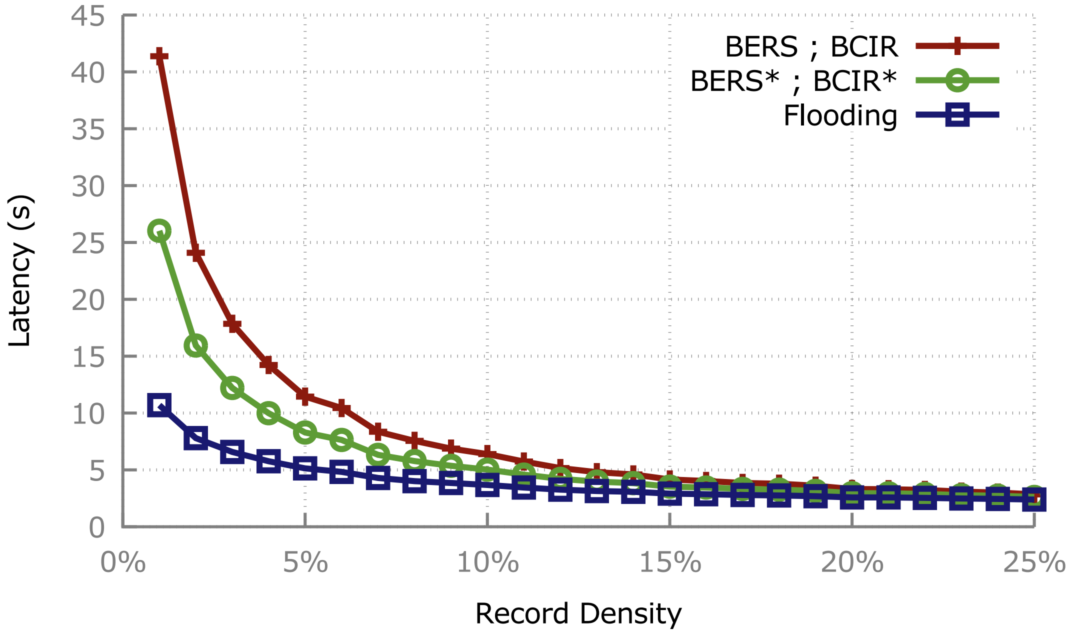


Figure 3.17: Comparing time efficiency for limited broadcast algorithms based on chasing packets.

BERS and BCIR is the same. A similar behaviour may be observed on the longer delay variants. The flooding algorithm (no added delay) is the fastest search algorithm and it can serve as a baseline for comparison with other algorithms.

The latency difference between BCIR and flooding algorithms is reduced near to half, when comparing BCIR* with flooding. This improvement in latency is better visualized in Figure 3.17 for lower record densities.

To improve graph readability, Figure 3.17 omits record densities exceeding 25%, since in that cases the probability of finding a record near to the initiator node is high, and all curves tend to overlap. This asymptotic behaviour may be observed for record densities near 20%.

Energy efficiency The retransmission ratio, *i.e.* $\mathsf{R} = \frac{\#\mathsf{Retransmissions}}{\mathsf{N}}$ (p. 27), may be measured by counting the number of retransmissions (#Retransmissions) in a multi-hop network with N nodes. A strategy to improve energy efficiency is reducing unnecessary radio transmissions.

Figure 3.18 compares the retransmission ratio of BERS, BERS*, BCIR, BCIR*

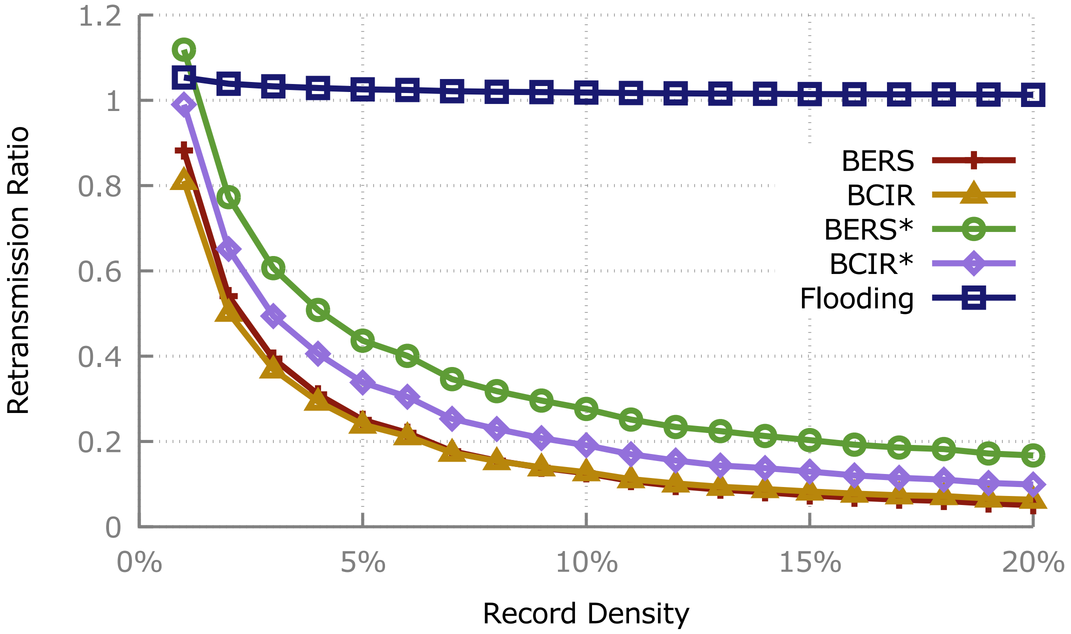


Figure 3.18: Comparing energy efficiency for limited broadcast algorithms based on chasing packets.

and Flooding algorithms, for several record density scenarios. It is very clear that chasing packets algorithms significantly reduce the retransmission ratio when comparing with flooding algorithms. As the record density starts to grow, the retransmission ratio rapidly decreases. For example, when record density is set to 5%, all the chasing packets algorithms save more than 50% of the average retransmission ratio when compared to flooding. As analytically predicted, the new BCIR algorithm is the most efficient of all, because there are no messages sent to the initiator, in contrast to what happens in BERS. However, the savings is marginal because the message is sent (point-to-point) back to the initiator.

As depicted in Figure 3.18, BERS* significantly increases the number of retransmissions (energy-inefficient) when compared with BERS. It suggests that improvements on latency have a negative impact on energy consumption, validating the predicted trade-off between energy and time efficiency. Likewise, Figure 3.18 shows that BCIR* has also a negative impact on energy efficiency when compared with BCIR. However, the new BCIR* shows a clear difference from BERS* energy performance.

As the likelihood of having a node hosting a record near to the initiator

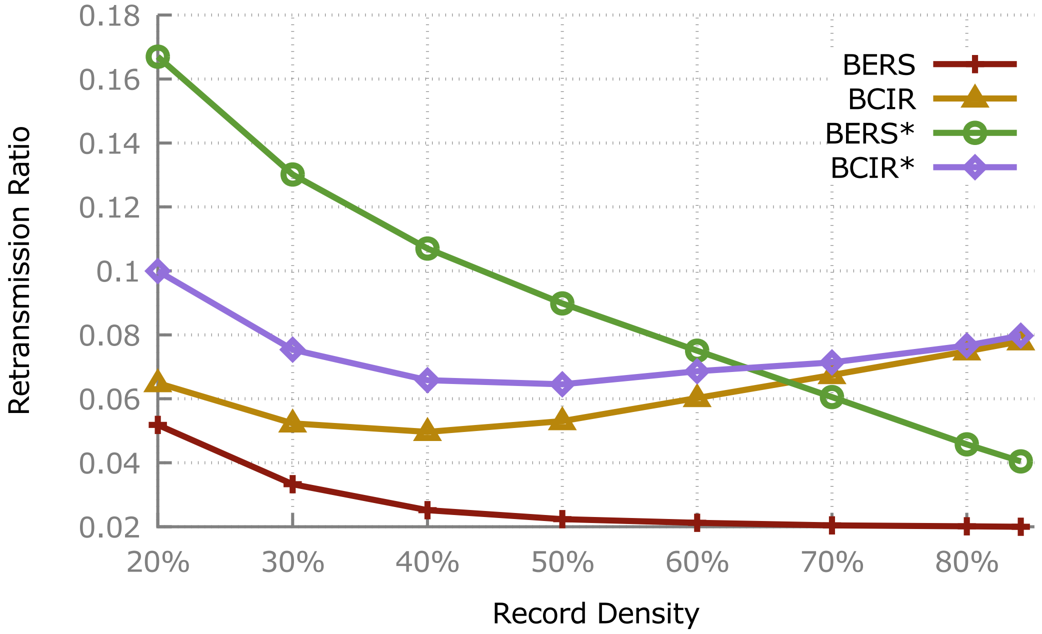


Figure 3.19: Energy efficiency observed considering a high record density.

increases (high record densities), all the chasing packets algorithms gradually improve their energy efficiency as target record density grows.

High record densities Considering scenarios with high record densities greater than 20%, all algorithms roughly conserve lower retransmission ratio, as depicted in Figure 3.19. However, both BCIR variations slightly compromises their performance. Flooding is not included in Figure 3.19 due to readability difficulties.

The retransmission ratio degradation observed in Figure 3.19 for BCIR and BCIR*, considering record densities above 60%, was not anticipated by the analytical model. A deviation attributed to the increasing probability of finding multiple record copies on the same hop (H), which in turn will lead to multiple concurrent cancellation waves (in contrast with a single cancellation wave from the source node in BERS and BERS*).

In these conditions, BCIR can introduce some duplication of cancellation messages at some nodes. The nodes have no information on the status of its neighbours and the multi-cancellation broadcasts will slightly increase the retransmission ratio for nodes belonging to distinct cancellation waves. Depending on the network topology, concurrent cancellation waves may result in some retransmissions overlapping.

3.5 Discussion

Previous evaluation section contrasts BCIR and BCIR* analytical model predictions with results from simulation experiments. The broadcast cancellation analysis led to the establishment of an analytical model for BERS, BERS*, BCIR and BCIR*, allowing to obtain formulas for latency and retransmission ratio estimation.

In BERS, the cancellation mechanism is triggered by the reception of a single acknowledgement message $\mathsf{m_a}$ that is sent point-to-point (optimal route path) back to the initiator node. Assuming the possibility of a temporary loss of connectivity, which may be caused by external adverse events or by packet drops during excessive network traffic, $\mathsf{m_a}$ delivery may fail and cancellation is compromised. In contrast, an interesting feature of BCIR is that $\mathsf{m_a}$ is included in $\mathsf{m_c}$ that is propagated using a broadcast approach (more resilient), with an immediate impact in the hosting node neighbourhood. An advantage of BCIR is to start cancellation as soon as possible, increasing the expected cancellation reliability by immediately initiating it, thus reducing the probability of topological changes, which may interfere with the cancellation process.

When BERS* algorithm is replaced with BCIR*, the expected energy saving Δ_R, increases linearly with the number of hops as predicted by Eq.(3.18) (p. 97). As more hops are required for a successful record discovery, more energy will be saved by the new BCIR* mechanism. Energy gains of the BCIR* mechanism are more visible in networks with low record density. As depicted in Figure 3.18 (p. 104), the increase in energy gain has more impact for networks with record densities lower than 40%. The BCIR* characteristics have a potential to be more efficient in large networks, assuming that record density decreases as the network size grows.

Dispersion An interesting discussion is a characterization of energy efficiency deviation from its average values, complementing the information provided in Figures 3.18 – 3.19. SRS computes the standard deviation (σ), characterizing the BCIR retransmission ratio variability and its dispersion with respect to the

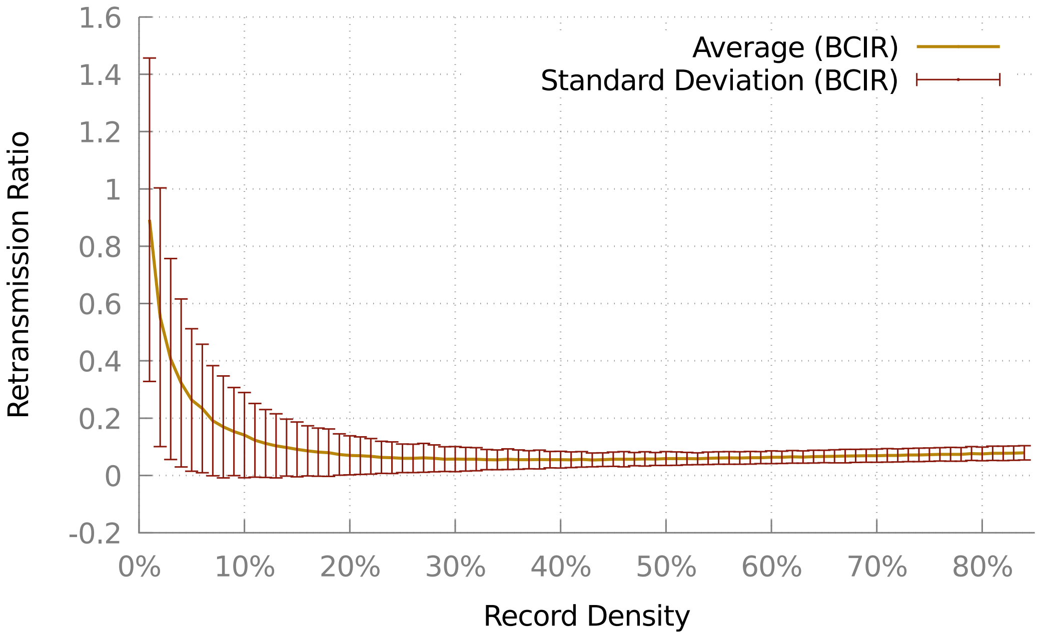


Figure 3.20: Retransmission ratio standard deviation (σ), estimating dispersion.

average values, as depicted in error bars from Figure 3.20.

Random distributions generate broader dispersion distributions. Thus for low record density scenarios the first target record may be represented by a random distribution and the expected higher dispersion values are confirmed by Figure 3.20. As the record density increases, it also increases the likelihood of finding a record closer to the initiator node, contributing to a lower dispersion of the values. Considering the retransmission ratio results, BCIR* algorithm has a greater energy efficiency than BERS*. Our proposed algorithm for BCIR is expected be suitable for MANETs live topologies, since cancellation is immediately initiated by hosting nodes, with a direct impact on target neighbourhood and due to the absence of point-to-point communications.

Results for BCIR show that it is possible to obtain a latency at least equal to the faster competing mechanism, with lower energy cost. However, all the compared cancellation mechanisms are vulnerable to jitter or faults occurrences during the cancellation message propagation, which may prevent broadcast termination and its corresponding query may flooding the entire network. An important result is achieved by BCIR* across most density ranges, that while keeping latency low may still perform a reduction in the retransmission ratio (when compared to BERS*), thus achieving important energy savings, as depicted in Figure 3.19 (p. 105).

More recently, the results obtained with Broadcast Cancellation Initiated at Resource (BCIR) were confirmed by another team that developed an autonomous work, proposing a similar mechanism known as timeBERS (tBERS) [220].

The development of BCIR confirmed that using chasing packets to cancel the query broadcast propagation is more energy-time efficient than hop-limited approaches. However, adding a minimum Delay in the search component, for reserving the required time for an effective cancellation, will generate a negative impact in time efficiency. Both hop-limited and chasing packets approaches improve energy efficiency at the expense of latency, leaving space for new research efforts for attenuate this trade-off, as verified in [217].

Results obtained by simulation show that it is possible to reduce the retransmission ratio without increasing latency, contributing to extend the network lifetime of battery powered devices. Contributions made by the BCIR approach may be extended to other domains, where uniformed search mechanisms assumes a major role in query services, with a positive impact on the availability of applications supported by battery powered devices.

Chapter 4

LBF: Linear Bloom Filter

The Bloom Filter (BF) [33] is a space-efficient probabilistic data structure, used to test whether a given element belongs to a set. In this chapter we propose a variation of Bloom filters named **Linear Bloom Filter (LBF)**. In contrast with the boolean characteristics of BFs, the LBFs associate with each element in a set, a real number quantized within the interval $[0, 1]$. However, LBFs keep the probabilistic nature of BFs allowing them to balance between storage space and response accuracy. Calibrating LBF parameters allows to efficiently manage the data structure, adjusting it to meet the application requirements and fitting to the problem scale. Characteristics of LBFs are appealing for wireless sensor networks applications, where devices share several constraints that demand saving critical resources like memory, CPU or power. In the particular case of multi-hop wireless networks, the probabilistic data structures are suitable to accommodate data whose relevance is not uniform and typically decreases with distance [155].

The LBF equally inherits a valuable property from BF, that is a data structure with an unalterable size supporting an unknown number of members, making it adequate for unstructured networks.

The remainder of this chapter is organized as follows: first, it provides the motivation that led to the LBF development and identifies some of its applications, in the context of MANETs and WSNs, such as resource discovery or data dissemination services; then, the relevant literature is reviewed, focusing on the most significant developments on the related work, regarding the use of probabilistic data structures by search algorithms; after that, the LBF characteristics, core operations and its representative linear quantization are described; then the LBF evaluation is resumed to include a LBF implementation to measure the impact of the calibration parameters in the LBF response estimation; finally, the LBF relevant improvements are discussed.

4.1 Motivation

Traditional Bloom Filters (BFs) are limited to a boolean indication of the presence/absence of each element in a set. The BF data structure is enough to manage locally hosted records on each node. However, it is inadequate to manage ambiguous data records, which are hosted by other network nodes, from which only an expected indication exists.

One of the main characteristics exhibited by BFs is its efficient management of the memory space, achieved by adopting a probabilistic approach for data representation [142]. However, there is always a certain degree of uncertainty to it, due to its non-deterministic nature, which may result in false positive answers. The probability of false positive answers may be tuned by adjusting the filter parameters, allowing for each application to establish its own boundaries for an acceptable error level.

The main research motivation to propose the new Linear Bloom Filter (LBF), emerges from the need to efficiently manage highly dynamic data membership, when searching for records in the WSNs application domain. The BF efficiency is an appealing characteristic for WSNs intensive query-based record search applications [255]. However, in a multi-hop network, and despite nodes does not have information about the target record locations, its relay function may be conditioned by the knowledge that a given record, *i.e.* elem_n, may be hosted by some neighbour node, and with its own relay capability contribute to speed up the search task. The motivation is to manage any indicative hint, that may be useful to accelerate the query propagation, by reducing the added delay, in regions closer to the target record. Focusing on a basic BF structure, in addition to insert local elements in a set, it would be useful to add membership information of nearby elements and associate them with an expected proximity. The lack of a simple structure serving the previous applications led us to propose a new type of Bloom filter designated as Linear Bloom Filter (LBF). The LBF extends the traditional BF, by including some uncertainty when defining a set of expected data values.

In WSNs, the sensor nodes frequently exchange messages for reporting or relaying the sensed data. Typically, uninformed searches are based on flooding approaches, where messages are propagated by intermediate sensors nodes to discover the first node hosting the target record, *i.e.* $\mathsf{elem_n}$, without any information about network topology or record location. Nevertheless, all these previous messages may be useful to acquire some network knowledge. The LBF may be piggybacked within those messages, and used to propagate some network property, an approach already used by several search mechanisms [52, 100, 165]. In the case of multi-hop networks, the LBF confidence parameter may be used to estimate distance, which is a very significant knowledge property for informed search mechanisms in unstructured networks.

Several informed search mechanisms may benefit from the inclusion of LBFs data structures within any exchanged messages. Then, relaying nodes may bias the broadcast propagation using hints given by the LBFs confidence parameter, which represents knowledge gained during previous exchanged messages with LBFs piggybacked. The LBF introduces a rather general Bloom filter technique, that may be potentially applicable in other contexts apart from wireless unstructured networks.

4.2 Related Work

The initial Bloom filter proposed by Burton Howard Bloom in 1970 [33] underwent several developments to address new application needs [255]. Initially, the Bloom Filter (BF), which is a set representation, allowed only for the addition of new elements ($\mathsf{elem_n}$) or answering a given query, as reviewed in Section 2.4 (p. 41). However, some applications need to remove elements from sets.

To allow for deletion operations, a new variant known as Counting Bloom Filter (CBF) was proposed [81]. The CBF vector representation replaces the single bit values with a counter, *i.e.* *n-bit*, representing multiple $\mathsf{elem_n}$ replicas. CBF introduces the possibility to decrement that counter, thus allowing for deletion operations. A generalization of CBF suggests two schemes to update each entry, enabling variable increments at the expense of calculating the double of the hash functions [229]. The Spectral Bloom Filter (SpecBF) is an extension of the original BF to support multi-sets, allowing the filtering of elements whose multiplicities are below a threshold, given at query time [62].

BFs were experimented as routing table substitutes, contributing to reduce memory and computing time consumption. Assuming that deletion operations in routing tables are a consequence of explicit withdrawal messages, the Withdrawal To annOuncement (WTO) protocol is an alternative approach to CBF. In WTO, the withdrawal messages are replaced by announcement messages, thus transforming deletions into additions or record changes [275].

In multiple systems that require the processing of large data sets, BF performance becomes critical in several aspects that need to be improved. Some optimizations were proposed to increase BF accuracy and reduce false positive responses, balancing memory consumption and processing overhead [117]. The widely application of BFs in many network operations requires faster BFs, as in the generalization proposed in [223]. The previous work shows that BFs may be configured with a smaller number of memory accesses, a smaller or equal number of `hash(x)` functions, and a smaller or comparable false positive ratio, thus outperforming traditional BFs configurations in practical scenarios. Another optimization approach improves the hash functions algorithms, for a more efficient algorithm execution and a reduction of the required storage space [35].

Scalable Bloom Filters (SBF) [15] allow for a dynamic adjustment of the filter quality. The accuracy of BFs is influenced by the ratio between the number of elements and the size of the data structure. SBFs avoid two problems: *i*) small predefined filters, will rapidly saturate, rendering the filter will be worthless, and *ii*) an oversized predefined filter, that may be responsible for allocating memory that will rarely be used. The use of BFs in distributed applications triggered a similar technique that allows for the implementation of dynamic growth BFs [101], where the filter size is adjusted (in run-time) as a function of the number of inserted elements.

The Bloomier [52] variant can encode functions by extending the traditional queries (limited to test whether or not an element is a set member), and allowing to associate attributes to the Bloomier added elements. The Bloomier data structure is made of multiple simple BF in parallel.

Some recent BF variants expand the traditional BFs application domains to include others such as the WSNs domain. Some examples of those applications are: *i*) network context acquisition and *ii*) record discovery, that typically include one or more BFs along the search messages [165, 255].

Some security measures should be considered to reduce the risk of attack when transmitting BFs over the network. For example, an *all-one* filter attack (warning about saturated filters) makes it impossible to distinguish the inserted elements from false positives responses. The Generalized Bloom Filter (GBF) strategy is to limit the rate of false positives [149].

In sensor networks, nodes may acquire context information by listening to its neighbours messages. The authors in [127] propose the use of BFs to summarize acquired context information, to identify groups of things with certain similar characteristics, and to make rapid routing decisions to reach the hosting nodes. However, no experimental results were presented.

The *attenuated Bloom filter* variant [227] was developed for searching the location of documents, in the context of a literature application. The data structure had to cope with document motion over *peer-to-peer* networks. This type of filter is formed by an array of BFs, that stores for each position **i** the resulting BF from the contribution of all servers that are at **i** distance. The filter data may be useful to estimate the distance (hop) between server-related documents. This mechanism allows for the acquisition of context from neighbour nodes and proposes a probabilistic localization algorithm. This approach reveals a high potential application for WSNs. However, the data structure size depends on the network diameter, which is a very demanding condition for WSNs with limited resources (memory, cpu, energy).

The Cuckoo filter was developed to allow for a dynamical removal of elements while improving the filter performance, measured in storage space and query time [80]. An example of application is a flash-based key-value storage system, known as Small Index Large Table (SILT), that uses a filter based on partial-key cuckoo hashing [157].

Assuming that in large networks the distant nodes are the least important ones, a new mechanism was proposed to decay the information contained in a BF and find a gradient that matches the route between wireless devices [100]. However, the diffusion mechanism is too heavy and does not operate efficiently in networks with dynamic topologies. Following the previous principle (that the contribution of nodes decreases with distance), the Exponential Decay Bloom Filter (EDBF) proposes a filter decaying mechanism [144]. The EDBF is a Bloom filter variation to efficiently aggregate and propagate probabilistic data hosted in the neighbourhood of a node. The proposed decay mechanism exponentially decreases the filter content with distance, to restrain the impact of the network dynamism, by giving more importance to information found nearby. The EDBF is the main component of a scalable query routing mechanism, that uses the probabilist information from EDBF to forward queries towards to the hosting node, with a significant advantage over the completely blind nature of uninformed flooding mechanisms.

4.3 LBF: Probabilistic Data Structure

For large networks, developing search mechanisms optimizations that require full network context, if theoretical acceptable, is nevertheless impractical for sensor networks, where devices have several resources constraints (*e.g.,* cpu, memory). Taking into account the fundamental BFs properties (Section 2.4) and driven by some search mechanisms requirements, we propose a Linear Bloom Filter (LBF) to efficiently manage hints about expected record locations.

LBF Description The Linear Bloom Filter (LBF) is a probabilistic data structure similar to a Bloom Filter [33]. The LBF extends the traditional BFs, allowing them to handle inaccurate data. The LBF introduces an additional parameter, designated as **confidence**, to each filter element. The LBF no longer represents just a simple set, but it also maps elements with quantities, forming a tuple $(\mathsf{elem},\mathsf{c})$. In addiction to answering questions about the membership of an element, the LBF response may also report an estimation of the data membership confidence, *i.e.* representing vaguely defined sets.

Generically, knowing that a real number that lies between two numbers in a set is also included in an arbitrary totally ordered set (real interval), and assuming that physical quantities may be represented by a given percentage, then any real quantity may be transformed to fit into the unit interval, *i.e.* $\mathsf{c} \in [0, 1]$. Notice that the unit interval is a subset of the real numbers ($\mathbb{R}$), with the same size as the whole set.

One main novelty aspect of LBF is the generalization of a Bloom filter set membership. The traditional BF provides boolean responses, an element either belongs (with some false positive answers) or does not belong to a given set. In contrast, the new LBF introduces a membership uncertainty, defined by the confidence parameter $\mathsf{c} \in [0, 1]$, to each element in a set. The general concept is related with fuzzy set theory, and c reflects the uncertainty of a given element to belong to a set [284]. The LBF confidence component enables a partial membership status, with respect to a given set of elements.

LBF Implementation The simplest implementation for LBF may be a vector of floats with a length of m cells. At default, LBF is initialized by setting all vector cells with 0 value (as usual, 0 indicates a non member set). Setting $\mathsf{c} = 1$ for all the inserted tuples, the LBF behaves as a traditional boolean BF. The main novelty introduced by LBF is the capability to manage uncertainty, with a grade of membership, by considering intermediate values between the two extremes given by $\mathsf{c} \in [0, 1]$.

To accommodate the c values, each LBF cell may not be limited to admit a single bit only. Depending on the precision required to represent c, a LBF cell may require several binary positions. The vector dimension must ensure that all cells are capable of accommodating the c binary representation. Typically, each cell is referenced by the corresponding position index (an integer value). As in Bloom Filters, LBF uses $\mathsf{hash}(\mathsf{x})$ functions to ensure an independent and random uniform distribution cell indexation. The LBF core operations (insert, query) manage the mapping information between each element and the corresponding confidence parameter, forming a tuple $(\mathsf{elem,c})$.

4.3.1 Core Operations

LBF considers two core operations: *i*) **insert**, for adding a tuple to the set, and *ii*) **query**, for questioning element membership. Both operations are similar to BFs core operations (Section 2.4). However, instead of pure binary data they operate with binary representation of a tuple. To insert a tuple into the LBF, the element and its confidence value must be known. Querying for an element returns its confidence estimation, which is represented by $\hat{\mathsf{c}}$.

The **insert** operation is implemented by function $\mathsf{INS}(\mathsf{elem}, \mathsf{c}, \mathsf{LBF}[])$ described

Algorithm 4: Linear Bloom Filter (LBF) core operations

```
const LBF[] ← [0, ··· , 0];                        // Linear Bloom Filter
const k ← 3;                                       // Number of hash functions
Function INS(elem, c, LBF[])
    for i ← 1 to k do
        idx ← $hash_i$(elem);
        LBF[idx] ← max(c, LBF[idx]);
    end
Function QRY(elem, LBF[])
    c ← 1;                                         // Confidance index
    for i ← 1 to k do
        idx ← $hash_i$(elem);
        c ← min(c, LBF[idx]);
    end
    return c;
```

by Algorithm 4. The LBF insert operation is similar to the same operation in the basic BF, but instead of setting a single bit for each of k cells, it computes the maximum between c and the cell value for each of k LBF cells.

Figure 4.1 depicts a LBF sample after adding tuples {(a;1),(b;0.9)} to an empty filter, by invoking functions INS(a,1,LBF[ ]) and INS(b,0.9,LBF[ ]) respectively, as described in Algorithm 4. For better readability, the following examples use a LBF with $m = 16$ cells and $k = 3$ hash(x) functions. The application of k hash(x) functions to each element resulted in the following LBF cell indexes:

$$\mathsf{hash_1(a)} \mid \mathsf{hash_2(a)} \mid \mathsf{hash_3(a)} = \{3; 5; 13\}$$
$$\mathsf{hash_1(b)} \mid \mathsf{hash_2(b)} \mid \mathsf{hash_3(b)} = \{5; 9; 14\}$$

The insertion of both tuples led to an overlap in cell 5, a potential ambiguity resolved by using a max operator, *i.e.* $\max(1; 0.9) = 1$, discarding the lower

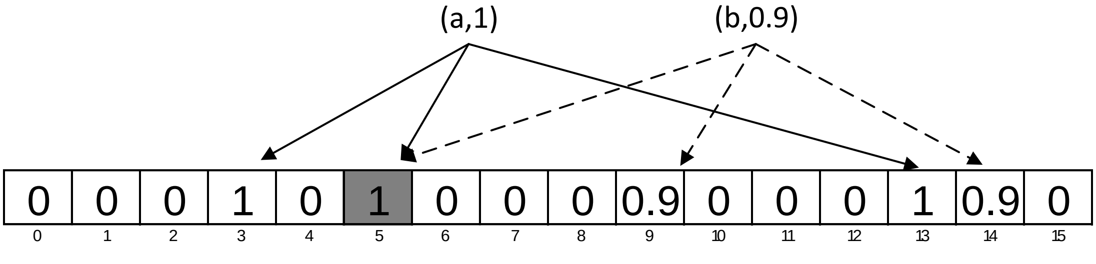


Figure 4.1: Representing elements membership with a LBF and the corresponding confidence values. LBF cells content upon tuples (a;1) and (b;0.9) insertion into an empty filter.

confidence values.

The LBF **query** operation is implemented by function $\mathsf{QRY(elem,LBF[\ ])}$, as described in Algorithm 4. Querying the LBF filter for elem, works as a grade of membership estimator $\hat{\mathsf{c}}$, returning the expected confidence associated to elem. The estimator $\hat{\mathsf{c}}$ returns the minimum value from the k LBF index cells values.

Figure 4.2 depicts a query LBF example, illustrating the confidence estimation for a non inserted element x. The inspected cells are obtained by $\mathsf{k} = 3$ $\mathsf{hash(x)}$ functions:

$$\mathsf{hash_1(x)} \quad | \quad \mathsf{hash_2(x)} \quad | \quad \mathsf{hash_3(x)} = \{3; 9; 12\}$$

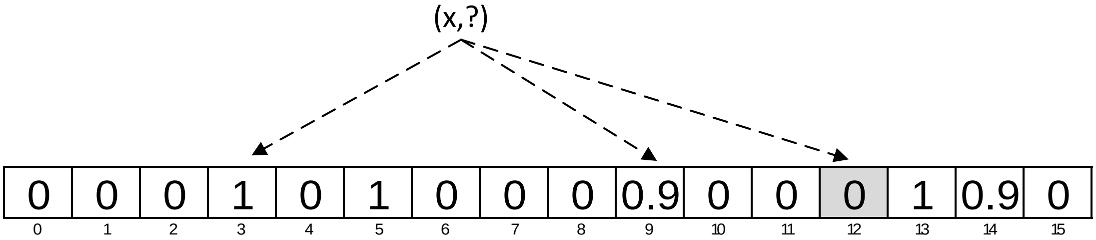


Figure 4.2: Query for element x membership and LBF reporting that it does not belong to the set.

The query example depicted in Figure 4.2 uses a filter that experiences a partial index overlapping (as depicted in Figure 4.1). Nevertheless, the min operator inside the QRY function ensures the LBF immunity to a partial overlap. In this case, the overlap in positions $\{3, 9\}$ does not affect the LBF response and it provides the correct answer, *i.e.* $\min(1; 0.9; 0) = 0$. Thus, the LBF answer reports an estimated confidence value of $\hat{\mathsf{c}} = 0$, suggesting that x does not belong to the set, which is a correct response. Although LBF is not immune to false positive responses, it does not admit false negatives.

A saturated LBF occurs when most of the filter cells have non null values. The example depicted in Figure 4.3 shows a saturated filter. As LBF is close to saturation, the occurrence of a false positive response increases.

$$\mathsf{hash_1(y)} \quad | \quad \mathsf{hash_2(y)} \quad | \quad \mathsf{hash_3(y)} = \{3; 8; 12\}$$

In the example depicted in Figure 4.3, that queries for a non inserted element y, *i.e.* $\mathsf{QRY(y,LBF[\ ])}$, the expected estimation returns $\hat{\mathsf{c}} = \min(1; 0.8; 0.9) = 0.8$, which is a false positive response.

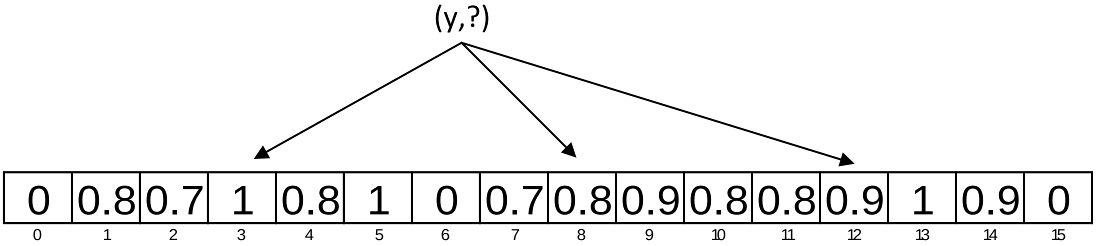


Figure 4.3: A false positive response occurrence. Query for element y returns an estimated confidence value of $\bar{\bar{\mathsf{c}}} = \mathsf{0.8}$, however no tuple with a y element was ever added to the set.

LBF Overestimation In LBF, the false positive effect is a special case of a more general problem denoted as overestimation. Inserting tuples using the max operator (Algorithm 4) in a filter causes that tuples with higher confidence values overwrite other tuples with lower confidence values. Adding too many tuples in a LBF with high confidence values will deteriorate the LBF response. Queries responses are affected by an overestimation of the confidence parameter in dense filters.

Both false positive and overestimation behaviours are a consequence of a wrong filter calibration. Assuming an optimal selection of the number of hash(x) functions, the number of LBF cells (that may be converted to filter size in bits) should be enough to manage the expected number of tuples.

The LBF allows successive **attenuation** operations, in which all filter cells are multiplied by a given factor att (*e.g.,* $\mathsf{att} = \mathsf{0.9}$). The consequence is a reduction in the confidence grade of membership for all LBF tuples. Recurring attenuation operations converges the filter cells values to 0. The attenuation is a very simple mechanism to decay information contained in the LBF.

Applying attenuation operations has many practical applications in the context of WSNs. A particular example may be found in Chapter 5 where it is used as a decaying mechanism. Decaying information dissemination over distance (the more distant the information is, the more attenuated it will be) enables the creation of gradient-based search mechanisms for multi-hop networks [161].

4.3.2 Linear Quantization

The representation of a real number in a binary format, requires a quantization operation which maps a continuous interval within a countable small set. However, there are infinite numbers in any given $\mathbb{R}$ segment interval, such as $[0, 1]$ and the process of generating a binary representation of a real number ($\mathbb{R}$) involves a lossy operation. The quantization error is the difference between the original value and its quantized representation.

LBF implementation requires a representation of the confidence parameter $\mathsf{c} \in [0, 1]$ in a binary format. According to the IEEE 754 standard, there are several ways to represent a real number in a binary form. IEEE 754 standard defines the appropriate formats for representing floating-point numbers, establishing different accuracy levels. The most common representations are: *i*) single precision using a 32-bit format and *ii*) double precision using a 64-bit format. LBF does not impose any restriction on the representation format to be used. However, a simple quantization mechanism is suggested. Moreover, when a small countable set is assumed, there is no need for an additional compression layer.

The LBF parameter c is a real number limited to the interval $[0, 1]$, which permits to apply some simplifications on the generic cases of the standard procedures used for binary representation. Since c does not have sign or exponent, it is enough to make the representation of its mantissa. The IEEE 754 single-precision standard allocates 23 bits to represent the mantissa, a precision that clearly exceeds the requirements for resource location in sensor networks, since: *i*) LBF is a probabilistic data structure (with uncertainty), and *ii*) sensor devices have significant memory and CPU constraints.

The quantization error is limited by the maximum number of bits (b) used in the quantization process. The b is also known as the quantization **precision** and is one of the key LBF calibration parameters. The amount of memory (in bits) allocated for each LBF is $\mathsf{dim}(\mathsf{LBF}) = \mathsf{m} \times \mathsf{b}$ *i.e.* the LBF size, where m is the number of cells available in a LBF. These parameters should be adjusted in line with the desirable accuracy for the final application, and by knowing that as b increases the quantization error decreases. The negative impact of increasing m or b is the linear increase in the amount of memory usage by LBF, which requires a rigorous accounting regarding both (m and b) parameters.

Algorithm 5: Confidence binary representation

```
Function c2bit(c, b)
    return floor( c × ((1 ≪ b) − 1) ;                    // floor(x) : ⌊x⌋
```

The greatest positive integer represented by a set of b bits is $(2^b - 1)$. Linearly rescaling $c \in [0, 1]$ may be accomplished by computing $c \times (2^b - 1)$. The result of this operation returns a real number ($\mathbb{R}$), which may be converted to an integer by quantization, by using a truncation function. Using previous operation, the quantization maximum absolute error ($\mathsf{LBF}_{\mathsf{ERR}}$) is given by Eq.(4.1).

$$\mathsf{LBF}_{\mathsf{ERR}} = \pm 1/(2^b - 1) \tag{4.1}$$

The $\mathsf{LBF}_{\mathsf{ERR}}$ may be parametrized to meet the application requirements. Considering the focus on search mechanisms, there is an implicit association that $c = 1$ may be used to represent a local record (there is no uncertainty in local hosted records), which should also correspond in its binary representation to set all the b bits to 1. To maintain the previous convention for local records, it is best to implement the quantization using the operator $\mathsf{floor}()$, which returns the integer part of the floating number. The $\mathsf{floor}()$ ensures that after LBF suffers a given attenuation (small as it may be), the binary representation will never have all bits set to 1. Algorithm 5 describes the function call; $\mathsf{c2bit}(c, b)$, used to generate the binary representation of the confidence parameter.

The LBF may be represented by an array of cells where each cell contains the binary representation of the confidence parameter c, as illustrated in Figure 4.4. It depicts the binary representation corresponding to the LBF [5-8] cells, resulting from choosing a precision of $b = 4$ bits for quantization. The expected absolute error is $\mathsf{ERR} = \pm 1/(2^b - 1) = \pm 0,067 \approx \pm 7\%$. As may be observed in Figure 4.4, the final LBF structure still is an one bit vector.

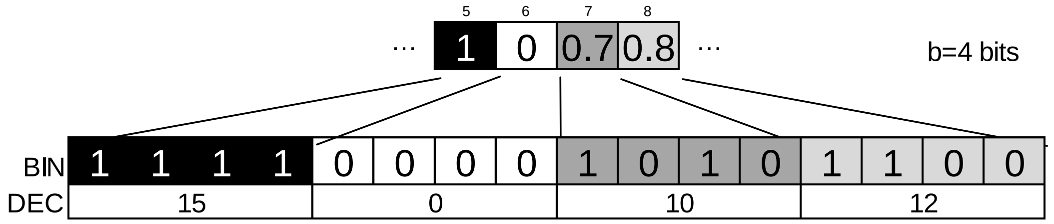


Figure 4.4: Binary representation for LBF cells content.

To estimate the confidence grade of membership of an element, *i.e.* $\hat{\mathsf{c}}$, the LBF may be inspected by function $\mathsf{QRY(elem,LBF[\,])}$, which performs two operations: *i*) the query searching for elem, and *ii*) the reverse quantization to obtain $\hat{\mathsf{c}}$.

Decaying LBF data For decaying the LBF information, an attenuation operation is required. The attenuation must have access to all non null LBF cells. It starts by applying the reverse quantization operation on each cell to obtain the corresponding decimal number using $\mathsf{BIN2DEC(x)}$; it subsequently applies the attenuation operation $\mathsf{att} \times \mathsf{BIN2DEC(cell_i)}$, then represents it in a binary form using the quantization method and update the cell contents.

4.4 Evaluation (LBF)

Querying a probabilistic data structure may return a different information from the one that was inserted, which allows to anticipate that LBF will present similar problems. The query response for a tuple $(\mathsf{elem_n}, \mathsf{c})$ returns $\hat{\mathsf{c}}$, which is an **estimated** grade membership for $\mathsf{elem_n}$, quantified by the $\hat{\mathsf{c}}$ value. Thus, LBF is mapping elements into another numerical quantity with some degree of uncertainty.

Mainly, evaluation assesses the impact of LBF parameters calibration into the quality of LBF responses. To quantify LBF performance, first we implemented it and then we submitted it to several stressing scenarios. The evaluation distinguishes two error components: *i*) errors caused by the probabilistic nature of the data structure, and *ii*) a deterministic error component imposed by the quantization process. It is expected that a good LBF calibration may reduce the previous errors to an acceptable level.

The evaluation considers the existence of two data structures (in parallel) for tuples $(\mathsf{elem_n}, \mathsf{c_n})$ membership: *i*) a deterministic structure associating each $\mathsf{elem_n}$ with their respective $\mathsf{c_n}$ value, and *ii*) a probabilistic one implemented with a LBF that estimates the corresponding c value, *i.e.* $\hat{\mathsf{c}}$. The evaluation experiments use both data structures to compare the LBF probabilistic response with a deterministic one.

For each experiment, n tuples $(\mathsf{elem_n}, \mathsf{c_n})$ are generated and inserted in both data structures, where $\mathsf{elem_n}$ is assigned with a unique sequence identifier and

Table 4.1: Environment experiments static parameters.

Attribute	**Value**
N - N° of experiments	100 000
dim(LBF) - Filter size	4096 bits
n - N° of inserted tuples for each experiment	100

c_n assigned with a pseudo-random uniformly distributed generator for the interval $[0, 1]$. The tuples mapping information is managed in both structures: *i*) deterministic (*e.g.,* vetorElem[ ]), and *ii*) probabilistic (*e.g.,* LBF). The LBF is evaluated by executing $N = 100\,000$ experiments, comparing the LBF response, *i.e.* the estimator with a deterministic approach.

For a fair comparison, we preferred to keep a stable LBF memory usage, by setting an invariable LBF size. For evaluation proposes, we maintained a constant LBF size of $\mathrm{dim(LBF)} = 512$ Bytes. The number of cells m depends on the number of bits used for quantization b, given by Eq.(4.2).

$$m = \frac{\mathrm{dim(LBF)}}{b} \tag{4.2}$$

Assuming a precision of $b = 4$ bits per cell, is sufficient to represent c, the filter has $m = \frac{4096}{4} = 1024$ cells. The number of cells occupied by n tuples is always less equal to $k \times n$. Assuming that $k < 10$, it seems reasonable to consider that $n = 100$ tuples may be accommodated in a filter with 1024 cells. Table 4.1 summarizes the previous considerations and brings together the main experiments parameters.

The most peculiar calibration parameters for LBF are: *i*) b - precision and *ii*) k - number of hash(x) functions that need to be settled in a LBF operational implementation. For those parameters, several evaluation experiments were put in place, considering a variable range for each one.

The impact in the LBF response caused by the number of precision bits b is evaluated taking into account a number of different scenarios, from a minimal binary case $b = 1$ to a maximum precision of $b = 16$ bits per cell. However, $b = 4$ seems to be a reasonable choice, corresponding to a quantization error of 6.25%. The k magnitude for evaluation, was determined in accordance with the recommendations given by the optimal solution [255]. One of this theoretical ap-

Table 4.2: Environment experiment dynamic parameters.

Attribute	**Range**
b - Precision (N° of bits per cell)	$1\,\text{bits} \leq \mathsf{b} \leq 16\,\text{bits}$
k - N° of $\mathsf{hash(x)}$ functions	$2 \leq \mathsf{k} \leq 16$

proach outcomes is Eq.(2.3), which returns $\mathsf{k} < \frac{4096/4}{100} \ln 2; \mathsf{k} \approx 7$. The evaluation scenario considers a minimum of $\mathsf{k} = 2$ to maximum of $\mathsf{k} = 16$ $\mathsf{hash(x)}$ functions. It is expected that scenarios resulting from parameters around $\mathsf{b} = 16$ and $\mathsf{k} = 16$ (with the same n), will create the conditions for some LBF saturation to be observed. Table 4.2 summarizes the dynamic of the previous parameters, used for evaluation tests.

Evaluated metrics A sample of $\mathsf{N} \times \mathsf{n}$ independent observations were considered. The query response is an estimation of the c value and denoted as $\hat{\mathsf{c}}$. The metrics used for characterize the LBF query response are:

Average The $\bar{\hat{\mathsf{c}}}$ sample average, given by Eq.(4.3).

$$\bar{\hat{\mathsf{c}}} = \frac{1}{\mathsf{N} \cdot \mathsf{n}} \sum_{\mathsf{i}=1}^{\mathsf{N} \cdot \mathsf{n}} \hat{\mathsf{c}}_\mathsf{i} \tag{4.3}$$

Standard Deviation The dispersion of $\hat{\mathsf{c}}_\mathsf{i}$ values is quantified with the standard deviation σ, given by Eq.(4.4).

$$\sigma = \sqrt{\frac{1}{\mathsf{N} \cdot \mathsf{n}} \sum_{\mathsf{i}=1}^{\mathsf{N} \cdot \mathsf{n}} (\hat{\mathsf{c}}_\mathsf{i} - \bar{\hat{\mathsf{c}}})^2} \tag{4.4}$$

Mean Square Error (MSE) To quantify the difference between the estimated value $\hat{\mathsf{c}}_\mathsf{i}$ and the its deterministic value c_i, the $\mathsf{MSE}(\hat{\mathsf{c}})$, that is the average of square errors, is given by Eq.(4.5).

$$\mathsf{MSE}(\hat{\mathsf{c}}) = \frac{1}{\mathsf{N} \cdot \mathsf{n}} \sum_{\mathsf{i}=1}^{\mathsf{N} \cdot \mathsf{n}} (\hat{\mathsf{c}}_\mathsf{i} - \mathsf{c}_\mathsf{i})^2 \tag{4.5}$$

Fill Ratio To quantify the filter density, the LBF fill ratio was measured and given by the ratio between the number of occupied cells and the filter size, *i.e.* $\mathsf{dim}(\mathsf{LBF})$ (expressed in cells).

The detailed results obtained from previous metrics are presented in Sections 4.4.1 – 4.4.3, each one dedicated to the LBF calibration parameters.

4.4.1 Capacity Calibration

The LBF capacity is related to the number of tuples n and the memory requirements $\mathsf{dim}(\mathsf{LBF})$ to manage them, without deteriorating the filter performance or wasting memory. Change the filter size has a direct impact on the filter capacity, which is a trivial dependency. For this reason, the LBF size was maintained invariant, ensuring a constant memory allocation given by $\mathsf{dim}(\mathsf{LBF})$. However, the number of filter cells depends on the precision, which is given by Eq.(4.2) (p. 122).

The existence of some overlap in the LBF indexes allows for a cell to be indexed by distinct tuples. Thus, the **fill ratio**, defined as the percentage of occupied cells, is a preferable metric for filter capacity. The LBF capacity is

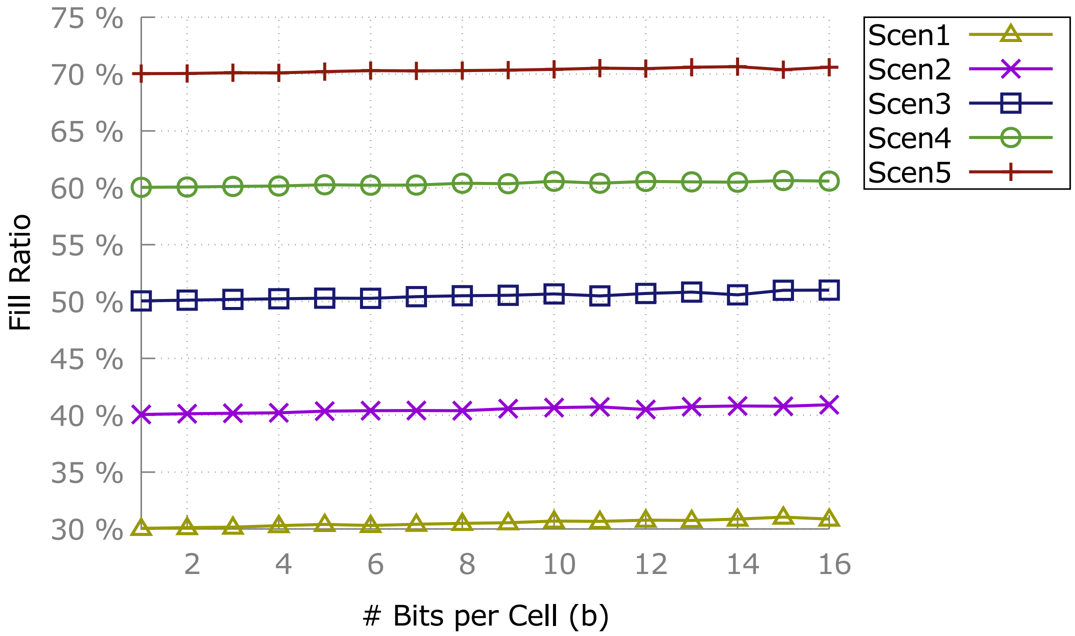


Figure 4.5: Adjusting the number of inserted tuples when attempting to keep the LBF fill ratio nearest to a designated value, and observing the LBF response for diverse precision scenarios.

reduced as the cells precision increases. The LBF capacity evaluation compares the number of tuples that may be inserted into the LBF for a given fill ratio and cell precision. As more tuples are added to the filter the number of occupied cells increases. It is expected that a fill ratio exceeding 70% will generate a significant degradation of the filter response, and cause the LBF saturation.

There is no deterministic manner to determine the number of tuples that need to be inserted for achieving an explicit filter density, *i.e.* fill ratio. To estimate the fill ratio, an iterative process inserts tuples into the LBF until the desired fill ratio is reached and counts the number inserted tuples. This iterative procedure is repeated 1000 times, to obtain the nearest integer to the average of all counters. This simple approach is sufficient to estimate the number of tuples to consider for each experience, without departing from the LBF fill ratio designated to each evaluation scenario. These scenarios contemplate the cell precision influence, *i.e.* the cells have a range of $1\,\text{bit} \leq \mathsf{b} \leq 16\,\text{bits}$ per cell, for introduce the precision variability component.

To evaluate the LBF capacity, the fill ratio was measured considering five scenarios (labelled as $\mathsf{Scen1}$ to $\mathsf{Scen5}$), intending to kept a constant fill ratio at pre-defined values, from 30% to 70%, as depicted in Figure 4.5. All these scenarios consider LBFs with $\mathsf{k} = 7$ $\mathsf{hash(x)}$ functions. At the end, and for validation proposes, the LBF fill ratio is computed and results are depicted in Figure 4.5 and it shows that the LBF occupancy is close to the intended constant fill ratio.

After making it possible to ensure an almost constant fill ratio, the LBF capacity was evaluated by measuring the number of tuples that could be inserted in the LBF, as increasing the confidence precision. The obtained results are depicted in Figure 4.6, showing the impact of the number of bits per cell in the LBF capacity. The capacity baseline corresponds to a fill ratio of 50%, according to the authors in [15]. The tests included some variations around the baseline, to compare the LBF capacity in most common situations.

Consulting Figure 4.6 for a precision of $\mathsf{b} = 4$ bits, the LBF baseline capacity is around 100 tuples. As expected, and comparing the distinct curves depicted in Figure 4.6, the fill ratio increases with the number of LBF tuples. However, the difference between distinct occupancy is mitigated as tuples increase their accuracy. The number of bits per cell (b) has a significant impact on the LBF capacity, suggesting that b calibration should balance precision and capacity.

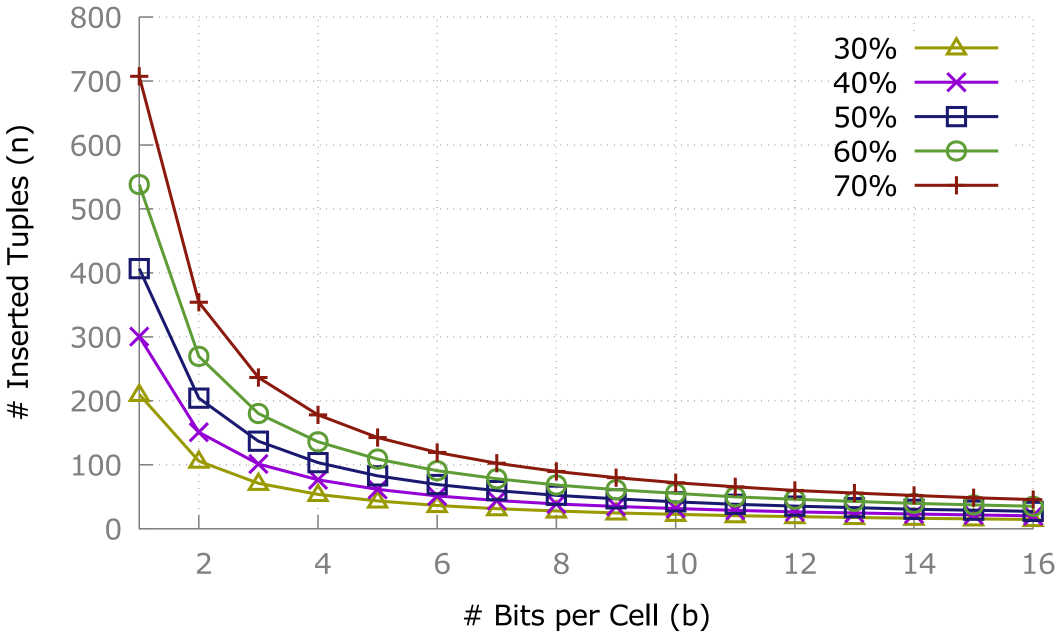


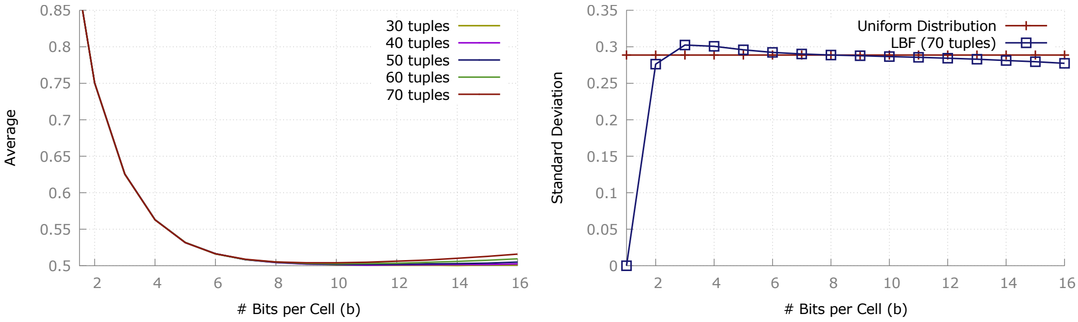
Figure 4.6: Impact in the LBF capacity caused by the number of precision bits (b), while controlling an almost constant fill ratio.

An analysis of Figure 4.6 shows that it is possible to estimate some bounds to the average capacity, useful to predict subsequent evaluation tests, without compromising the performance of the LBF.

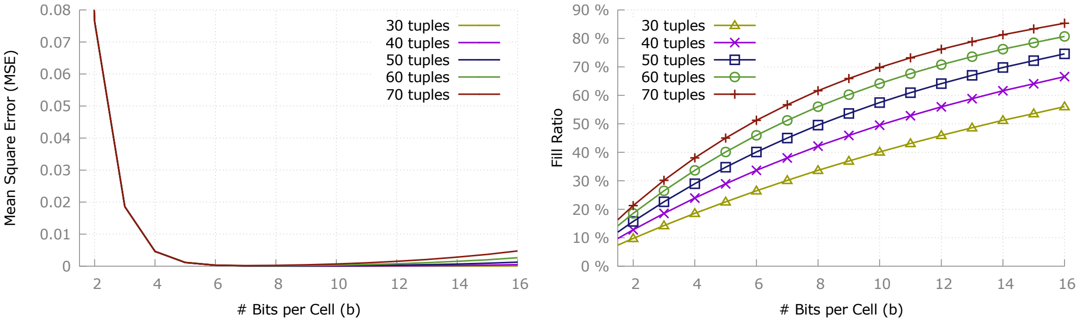
4.4.2 Quantization Precision Calibration

The quality of the LBF response also depends on the quantization levels used to represent the confidence value (c). The quantization levels are defined by the **precision** (b), which is the number of bits used in the binary representation (Section 4.3.2).

To evaluate how the precision calibration affects the LBF performance, some tests were performed by keeping a constant LBFs size of $\mathsf{dim}(\mathsf{LBF}) = \mathsf{4096}$ bits and using $\mathsf{k} = \mathsf{7}$ $\mathsf{hash}(\mathsf{x})$ functions, as given by Eq.(2.3). To evaluate the quality of the estimator $\hat{\mathsf{c}}$, five scenarios were chosen covering experiments from lower occupancy rates to near filter saturation. The scenarios submit the LBFs to the insertion of $\mathsf{30}$, $\mathsf{40}$, $\mathsf{50}$, $\mathsf{60}$ and $\mathsf{70}$ tuples. For each tuple, the confidence (c) assumes values uniformly distributed on the interval $[\mathsf{0}, \mathsf{1}]$, given by a standard uniform distribution $\mathsf{U}(\mathsf{0}, \mathsf{1})$. The $\mathsf{U}(\mathsf{0}, \mathsf{1})$ average value is $\frac{1}{2}$. To verify it in



(a) LBF average response (b) LBF response dispersion



(c) LBF Mean Square Error (MSE) (d) LBF density monitoring

Figure 4.7: Characterizing the precision bits (b) impact in the LBF ($\bar{\hat{c}}$) response.

practice, we consulted the deterministic data structure where all the generated c values are stored, and computed its average $\bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i = 0.5005$. Concluding that theoretical prediction is in line with $\bar{c}$ result, which is a good indicator of the $U(0,1)$ quality used during the performed experiments.

To minimize the disturbance caused by some event/feature that may arise when comparing the quality of the estimator (LBF), we measured the average of the estimator responses, *i.e.* $\bar{\hat{c}}$.

The evaluation results are depicted in Figure 4.7. Figure 4.7a depicts the estimator $\hat{c}$ average, showing its dependence on the number of bits used for quantization. The estimator $\bar{\hat{c}}$ departs from its theoretical prediction value given by the baseline $\overline{U(0,1)} = \frac{1}{2}$. For lower precision scenarios (*e.g.,* $b < 8$), Figure 4.7a clearly shows that the impact of the quantization is significant. However the number of inserted tuples is not relevant if above 50% of the LBF fill ratio (Figure 4.7d). The experiments show that there are no benefits in increasing the accuracy beyond 10 bits per cell. For higher precision scenarios the quantization error is reduced, although it also requires additional memory space per cell. Considering a rigid filter size, it reduces the number of the available cells and pushes the LBF to saturation. Thus, ratified by an $\bar{\hat{\mathsf{c}}}$ overestimation observed for $\mathsf{b} > 14$ (Figure 4.7a), where the average departs from the asymptotic theoretical value.

To quantify the dispersion of $\hat{\mathsf{c}}$ relatively to the LBF average ($\bar{\hat{\mathsf{c}}}$) response the standard deviation was measured, and the experiment results are depicted in Figure 4.7b. For clarity, Figure 4.7b only shows the 70 tuples insertion case. Figure 4.7b confirms our expectations that the dispersion of $\bar{\hat{\mathsf{c}}}$ is similar to the dispersion characteristic of a uniform distribution $\sigma_{\mathsf{U(0,1)}} \approx 0.289$, except in the case where $\mathsf{b} = 1$ (by using only a single bit, the dispersion is null).

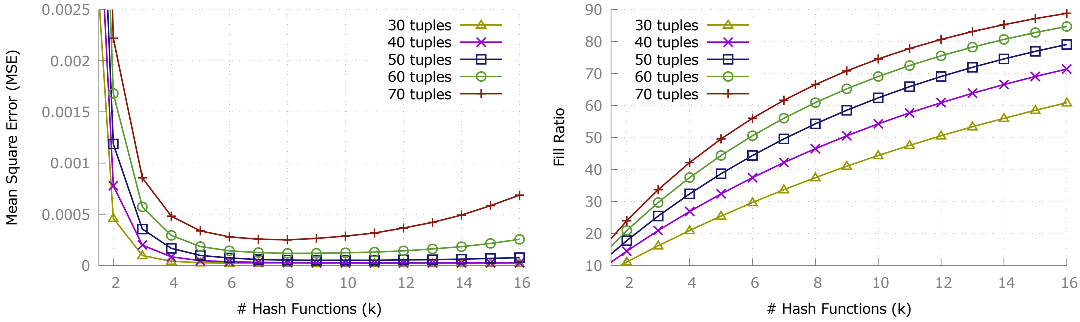
The Mean Square Error (MSE) obtained during the testing scenarios is depicted in Figure 4.7c. It clearly suggests the existence of a lower and an upper limit for the number of bits per cell, since the observed MSE is not significant when $5 \leq \mathsf{b} \leq 12$. To meet the scenarios conditions, it is enough to use the lower limit of $\mathsf{b} = 5$ bits per cell, to perform the c quantization and its respective binary representation.

Figure 4.7d compares the LBF fill ratio, in terms of the number of bits per cell b. Figures 4.7c−4.7d show that, in the cases studied, the observed error has distinct origins that have to be balanced: *i*) the error when $\mathsf{b} < 5$ is caused by quantization errors, since the LBF fill ratio is less than 50%, and *ii*) the error when $\mathsf{b} > 12$ bits (70 tuples) is mainly caused by a dense LBF, since the LBF fill ratio is over 75% and leading to the saturation zone.

4.4.3 Number of $\mathsf{hash(x)}$ Functions Calibration

As introduced in Section 2.4, each tuple in the LBF is indexed by $\mathsf{hash(x)}$ functions. The number of $\mathsf{hash(x)}$ functions, denoted as k is a key configuration parameter which balanced by the filter size, regulates the expected false positive rate of a traditional Bloom filter. The LBF responses will also be affected by the k selection, thus requiring an evaluation of its impact in the LBF response (Mean Square Error), and simultaneously controlling the LBF fill ratio.

To evaluate the impact caused by the chosen k $\mathsf{hash(x)}$ functions in the LBF quality of the estimated responses, we defined several scenarios where k assumes the integer values in the interval $2 \leq \mathsf{k} \leq 16$. All the scenarios consider a constant precision of $\mathsf{b} = 8$ bits per cell, and the range of inserted tuples is similar to the



(a) LBF response MSE (b) LBF density monitoring

Figure 4.8: Characterizing the number of $\mathsf{hash(x)}$ functions (k) impact in the LBF response.

previous LBF assessments (Sections 4.4.1 and 4.4.2).

The results upon varying k during the $\mathsf{N} = 100\,000$ experiments show that the average value of the estimator ($\bar{\hat{\mathsf{c}}}$) and its standard deviation ($\sigma_{\hat{\mathsf{c}}}$) remained practically constant. The statistic summary results for LBF responses are:

$$\bar{\hat{\mathsf{c}}} \simeq 0.5074 \pm 0.0051$$

$$\sigma_{\hat{\mathsf{c}}} \simeq 0.2882 \pm 0.0007$$

The filter response does not show significant variation on average and standard deviation, *i.e.* these metrics return similar results to the theoretical predictions for an uniform distribution $\mathsf{U(0,1)}$, where $\overline{\mathsf{U(0,1)}} = 0.5$ and $\sigma_{\mathsf{U(0,1)}} \simeq 0.2887$.

The impact on the LBF responses caused by the calibration of the parameter k was measured in terms of *i*) Mean Square Error (MSE) and *ii*) Fill Ratio, and the results are depicted in Figure 4.8. The error magnitude observed in Figure 4.8a is very small when compared to the MSE values from quantization component depicted in Figure 4.7c, thus suggesting that the LBF response is very reliable, without significant fluctuations caused by k calibration.

In contrast to theoretical optimization expressed by Eq.(2.3) in p. 45, results show that it is not advisable to increase too much the number of $\mathsf{hash(x)}$ functions. The LBF presents a reasonable performance where using just $\mathsf{2}$ or $\mathsf{3}$ $\mathsf{hash(x)}$ functions. The LBF overestimation error is mainly brought in by the quantization component. However, the theoretical value obtained by Eq.(2.3) may be confirmed by the results depicted in Figure 4.8a, which correspond to the zone where the MSE is minimal.

Reducing the number of $\mathsf{hash}(\mathsf{x})$ has some advantages: *i*) decrease the space occupied by each tuple in memory, and *ii*) reduces the amount of node computation. Figure 4.8b shows that, for the cases under consideration and $\mathsf{k} > 12$ the LBF is closer to saturation, and the MSE starts to increase (Figure 4.8a), which correspond to calibration conditions that should be avoided.

4.5 Discussion

The initial motivation to develop a LBF emerged from the need of an efficient data structure for information dissemination through WSNs. Nevertheless, the LBF reveals to be a more generic data structure, capable to embrace applications in other domains that depend upon imprecise data or uncertainty prediction membership.

The Linear Bloom Filter (LBF) is a new variation of a Bloom filter that maps each filter member elem with a confidence value in range $\mathsf{c} \in [0, 1]$, managing dynamic data while maintaining a constant size data structure in memory. The confidence ($\mathsf{c} \in [0, 1]$) parameter was initially proposed to represent the existence of an intuition (uncertainty) about records location in WSNs applications, and for that reason it could be set in any continuous interval. Our solution proposes the $[0, 1]$ interval, allowing for an affinity with a generic probability concept. Other interesting feature of the confidence parameter is its ability to be set with different values for the same element in distinct nodes on which it is registered, which will be explored in Chapter 5.

The LBF estimator principle contrasts with other techniques that use deterministic data structures or Bloom filter arrays approaches that require greater memory capacity to cope with frequently network topology changes. The LBF dimension is invariant with the number of added records and thus with the network size, evidencing grater applicability for WSNs membership dissemination protocols. Due to the stable LBF dimension, WSNs protocols may easily control the LBF impact in the packet size when include a LBF during message dissemination. The calibration parameters should be adjusted to meet the application requirements. An overestimated calibration will waste memory space which is a critical resource for WSNs. Choosing bigger LBFs will have a negative impact in communication, when the LBFs is included on every message for multi-hop dissemination.

Evaluating the LBF performance is a sensitivity analysis to characterize the LBF parameters calibration. The results show that a proper b and k calibration limits the LBF expected errors, thus benefiting a satisfactory estimation when comparing the LBF response with a deterministic data structure. However, the LBF estimator error is minimally influenced when $\mathsf{k} \geq 2$, conceding the lead role in the error influence to b calibration. In LBF, the quantization error is not negligible when compared to typically accepted errors by other probabilistic data structures. When the WSNs operation conditions change significantly, a new calibration should be performed, thus avoiding the filter saturation or an oversized memory allocation.

In future, we intend to dynamically adapt the LBF dimension to the expected number of tuples, using an expansible approach in line with the *scalable* Bloom filter technique [15].

Chapter 5

ABC: Adaptive Broadcast Cancellation

Usually, uninformed search mechanisms explore the network connectivity, expecting to find the location of a given target record without knowing any information related with the network topology. Chapter 2 revisited relevant approaches for energy-efficient uninformed search mechanisms specifically developed to cope with unstructured wireless networks operational conditions.

The Broadcast Cancellation Initiated on Resource (BCIR) algorithm proposed in Chapter 3 was developed to stop a query propagation, as soon as the record is found, improving the energy efficiency of uninformed search mechanisms [160]. However, the BCIR algorithm does not consider the knowledge gathered with previous searches. On the other hand, it is expectable that having an intuition about record location is better than not knowing any context information at all.

Assuming some network stability, previous queries may contribute with some knowledge helping to redirect a query propagation to locations with higher probability of finding the target records. Moreover, an opportunistic approach of the broadcast retransmissions may propagate location knowledge among neighbour nodes, without considerably increasing the communications costs. Inspired by previous notions, this chapter presents the new **Adaptive Broadcast Cancellation (ABC)** search mechanism to improve the search performance using the acquired location intuition.

ABC adopted the Linear Bloom Filter (LBF) mechanism described in Chapter 4 to accommodate and propagate information in its local vicinity, thus contributing for an efficient utilization of the bandwidth and node limited memory. ABC assumes that acquiring some intuition about record location is useful to accelerate the relaying operations on nodes that have a higher probability to be involved in a successful discovery, thus expecting to improve latency and energy consumption. ABC opportunistically takes advantage of the query propagation mechanism to exchange context information, by piggybacking LBFs embedded on every exchanged messages. ABC operates without increasing the number of messages to be retransmitted, although assuming a small penalty in message size.

The ABC algorithm was evaluated under several random network scenarios, using the *ns-2* network simulator, one of the most popular simulators accepted by the Mobile Ad-hoc NETworks (MANETs) research community [146]. We extend the *ns-2* to support the new ABC algorithm and its competitors approaches. Evaluation shows that with ABC it is possible to speed up the searching tasks without compromising energy efficiency. The ABC algorithm may significantly improve the retransmission ratio and simultaneously reduce latency, specially in long term scenarios where search queries are frequently repeated.

The remainder of this chapter is organized as follows: first, the motivation section alert to the facts that prior searches may benefit the performance of the subsequent ones, and supporting that perception in the related work section; it proceeds describing the ABC algorithm, that represents distance with a probabilistic data structure (LBF), in line with some gradient decaying algorithms; the evaluation section goes on to report the experiments setup and their results analysis. The chapter ends with a summary discussion.

5.1 Motivation

Wireless Sensors Networks (WSNs) applications are usually based on the periodic report communication model, increasing their resilience against erratically node failures, by favouring the discovery of alternative communication paths between nodes [86, 189]. However, they exhibit an inefficient performance for multi-hop networks, because they are too resource intensive (*e.g.,* memory, CPU and power), especially when there is no new data to be retransmitted. The communication systems may be even more demanding when sensors are deployed in environments with adverse operational physical conditions. The network topologies are very dynamic due to sensor faults, link failures, interferences and environment conditions. Static solutions unfit to these scenarios and wireless sensor-based applications are privileged, mainly driven by the flexibility to adapt the network operability to a changing environment.

To embrace those challenges many systems based on MANETs or WSNs are being developed for monitoring and locating resources in adverse scenarios. Examples of those conditions may be found in mining and extraction companies, which increasingly use wireless sensors to gather data in extreme operating environments [153]. Nowadays, low cost wireless sensors may be used to deploy unstructured networks in underground Mines, as it: *i*) avoids extending copper wires over several kilometres, *ii*) allows for dynamic network topology, *iii*) adapts the reporting sampling rate to the physical phenomena event detection and *iv*) trigger event alarms in a efficient way [163, 174].

Tracking and monitoring are some of the major application examples that use MANETs or WSNs to support event detections related to people, vehicles, fires, seismic or other physical phenomena. A search mechanism is a fundamental component and it is not just about finding records on the network. It must improve the performance of broadcast primitives and the information gathering processes. The search mechanism should match the requirements for physical data sensing and efficiently managing how this data is aggregated by the network, while always being aware of the sensor constraints and application requirements.

Location and position based on broadcasting techniques are explored by several scalable search mechanisms such as LAR [138] or DREAM [29] protocols. As in previous protocols, nodes may exchange location information to reduce the number of participants in searching processes, and delimit the query propagation to a geographical region [137]. Instead of managing a vast amount of network information, as compared to uninformed search techniques, they rely on precise position information. However, location-limited techniques based on external sources, such as GPS, are vulnerable to: *i*) inoperative GPS device, *ii*) difficulties with signal reception, or *iii*) inaccurate devices. In contrast, limiting the query expansion by managing the knowledge acquired during previous queries does not require any special device, avoiding faults and the dependency on external localization sources.

The **main motivation** to the work presented in this chapter results from a very simple intuition. Although the network topology is constantly changing and records can migrate to other neighbours or even disappear, it is expected that previous search queries contain valuable information, concerning the location of the target record and network topology, that should not be discarded.

Focusing on an underground Mine monitored by an unstructured network and considering it as the main motivation scenario, some of the main key issues can emerge with a simple question: *"Where is the excavator tool?"*. Assuming that the *"excavator"* is lost inside the Mine and sensors did not detect it in the expected place, answering that question may become a complex task.

The system may query the sensor network, trying to find a physical item, by searching for the respective RFID tag record. This is an example commonly addressed by uninformed search approaches. However, an underground Mine may have thousands of catalogued equipments, some of those often abandoned by workers. Thus, similar questions need to be answered or updated frequently and multiple queries are expected to be executed.

It should be noted that when searching for a target record, the query will contact intermediate nodes during its propagation, creating a chance to find other records. Having a mechanism to manage the gathered information may be very useful for predicting record locations. Thus, subsequent queries may run faster when an intuition on the record location is known.

There is a strong motivation to improve search mechanisms that rely only on previous exchanged messages, in a context where each node announces the hosted records. In particular, by noticing that performance improves with the number of exchanged messages [164]. Recently proposed search mechanisms are exploring gradient-based techniques that exponentially decay the information while propagating it to other nodes [100]. However, to the extent of our knowledge, there is no energy-time efficient search mechanism that considers information gathering and aggregation while forwarding data in a multi-hop network, which are some of the major concerns for unstructured networks.

5.2 Related Work

There is a large amount of work on informed search mechanisms, much of which has been analysed in [1, 270]. The significant related work was categorized in Section 2.3 (p. 40) and on continuation, the most representative algorithms for searching on unstructured wireless sensor networks are presented.

The following works propose efficient broadcasting techniques when compared with simple flooding approaches, the main goal of which is to reduce the number of retransmissions, while attempting to ensure that the search mechanisms eventually discover the record location.

The idea to introduce a measurable heuristic to decide which node should retransmit and to dynamically guide a searching task was presented in [60]. Authors consider that most of management and scalability issues in large-scale networks may be addressed by selecting which nodes should be contacted. Reducing the contacted nodes can drastically reduce the search mechanisms latency. It also enables progressive accuracy, based on measurements from several nodes, by incrementally computing a cost index, designated as *"belief"*. The paper [60] presents the Information-Driven Sensor Querying (IDSQ) algorithm, that uses a compact representation of the *"belief"* state, together with the search query, allowing to incrementally update the *"belief"* during the query propagation. Their approach enables the activation of nodes when there are events to report, and only nodes from those parts of the network will retransmit the query messages, discarding unnecessary information retransmissions. Although the paper equally proposes an incremental algorithm to the *"belief"* estimation problem, IDSQ uses mainly a theoretical approach.

In contrast with uninformed search mechanisms (Section 2.3), the FResher Encounter SearcH (FRESH) protocol proposed in [73], improves the performance of the Expanding Ring Search (ERS) mechanisms [48,125]. The FRESH approach explores the history of last encounters between neighbour nodes. Every node maintains the timestamps of its last encounter with every other node, in a self sensing approach. The encounters timestamps are used to guide the searching towards a given record, by replacing a single wide network search with a succession of smaller searches. In FRESH, instead of searching for the node hosting the record, the search goes for any intermediate node that encountered the record more recently than did the node itself. The intermediate node then searches for a node that encountered the record even more recently, iterating until the node owning the record is discovered. Using small searches of this king reduces the number of contacted nodes and message retransmissions, improving its energy efficiency, when compared to flood-based approaches.

Some protocols (*e.g.,* FRESH) explores the broadcast relaying of HELLO messages, fetching the role of control packets, as originally described in the AODV protocol, used for gathering neighbour information [211]. In several situations, MANETs are stable for short time periods. During those intervals, the network topology is mainly static and the HELLO messages may be used to collect neighbour node information, which is kept locally on each node. However, the excessive broadcast of HELLO messages has a negative impact on the number of retransmissions, *i.e.* they are responsible for excessive energy consumption, motivating the research for new approaches to suppress unnecessary HELLO messages [107].

Unlike flooding, the Weighted Rough Set (WRS) [192] is a broadcasting protocol that reduces the retransmissions redundancy and expedites the propagation of query messages, by performing a *2-hop* away neighbour informed searching.

The random walk [168] uses a random content discovery based strategy to search for target records. An adaptive method to optimize the random walk protocol was proposed in [22]. The optimization procedure uses a gradient descend technique. It computes a cost function which accounts three parameters *i*) hit rate, expressing the percent of query success, *ii*) response time, which is the time spent in finding a given record, and *iii*) energy consumption, that considers four analytical models to represent the corresponding states of energy consumption (reception, transmission, idle and sleep). The optimized random walk protocol, proposed in [22] uses a pure analytical approach, however it suffer from a lack of evaluation. It could benefit from a more realistic simulation to test its performance.

Every physical event results in a natural information gradient in the proximity of the phenomenon [82]. This fact triggered the development of efficient informed searching mechanisms for unstructured networks. One such example is the directed diffusion approach presented in [120]. It uses a simple remote surveillance application example to show how nodes may coordinate together and perform in a distributed sensing environment using a publish/subscribe data-centric approach. Previous ideas evolved into the concept of *information-directed*, where the objective is to minimize communication cost while maximizing the information gain [166]. It measures the quality of communication around sensors and considers two common information extraction patterns, forwarding the query to *i*) a pre-specified node or *ii*) an arbitrary node in its vicinity.

The GRAdient Broadcast (GRAB) [279] uses a gradient approach, and was designed specifically for robust data delivery. GRAB is capable of coping with unreliable nodes and fallible wireless links. It builds and maintains a cost function, providing for each node a favourable direction (gradient) to forward the query message. The initiator node simply transmits a message to the radio channel without designating any neighbour as the next hop. Each receiving node independently decides whether it should further forward the message, towards the expected record direction.

The idea of using *attenuated Bloom filters* to manage some location perception and use it to direct the queries propagation towards to records was originally described in [227]. Their design is based on a vector of standard binary Bloom filters data structures working as a lossy distributed index, similar to the gradient concept. Despite their applicability (as a probabilistic document localization algorithm) being out of this dissertation scope, the same principles may be applied for searching records on wireless networks.

Other algorithms adopt gradient approaches to estimate the preferred directions to search for records in a multi-hop network, such as the *Wader* [100]. The *Wader* strategy uses a Bloom filter to manage local membership information and exponentially decay it within a given range. The filter decay mechanism consist of randomly setting some bits to 0, as filters are retransmitted (multi-hop), so that distant records have a lower representation in the local filter. They also derive the necessary and sufficient condition for a feasible gradient-based routing mechanism and demonstrate that *Wader* based searching mechanisms are efficient, revealing a high probability of finding the target records.

Some sensor networks use periodic reporting with intensive loads of queries for information discovery. The size of the network creates additional scalability problems. A strategy to efficiently manage a large amount of data is to employ probabilistic data structures, as described in Chapter 4. From the related review, a common implementation strategy emerges. Nodes use a Bloom filter to manage the membership information of its record items, *i.e.* whether or not a record is located at that particular node. Our contribution, the Linear Bloom Filter (LBF) [162], is a Bloom filter [33] generalization that proposes a memory-efficient data structure suitable to manage hosted records information.

The uninformed search mechanisms may operate without global topological information (*e.g.,* diameter and size) to cope with large multi-hop networks, and still prevent an uncontrolled number of message retransmissions, as proposed in our contribution [160]. In general, uninformed searches equally explore every link between nodes. However, after performing a few queries, the search mechanism may acquire an intuition of the target record locations and use it to accelerate the success of a given query. Propagating a decaying LBF through the network may be used to implement a gradient-based search mechanism. This notion potentiates the development of a new adaptive algorithm to efficiently speed up the searching towards the expected record location, as described in the next section.

5.3 ABC: Algorithm Description

As discussed in Chapter 3, the cancellation approach produces a linear increase of delay with the hop distance from the query source [160, 203]. Increasing delay is required to create a time window opportunity to accommodate the cancellation propagation for stopping an ongoing broadcast, paving the way to reducing the retransmission ratio at the expense of some time efficiency. BERS* [203] and BCIR* [160] are good examples of algorithms with such characteristics. Nevertheless, none of those algorithms get advantage of the knowledge obtained during previous queries to improve the efficiency of the next one.

This section describes the **Adaptive Broadcast Cancellation (ABC)** algorithm. Assuming some topology stability, ABC estimates some record location intuition, which is useful to improve the performance of the query dissemination [120]. The ABC algorithm abandons the blind delay increase approach, by tuning it with the knowledge gained from previous queries. The key idea is having nodes that propagate queries faster and towards the expected record location. In contrast, when the relaying by a given node is presumed to be less relevant for a particular query, the delay is kept at an adequate level to facilitate the cancellation process.

In order for ABC to manage scalability problems resulting from variable and large diameter networks, we chose a solution that adopts a probabilistic distance representation, by assuming an acceptable uncertainty to the expected record location.

5.3.1 Probabilistic Distance Representation

Monitoring the location of physical items within a wide geographic area involves the placement of a large number of sensors to cover the deployment area and therefore rising the intrinsic scalability problems. Deterministic data structures are unable to cope in an efficient manner with big datasets and they do not fit directly in the sensor memory. As an example, consider a Mine scenario where a given sensor node detects the RFIDs tags that are physically attached to Mine equipment. Figure 5.1[1] depicts a photograph of the RFID reader and another photograph of the robust RFID tag placed on a Mine trunk. In this scenario, each physical item (*e.g.,* Mine trunk) is uniquely identified, and represented in a tracking application by its respective record.

The ABC strategy abstracts the knowledge about physical items location on a space-efficient probabilistic data structure named as **Linear Bloom Filter (LBF)** [162] and described in Section 4.3. Summing up, the LBF is a variation

(a) Mine: RFID reader in a tunel (b) Mine: RFID tag in a mine truck

Figure 5.1: Physical items from real world mapped to local records as discovered by sensor nodes.

---

[1] Photos from remote asset tracking system in UCILs Narwapahar Mine in India

of standard Bloom filter that, instead of operating exclusively with binary values, introduces the possibility to assign a totality ordered confidence value $\mathsf{c} \in [0, 1]$ for each record in a set. A confidence c index equal to 0 represents the absence of knowledge and 1 the highest confidence level matching a local record, *i.e.* the node is hosting a record and the confidence is set to its maximum value, informing that the location of the record is the node itself. The intermediate confidence values result from a decaying mechanism, to be described below.

Consider a Mine scenario where a sensor node using a RFID reader detects the presence of a truck when it recognises its RFID tag. The truck RFID tag may be represented as $\mathsf{recordA}$. The node can manage the truck location by inserting $\mathsf{recordA}$ into its memory using a LBF data structure and setting its confidence value to its maximum, *i.e.* $\mathsf{c} = 1$ and representing it as a local record.

For visual simplicity consider a small size LBF with $\mathsf{m} = 16$ cells and $\mathsf{k} = 3$ $\mathsf{hash}(\mathsf{x})$ functions, as depicted in Figure 5.2. Note that LBF is previously initialized empty of any records, with all the array cells content set to 0.

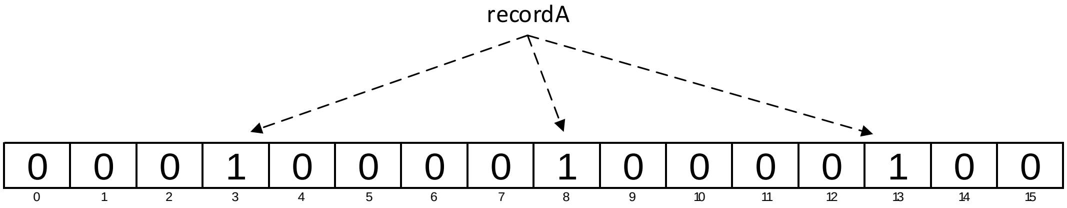


Figure 5.2: ABC inserts a local record $\{\mathsf{recordA}\}$ into the node memory, handled on a $\mathsf{LBF}[\,]$ and for better readability, the represented set starts empty (no previous data).

The insert operation of a local record, for example $\mathsf{recordA}$, which corresponds to tuple $(\mathsf{recordA,1})$, uses the function $\mathsf{INS}(\mathsf{recordA}, 1, \mathsf{lLBF}[\,])$ from Algorithm 4 (p. 116). After applying $\mathsf{k} = 3$ $\mathsf{hash}(\mathsf{x})$ functions to $\mathsf{recordA}$, it generates the LBF indexes cells to set, *e.g.,* $\mathsf{hash_k}(\mathsf{recordA}) = \{3, 8, 13\}$.

Choosing an array for $\mathsf{lLBF}[\,]$ memory representation, the previous indexes are the array positions to set with a maximum confidence index $\mathsf{c} = 1$ (local record), as depicted in Figure 5.2. On demand, any $\mathsf{node(i)}$ may invoke the function $\mathsf{QRY}(\mathsf{recordA}, \mathsf{lLBF}[\,])$ from Algorithm 4 described in Section 4.3.1, to consult a local $\mathsf{lLBF}[\,]$ for an estimation of the confidence value $(\hat{\mathsf{c}})$, assigned to a particular record (*e.g.,* $\mathsf{recordA}$).

Algorithm 6: Decaying information with distance

```
const att ← 0.9;                                  // Attenuation factor
Function mergeLBF(lLBF[ ], rLBF[ ])
    for i ← 1 to rLBF.size do
        lLBF[i] ← max(lLBF[i], rLBF[i] × att);
    end
```

5.3.2 Distance Decaying Data

In physical systems whose properties are defined by undifferentiated entities, the contribution from near entities is superior to those from the most distant ones. Moreover, when it is acceptable to lose some information, the decision with the lowest impact is to ignore the consequences of an distant entity rather than a near one.

Assuming that the previous principle is valid for multi-hop networks, ABC adopts a similar strategy, implementing a **decaying** mechanism that attenuates the information with distance. To accomplish that, the ABC extends the LBF core operations, described in Algorithm 4 (p. 116), with a new function designated as mergeLBF(lLBF[ ], rLBF[ ]) and described in Algorithm 6.

The rLBF[ ] is the filter received from a neighbour, possibly using a broadcast message. The Algorithm 6 combines node local information (lLBF[ ]) with the received information from a neighbour node (rLBF[ ]) that is attenuated by a constant factor att lower than 1, thus privileging local information.

To clarify the ABC information decaying component and its dissemination, focus on a single node with two neighbours as depicted in Figure 5.3. Considering that node(30) initially only knows its hosted records (recordA) as represented in the lLBF[ ] depicted in Figure 5.2. When node(30) receives the broadcast messages sent from its neighbours, it will access the rLBF[ ], and apply the mergeLBF(lLBF[ ], rLBF[ ]) decaying operation described in Algorithm 6. The process is repeated for all received messages. According to the multi-hop relaying schedule, node(30) will retransmit the search message and opportunistically include the lLBF[ ] in the message header, disseminating its knowledge representation. Note that lLBF[ ] has information about node hosted records and the attenuated data received from its neighbours.

Figure 5.3 illustrates the decaying operations implemented by ABC, giving and example of LBFs representations. The top of the figure shows the lLBFs[ ]

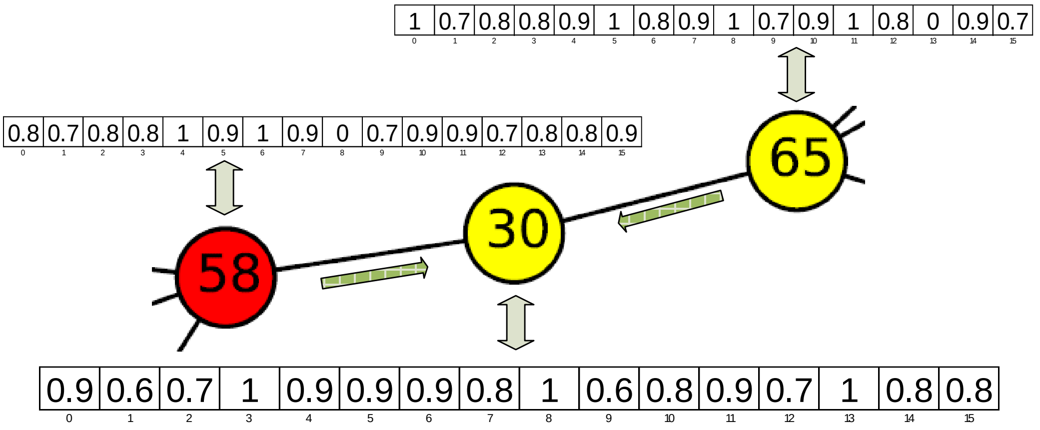


Figure 5.3: ABC decaying mechanism, node(30) hosts {recordA}; node(30) decays the received data from its neighbours, and its local LBF[ ] is updated.

(local filters) for both node(58) and node(65). Both nodes, {node(58), node(65)}, asynchronously broadcast their messages, that will be received by node(30). The bottom of the figure depicts the lLBF[ ] internal values, hosted by node(30), after applying the decaying dissemination mechanism using Algorithm 6.

The query mechanism may benefit from the available local (lLBF[ ]) information. Considering the merged lLBF[ ] stored in node(30) as depicted in Figure 5.3, querying for recordA in node(30) returns $\mathsf{c} = 1$, thus confirming that recordA is hosted by node(30). When querying for a given target record, hosted by node(58) or node(65), and the lLBF[ ] estimates a high confidence value (*e.g.,* $\mathsf{c} = 0.9$), it suggests that the record is located near the query source node. When the query estimates (consulting lLBF[ ]) a lower confidence value (*e.g.,* $\mathsf{c} = 0.6$), it suggests that the record may be hosted by a distant node.

The ABC progressively attenuates the confidence values by att (Algorithm 6 in p. 143), as the LBF (included in the message header) is gossiped in a multi-hop dissemination. A record that is located at hop distance, has an expected confidence level of $\mathsf{c} = \mathsf{att}^{\mathsf{hop}}$. At any time, a node(i) can consult its local filter (lLBF[ ]), which is a probabilistic distance representation, by query for target record and be able to infer the expected record hop distance. As successive queries are executed and multiple messages are relayed, the lLBF[ ] distance prediction is similar to a **gradient**, giving an estimation of the best direction to go searching.

5.3.3 Query Dissemination

Node relaying is a basic primitive operation for multi-hop WSNs and is responsible for node communications through long distances. The network communication model is asynchronous and nodes autonomously decide when to relay a message.

The ABC uses the relaying mechanism for query dissemination. The ABC assumes that a node(i) can have three primitive actions related with each query, it can *i*) start a new query by broadcasting a search message ($\mathsf{m_s}$), and *ii*) initiate the cancellation process by sending a cancellation message ($\mathsf{m_c}$), or *iii*) forward either or both messages $\mathsf{m_s}$ and $\mathsf{m_c}$. Messages $\mathsf{m_s}$ and $\mathsf{m_c}$ are tagged with a queryID, so that node(i) can track similar but distinct concurrent queries, avoiding message duplication and spurious retransmissions propagating over the WSN.

Initially, all the LBFs are empty (there is no past). When node(i) starts a new query it creates a new packet following Algorithm 7. Each node(i) only knows its hosted records, which are represented in the local filter lLBF[ ]. A node(i) copies the lLBF[ ] into the $\mathsf{m_s}$ message header. The query is launched by broadcasting message $\mathsf{m_s}$, thus initiating the dissemination process. The first query broadcast is sent without any record information, except the node hosted records.

5.3.4 Query Handling

Once node(i) receives a message it initiates the handling that the ABC protocol provides to that message. The first operation distinguishes the message type: *i*) a search message ($\mathsf{m_s}$), or *ii*) a cancellation message ($\mathsf{m_c}$). These two type of messages have distinct handling, as described in Algorithm 8, depending on their functions handleQuery(msg) and handleCancellation(msg) respectively. Both functions confirm the message validity and discard it when *i*) a cancellation for this message was already received or *ii*) it is a duplicate.

Afterwords, the handleQuery(msg) may produce two outcomes. When node(i) hosts the corresponding record it starts the cancellation mechanism by broadcasting $\mathsf{m_c}$ which also propagate the query answer, otherwise, node(i) schedules the search message ($\mathsf{m_s}$) for retransmission. The details on added delay are described in Section 5.3.5.

After receiving $\mathsf{m_c}$ a node(i) triggers the handleCancellation(msg) execution

Algorithm 7: Query starting

```
begin
    pkt $\leftarrow$ createPacket($\cdots$);
    queryList $\leftarrow$ queryList $\cup$ {msg.queryID};
    pkt.bloom $\leftarrow$ node.lLBF;
    send(pkt);
end
```

Algorithm 8: Query handling

```
Function handleQuery(msg)
    mergeLBF(node.lLBF, msg.rLBF);
    pkt.bloom $\leftarrow$ node.lLBF;
    if msg.queryID $\notin$ queryList then
        queryList $\leftarrow$ queryList $\cup$ {msg.queryID};
        if queryLBF(msg.record, lLBF) $= 1.0$ then
            cancelList $\leftarrow$ cancelList $\cup$ {msg.queryID};
            send(CancelPacket(pkt));
        else
            time $\leftarrow$ calcDelay(msg.header, ABC);
            insertTxQueue(time, pkt);
        end
    end
Function handleCancellation(msg)
    mergeLBF(node.lLBF, msg.rLBF);
    pkt.bloom $\leftarrow$ node.lLBF;
    if IsInTxQueue(pkt) then
        removeTxQueue(pkt);
    else
        if node.HostsRecord(msg.record) $\wedge$ (node.recordFound[msg.queryID] $=$ false)
          then
            node.recordFound[msg.queryID] $\leftarrow$ true;
            msg.recordHost[msg.queryID] $\leftarrow$ node.ID;
        end
        if (msg.queryID $\notin$ cancelList) $\wedge$ (msg.queryID $\in$ queryList) then
            time $\leftarrow$ calcDelay(msg.header, FLOOD);
            insertTxQueue(time, pkt);
        end
        cancelList $\leftarrow$ cancelList $\cup$ {msg.queryID};
    end
```

that discards all future schedule $\mathsf{m_s}$ retransmissions, giving priority to the cancellation. The query finds the record location when it reaches a node hosting the target record. The cancellation mechanism is initiated and the collected informations about the target record location are included in the $\mathsf{m_c}$ header. For the same query, *i.e.* messages referencing the same queryID, every node that relayed a $\mathsf{m_s}$, will relay the first $\mathsf{m_c}$ received, providing its contribution to the cancellation dissemination without adding any intentional delay. The query successfully delivers the answer when the first $\mathsf{m_c}$ (piggybacking the record location) reaches the initiator node.

Similarly to BERS* [203] and BCIR* [160] approaches, the ABC progressively increases the query propagation delay during the target discovery process, facilitating the cancellation messages dissemination until the record target is found. This is reflected in the query handling procedure depicted in Algorithm 8.

As nodes forward the received messages, ABC implements a learning mechanism, where nodes update its local knowledge representation lLBF[ ] with the received information from its neighbours rLBF[ ]. The core of the learning mechanism is performed by functions handleQuery(msg) and handleCancellation(msg) from Algorithm 8, which implement the message relaying mechanism. Both functions invoke mergeLBF(lLBF[ ],rLBF[ ]), decaying the received data rLBF[ ] and updating the node local context lLBF[ ].

The gradient information corresponds to an expected record distance, probabilistically represented in the lLBF[ ]. The relaying operations and the ABC learning mechanism disseminate and update the gradient information respectively. ABC is capable of disseminating gradient information, *i.e.* lLBF[ ], without knowing the number of nodes or topological properties, such as network diameter.

Learning with a piggybacking approach prevents the dissemination of additional messages, saving energy otherwise spent on unnecessary retransmissions. The increase in size of the payload resulting from the piggybacking is not significant and is limited to the LBF size, making it invariant with the network diameter.

5.3.5 Gradient-based Adaptive Delay

The ABC adaptive delay applied during the query dissemination follows Algorithm 9 and considers three components. The record location independent component, given by a static value (DELAY_MIN) plus some jitter modelled by a random uniform distribution. The location dependent component, that makes the delay proportional to the message hop counter (Algorithm 8), to facilitate message cancellation. The third component is dictated by the nodes estimation of their distance to the record, given by the LBF gradient prediction. This

Algorithm 9: Gradient-based adaptive delay

```
1  const DELAY_MIN ;                                          // Minimum delay
2  const JIT_MAX ;                                            // Maximum jitter
3  Function calcDelay(msg, proto)
4      jitter ← runif(0, JIT_MAX);                            // Random Uniform
5      $time_{FLOOD} \leftarrow DELAY\_MIN + jitter$;
6      if proto = FLOOD then
7          return now() + $time_{FLOOD}$
8      end
9      if proto = ABC then
10         $time_{BCIR^{\star}} \leftarrow \max(2 \cdot (msg.hop - 1) \cdot time_{FLOOD}, time_{FLOOD})$;
11         $c_1 \leftarrow$ QRY(msg.record, msg.rLBF);
12         $c_2 \leftarrow$ QRY(msg.record, node.lLBF);
13         if $c_1 < c_2$ then
14             $time_{ABC} \leftarrow \max(2 \cdot (msg.hop - 1) \cdot DELAY\_MIN \cdot (1 - c_2) + jitter, time_{FLOOD})$;
15         else
16             $time_{ABC} \leftarrow time_{BCIR^{\star}}$;
17         end
18         return now() + $time_{ABC}$;
19     end
```

component given by $(1 - c_2)$ in Algorithm 9, that aims to speed up the query propagation in the predicted direction of the hosting node.

The calcDelay(msg, ABC) function in Algorithm 9 returns the schedule time for message retransmission, by considering these three delay components. This function is responsible for adapting the intentional added delay by comparing the estimation distance to the target record from two sources: *i*) node, represented by c_2 and *ii*) message header, represented by c_1. When $c_1 < c_2$ the expected distance locally estimated in the node is shorter than the expected distance extracted from the received message header, thus the added delay is reduced (Algorithm 9, line 14).

In ABC, the LBFs predictions is useful for accelerating queries towards the target record, producing a behaviour similar to a gradient. In this context and during the query dissemination a gradient-based adaptive approach adjusts the intentional added delay towards the query to the expected hosting node. The next example shows the ABC gradient-based functionality. ABC uses LBFs predictions, represented by the confidence parameter c, to estimate the distance to the node hosting the record, and applying an adaptive delay before relaying the search message.

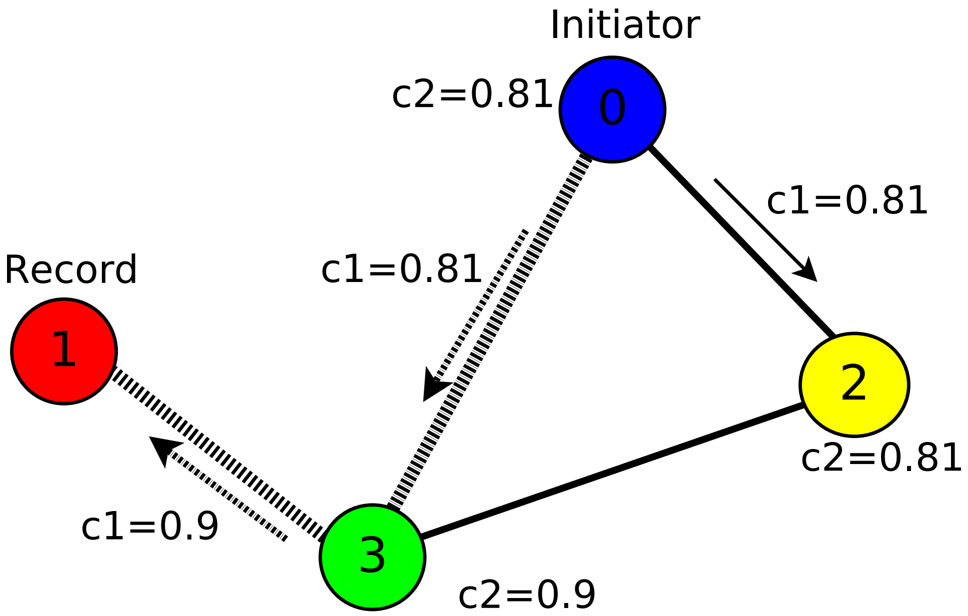


Figure 5.4: Gradient-based adaptive delay, speeding up the query towards the hosting node.

Figure 5.4 illustrates how ABC accelerates the query propagation into the most likely direction. In this example it is assumed that LBFs were populated by some previous queries executions. It considers that $\mathsf{node}_0(0)$ starts looking for a record hosted at $\mathsf{node}_H(1)$ and assumes that nodes know the confidence c associated with the expected hop distance. When $\mathsf{node}_1(3)$ receives m_s the adaptive delay component tests whether $\mathsf{c}_1 < \mathsf{c}_2$ is true. Therefore, an attenuation of $(1 - \mathsf{c}_2)$ is applied to the delay, speeding up the query towards $\mathsf{node}_2(1)$. On the other hand, given that $\mathsf{node}_1(2)$ receives m_s and $\mathsf{c}_1 \geq \mathsf{c}_2$, leads to decide that the delay suffers no reduction, *i.e.* it continues with a hop count proportional delay. After the record is discovered, $\mathsf{node}(1)$ initiates the cancellation process taking the opportunity to prevent $\mathsf{node}(2)$ retransmission.

5.4 Evaluation (ABC)

The evaluation follows the guidelines presented in the research methodology (Section 1.3). We decided to abandon the assumption of an idealised network environment, used during the preliminary theoretical predictions, due to the fact that wireless network communications are affected by the several conditions (*e.g.,* devices technology, radio propagation, distance, interference, signal sharing, spectrum limitations, antennas, signal power, communications load).

The two main possibilities for ABC protocol evaluation are: *i*) experimentation in real world deployments or *ii*) use simulation techniques in virtual environments. The main reasons for performing ABC evaluation based on simulation tools are: *i*) having to consider hundreds of sensor nodes makes experiments in real world very expensive, *ii*) it consumes a lot of resources (*e.g.,* time) to maintain physical sensors operational and removing errors from protocols, *iii*) simulation allows to test protocols under controlled conditions, creating scenarios representing unfavourable environments, such as found in underground Mines and *iv*) the simulation experiments can be repeated several times in the same scenarios, adjusting the parameters under evaluation.

There are several tools available for wireless networks simulation that were surveyed in [57]. To perform ABC evaluation, the elected simulation tool should be able to reproduce multi-hop unstructured wireless communications for MANETs and WSNs applications. Furthermore, it must have models for reproducing realistic broadcast communication scenarios, avoiding the influence of any routing or transport protocols. One of the most used simulation tools with such characteristics is the *ns-2* [267].

For that reason, the ABC evaluation was conducted using the *ns-2* simulator, which is a simulation tool capable of exposing some discrepancies from pure theoretical results [178]. The *ns-2* is an open source free tool, where the protocol stack is written in `C++`, enabling the addition of new protocols with low-level control of the internal functionalities. Models in *ns-2* are capable of reproducing realistic scenario conditions that include radio propagation and packets loss, not available in the SRS (p. 51). ABC protocol was evaluated with an asynchronous network communication model, using the *ns-2*[2] most recent version.

The *ns-2* simulator was extended to support the implementation of our new protocols (BCIR, BCIR*, ABC) and also other competitive alternatives (BERS, BERS*), that are not included in the simulator generic distribution. The developed code and its integration with *ns-2* is hosted in the `https://github.com/rmlima/BCUN` repository [158].

---

[2] ns-allinone-2.35, release date: 4^{th} November 2011

The initial observations were sufficient to motivate that some explanations before go further with evaluation details, such as:

Collisions and Partitions In most communication standards used by several wireless devices, such as the IEEE 802.11, broadcast messages are transmitted without expecting a confirmation sent by destination nodes. Collisions between transmissions may prevent a given node receiving a particular transmissions, even through the medium access controlling mechanisms are able to retransmit some packets or defer their transmissions in time, as an attempt to reduce the collisions cardinality. These functionalities make it very difficult to predict the elapsed time (T_{slot}) needed to accommodate all transmissions in a single hop, even when the number of nodes on each hop is known. Collisions can be responsible for network partitions. In such cases, when the initiator node and the target records are in distinct partitions, the query will fail. For random topologies, partitions may occur frequently, which may reduce the number of significant cases analysed.

Concurrency Relaying node retransmissions are locally and independently decided by each node, which can cause concurrency since no hold time is considered.

Node Deployment and Density Modelling MANETs and WSNs with random-based topologies causes a further problem generated by possible connectivity faults. Our strategy to address this problem consists in gradually increasing the connectivity radius for the same number of nodes and area, until a connected graph is generated. However, the node deployment is not uniform, and zones with high node density (many nodes in the same range) are more favourable to collisions and subsequent packets drops, that may prevent the query propagation.

The realistic conjugation of this effects alerts to the fact that multi-hop wireless communication do not always correspond to well established expanding rings.

The *ns-2* logfiles trace all the relevant events from implemented protocols in time tagged extended reports, and are the source of data for evaluation analysis. A logfile inspection denotes the consequences from collisions, partitions, concurrency and node density. The *ns-2* text-based logfiles are excessively large for direct inspection. Scripts based on *bash*[3] were developed to extract the data for each evaluation metric.

The ABC evaluation does not focus only on implementation issues. The main purpose is to compare ABC with a naive implementation using plain flooding, and with BCIR variations which are the most efficient algorithms to control the query searching scope.

The next section is devoted to network topologies, a component which has a significant impact on the performance of multi-hop algorithms. The following evaluation uses multiple network topologies class models, including extreme deployments occurring in Mines.

5.4.1 Network Topologies Classes

The OSI[4] model describes a conceptual model for communication, abstracting the supporting technologies [293]. We focus our attention on the first two media layers that are independent from any particular sensor network technology. The physical layer is a fundamental layer specifying how bits are transmitted and the communication primitives are very close to the respective hardware and transmission technologies. The data link layer describes how data frames can be transmitted between nodes across the physical layer. The *ns-2* simulation tool contains several modules that are considered valuable and realistic models for both layers.

The *ns-2* geographical node disposition is handle by a linear transformation mapping topological graphs generated with the NetworkX[5] and positioning the network nodes in the *ns-2* two-dimensional coordinate system. All topologies have one connected component, *i.e.* for any chosen node in the network it is possible to reach any other node, using a multi-hop communication primitive. The physical settings used for evaluation purposes are described in Section 5.4.2.

To illustrate the evaluated network topologies, some images were captured from Network AniMator (NAM)[6], as depicted in Figures 5.5 – 5.7. NAM is a graphical animation tool for viewing network simulation traces and real world radio communication packet traces. At the end of a simulation, the *ns-2* outputs

---

[3] `https://www.gnu.org/software/bash/`

[4] Open Systems Interconnection

[5] `http://networkx.github.io`

[6] `http://www.isi.edu/nsnam/nam/index.html`

a text-based (**.tr*) logfile and an animation file (**.nam*). The animations allow users to visualize the progression of the algorithm. It displays the expanding ring in progress as a consequence of broadcast retransmissions in a wireless multi-hop network. The animations include main communication events (*e.g.,* node transmissions) and signal propagation over time. However, without the required detail for an accurate evaluation. Instead, the logfiles (**.tr*) have the crucial communication minutia, analogous to a real data fingerprint trace. Analysing the *ns-2* logfiles containing all the reported events is a hard process, which was automated by scripts in *BASH*, and relevant data were extracted for a file in a format that facilitates the evaluation metrics processing.

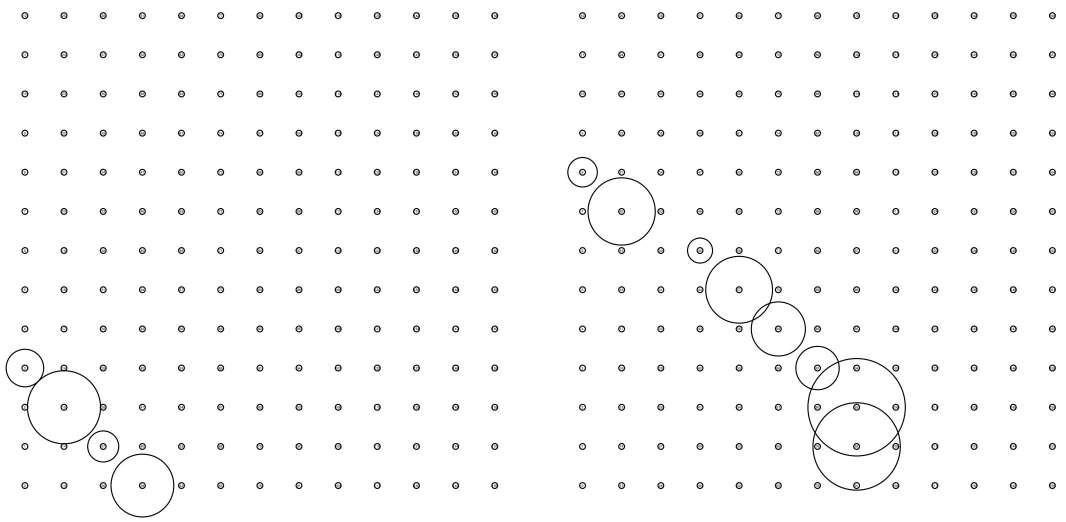
Regardless of the *ns-2* communications primitives, the evaluation inspects three distinct two-dimensional synthetic topology class models: *i*) square Lattice, *ii*) Manhattan, and *iii*) Mine. Considering these high level topologies, the evaluation emphasises on testing how the algorithms performance is affected in typically network topologies deployments. These topologies were chosen to mimic real world situations since they are representative models of what can be found across relevant literature (Section 5.2).

5.4.1.1 Lattice

The most typical topology shapes for base-station structured wireless networks are: *i*) hexagon Lattice and *ii*) square Lattice. The hexagon Lattice topology is one of the most used topologies for antenna positioning, minimizing overlapped radio coverage area [61]. It is typically used in wireless networks that rely on base-stations for communication relaying. It is not suitable for multi-hop unstructured networks, where it is necessary to maintain partial connectivity between undistinguishable nodes.

Nevertheless, the square Lattice is one of the most used network topologies to cover an open field geographic area, and for this reason it is considered for evaluation. In square Lattice, nodes are placed equidistant in rows and columns in a grid, thus corresponding to a regular pattern topology. The interior nodes (except the nodes at the frontier) have the same number of nearest neighbours and the same distance between all pairs of nearest neighbours.

Figure 5.5 shows two images captured from NAM. It depicts the square Lattice topology, generated with $13 \times 13 = 169$ nodes uniformly distributed along



(a) ABC query propagation at $\mathsf{t} = 4\,\mathrm{ms}$ (b) ABC query propagation at $\mathsf{t} = 122\,\mathrm{ms}$

Figure 5.5: Propagation snapshots of a Lattice topology, with ($\mathsf{N} = 169$).

the rows and the columns. The topology is covering an area of $3200 \times 3200\,\mathrm{m}^2$ resulting in a node density of $\simeq 16.5$ Nodes/Km^2. The distance between nodes is $200\,\mathrm{m}$. The diagonal distance is $\sqrt{2} \times 200 \simeq 283\,\mathrm{m}$ and the radio range is $250\,\mathrm{m}$. Therefore, no diagonal adjacent node communication is possible. When an interior node transmits, all four neighbours hear a busy signal and they cannot retransmit. The neighbours have to wait some backoff time, thus delaying the broadcast process, until one of the next-hop (four nearest neighbours) nodes will rebroadcast the received message.

The snapshots in Figure 5.5 include circles representing the radio propagation signal. Both figures were captured from a single ABC query, launched by the bottom left node. The NAM animation helps to identify the most recently active nodes and provides an instant view of the spreading radio propagation signal.

Figure 5.5a depicts the network state after $4\,\mathrm{ms}$ from the beginning of the query. As expected, the theoretical ring expansion in the Lattice topology results in the retransmitting nodes being aligned in a diagonal form. However, in Figure 5.5b after $122\,\mathrm{ms}$ elapsed the diagonal is vanishing, due to nodes being unable to retransmit at the same time in a shared medium.

5.4.1.2 Manhattan

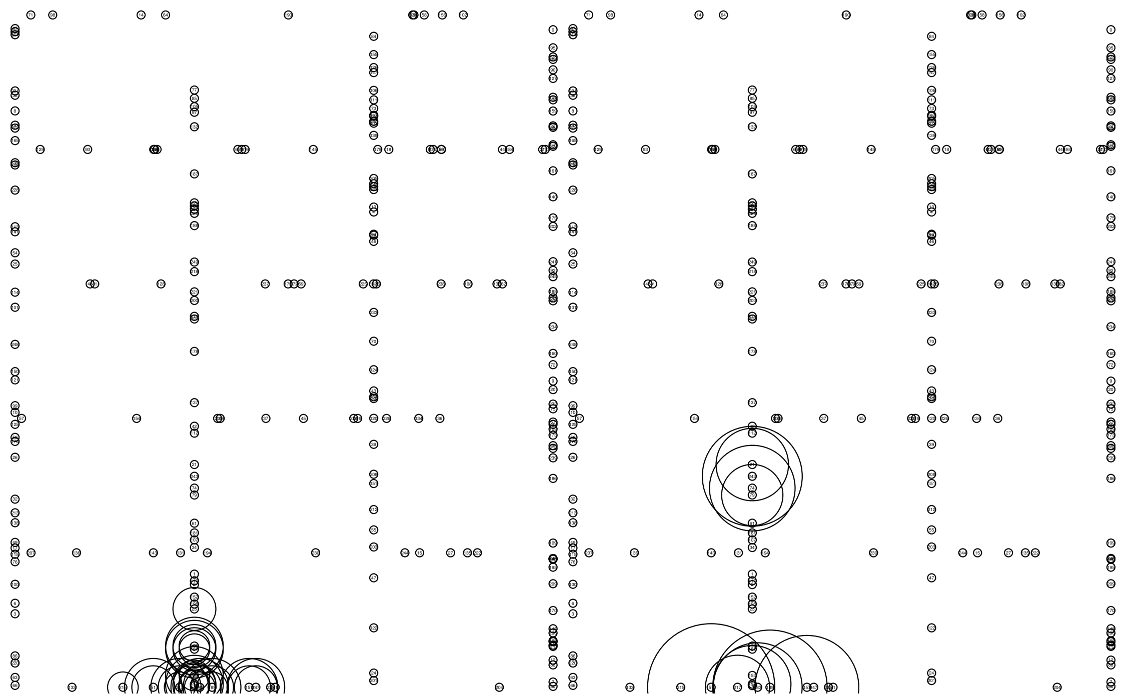
The Manhattan model randomly distributes mobile wireless devices on urban areas. Manhattan topologies are used when a pervasive communication infrastructure is deployed through the city streets, mainly imposed by the physical infrastructures (streets, buildings, etc) and represented by the respective maps.

The Manhattan topology depicted in Figure 5.6 was generated using the BonnMotion[7] tool with 250 nodes, distributed along the 6 horizontal and 4 vertical streets, covering an area of $2\,000 \times 2\,500\,\mathrm{m}^2$, resulting in a node density of $50\,\mathrm{Nodes/Km}^2$.

The Manhattan scenarios contemplate streets equidistant from each other, ensuring that the interior streets share equitably distance to cover the target area. The Manhattan topologies have two main characteristics that differentiate them from the Lattice regular pattern topology: *i*) the distance between the streets is greater than the radio range, to avoid direct communication from one

---

[7] `http://sys.cs.uos.de/bonnmotion/index.shtml`



(a) ABC query propagation at $\mathsf{t} = 40\,\mathrm{ms}$ (b) ABC query propagation at $\mathsf{t} = 58\,\mathrm{ms}$

Figure 5.6: Propagation snapshots for a Manhattan topology, with ($\mathsf{N} = 250$).

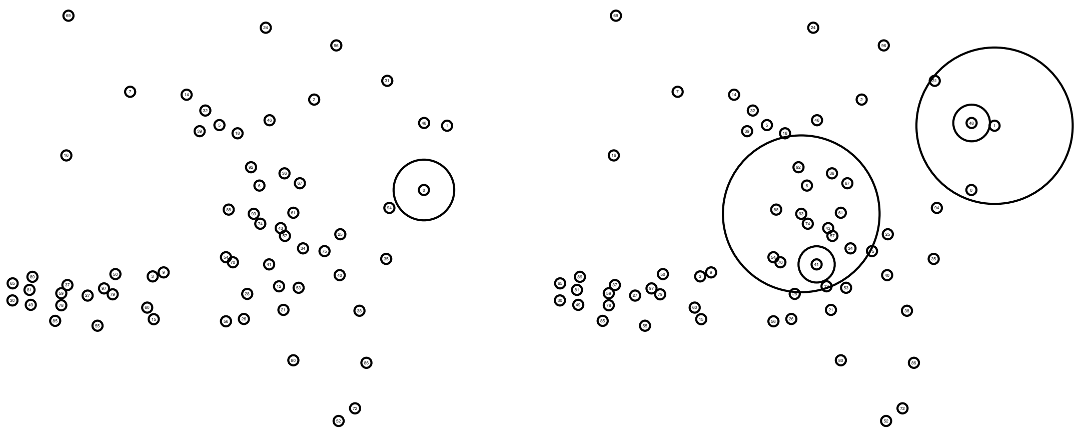
street to another and *ii*) nodes are randomly placed along the streets, resulting in different distance between a none and its neighbours. Thus, to ensure the connectivity of the network, the node density was increased significantly when compared with the square Lattice.

Both screen images depicted in Figure 5.6 show an ABC query expanding through the city streets. Although the initiator is randomly chosen, a quick inspection clearly identifies which node launched the search query.

5.4.1.3 Mine

The creation of the Mine topologies mirrors previous works using two-dimensional Random Geometric Graphs (RGGs) to model wireless networks topologies [105, 133]. The $\mathsf{RGG}(\mathsf{N}, \mathsf{r})$ model is a random geometric graph with N nodes that are randomly placed inside an unit square area and any two nodes whose euclidean distance is below r are linked by an edge.

Figure 5.7 depicts one of the generated RGG topologies with $\mathsf{N} = \mathsf{64}$ nodes, with two snapshots captured from NAM. On the first snapshot ($\mathsf{t} = \mathsf{2}\,\mathrm{ms}$), the animation depicts the initiator node transmission, starting a new query. The second snapshot ($\mathsf{t} = \mathsf{54}\,\mathrm{ms}$) shows how the ABC query expands in the searching



ABC query propagation at $\mathsf{t} = \mathsf{2}\,\mathrm{ms}$ ABC query propagation at $\mathsf{t} = \mathsf{54}\,\mathrm{ms}$

Figure 5.7: Propagation snapshots on a Random Geometric topology, with $\mathsf{N} = \mathsf{64}$, and the number of nodes were reduced for better legibility purposes only.

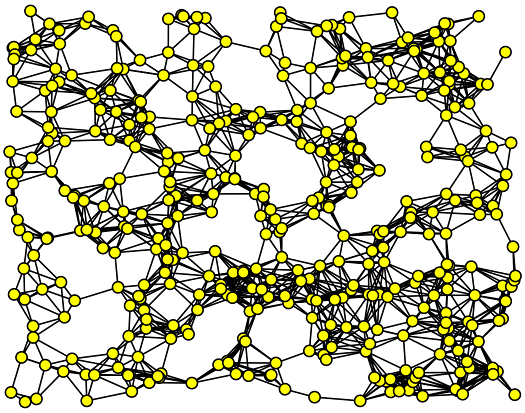


Figure 5.8: Random geometric graph $(\mathsf{N} = 400,\ \mathsf{r} = 0.095)$.

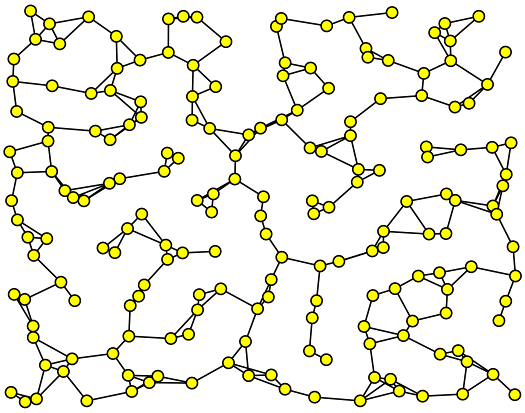


Figure 5.9: *Modified* random geometric graph $(\mathsf{N} = 189,\ \mathsf{r} = 0.095)$.

phase, with nodes relaying the broadcast message. The second snapshot confirms our predictions made in Section 2.5.3, that the network connectivity in a RGG topology can be accomplished by increasing the radio range, even through it will affect significantly the number of nodes contacted by a single retransmission. This can result in dense networks (Figure 2.11 in p. 57) generating additional congestion problems arising from overlapped transmissions, as observed when increasing the geographic node density.

Figure 5.8 depicts an original RGG as created by NetworkX and defined by $\mathsf{RGG}(400, 0.095)$. Unfortunately such topologies are unsuitable for representing Mine topology scenarios because they reveal an excessive number of clusters that will regenerate an unneeded amount of redundancy.

To represent more realistic Mine scenarios, two operations were performed: *i*) remove node neighbours that are too close, *i.e.* whose Euclidean distance is less than 2 meters and *ii*) a maximum graph degree was defined by removing nodes that exceed a predefined threshold. These operations satisfy a practical principle: sensor deployment does not consider extra sensors in areas that are already sufficiently covered. The final result is a *modified* random geometric graph that is deemed to better represent Mine tunnel structures, resulting in a topology with less redundant paths and lower node density.

Figure 5.9 depicts the *modified* RGG, considering a node degree limit given by $\mathsf{deg}(\mathsf{v}) < 5$, *i.e.* the maximum degree is $\Delta(\mathsf{RGG}) = 4$.

The number of removed nodes depends on the initial RGG topology. Thus, in the *modified* RGG the number of nodes is only known subsequently, after stabilizing the final *modified* topology. There is a strong correlation between both topologies represented in Figures 5.8 – 5.9, illustrating the proposed transformation for the generation of Mine scenarios.

Initially, each topology corresponds to a RGG with 400 nodes. Upon applying the Mine transformation previously described the final number of nodes is unknown. A pool of 500 samples of Mine topology scenarios were generated and the number of nodes distribution is characterized by $\overline{\mathsf{N}} \simeq 164.842$ ($\sigma = 4.55$). The number of nodes 142, 152, and 165 represent the 25th, 50th, and 75th percentiles, respectively. The observed variability is justified by the randomization effect and falls within an acceptable dispersion. The nodes are accommodated in a square area of $2500 \times 2500\,\mathrm{m}^2$, thus considering the average values, and results in a node density of 26.37 ($\sigma = 0.73$) nodes/Km^2.

5.4.2 Experiment Setup

Comparing the performance of distinct algorithms requires a detailed description of experiment conditions. Each scenario changes according to the parameter under evaluation. For fairness, all the other parametrizations are ensured to be the same, enabling the repetitions of experiments for other algorithms in a different occasion and in the same circumstances. One critical simulation environment component is the network topology. The information concerning the network topology and simulation settings are saved on external files, thus enabling to evaluate distinct algorithms in the same simulation scenarios. The developed scripts are generic and may be parametrized to meet to users requirements. The general evaluation parametrization is taken as follows:

- Initially, a set of 500 graphs were generated for each topology model, creating the pool of network topologies. Those files are transformed to store node positions in the *ns-2* format, becoming ready to be integrated by *ns-2* scripts. Thus, all the generated synthetic topologies (Lattice, Manhattan and Mine) are reproduced in *ns-2*. Each simulation run corresponds to one topology scenario. Considering the radio transmission range, node positions are estimated in order to keep an expected connectivity similar to the synthetic model.

- Network traffic is dynamically generated at run time. For each topology,

> 200 queries are scheduled using an uniform distribution over the maximum simulation time. Each query packet has a size of 1 000 Bytes. Both the query initiator node and the target record are randomly chosen. Every query is uniquely identified with a `queryID` tag. Previous settings plus selecting the algorithm to use during the searching process (FLOOD, BCIR, BCIR* or ABC) are included in the traffic scenario file and stored in the *ns-2* format.

The precision of the estimation increases with the sample size. The number of experimented queries for a given topology class is composed of $500 \times 200 = 100\,000$ samples, ensuring a good significance to the quantitative research methodology. Running those simulations returned a considerable amount of data, in a total of 208 GB for almost 4 000 logfiles with pure text traces from *ns-2*.

A summary of the radio communications physical settings is presented in Table 5.1, specifying the main physical layer parameters and the *ns-2* simulation environment definition. Antenna characteristics, signal power and receiving threshold are established to proportionate approximately 250 m for maximum radio range. The queries are scheduled in non concurrent random order and they search for random records.

Table 5.1: Simulation parameters for *ns-2*.

Attribute	**Default Value**
DELAY_MIN	10 ms
Jitter	7 ms
Packet size	1 000 Bytes
Max simulation time	17 000 s
Number of topologies	500
Radio range	250 m
Network links	11 Mbps
Transmission power	25 dBm
Antenna gain	1
Antenna polarization	Omnidirectional
ns-2 propagation model	TwoRayGround

The *ns-2* is configured to use the **NO Ad-Hoc routing agent (NOAH)**[8] to ensure that results are not affected by any default built-in routing protocol.

For ABC evaluation, the performed simulations use a LBF with $\mathsf{m} = 256$ cells and $\mathsf{k} = 4$ $\mathsf{hash}(\mathsf{x})$ functions. The magnitude for previous values are suggested by Eq.(5.1) extracted from [100]. The expected false positive probability is up to $\mathsf{fp} = 5\%$ assuming that LBF can keep information for at least 40 records, *i.e.* $\#\mathsf{LBF} < 40$.

$$\mathsf{m} = -\frac{\#\mathsf{LBF} \cdot \ln(\mathsf{fp})}{(\ln 2)^2} \quad ; \quad \mathsf{k} = \ln(2) \cdot \frac{\mathsf{m}}{\#\mathsf{LBF}} \tag{5.1}$$

The ABC attenuation factor *att* (used in Algorithm 6) may be any value within $[0, 1]$ and we decided to choose $\mathsf{att} = 0.9$, thus corresponding to 10% of attenuation for each hop.

The FLOOD, BCIR and BCIR* are uninformed mechanisms and consecutive queries are independent from each other, therefore, no warm-up time is necessary before starting to collect results. Unlike those mechanisms, ABC depends on historical data, and the LBFs start empty. To warm-up the ABC algorithm, the first 10 executed queries are discarded, their only purpose to fill the LBFs with some context and put ABC into normal execution conditions.

5.4.3 Result Analysis

The distance (hop) between the query initiator and the first node responding with a successful answer is one of the major key parameters in multi-hop networks. Thus, the main results are presented as a function of the distance. The ABC is a distributed adaptive algorithm, where the retransmission decision occurs autonomously on each node, triggered upon message reception and typically dependent on the estimated distance. For a fairness evaluation, FLOOD, BCIR, BCIR* and ABC algorithms, were submitted to the same random query traffic conditions for performance comparison.

The performance evaluation takes into account the impact of the adaptive delay using two main **metrics**: *i*) **Latency**, the average end-to-end elapsed time that goes on from the moment a query is broadcast until its sender receives a reply (as described in p. 26), and *ii*) **# Retransmissions**, which counts all the

---

[8] `http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/`

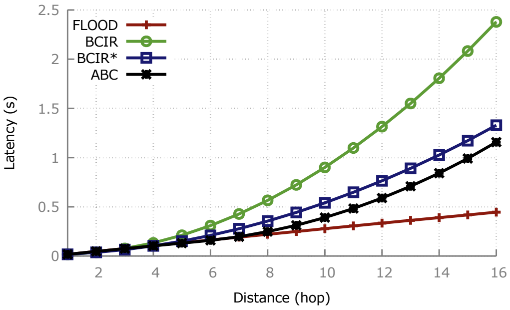


Figure 5.10: Estimated latency for Lattice scenarios.

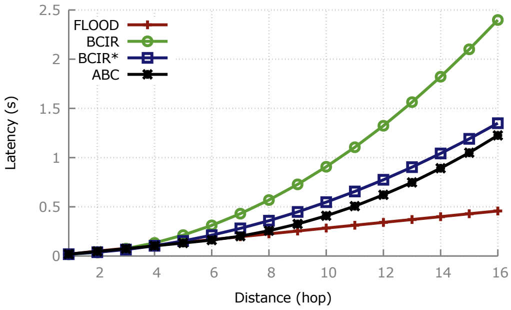


Figure 5.11: Estimated latency for Manhattan scenarios.

performed retransmissions until the query process ends. Assuming that any two retransmissions consume the same amount of energy, the number of retransmissions is an indicator for protocol energy efficiency (not considering physical issues that are hardware dependable).

Latency The latency time is computed on every search by measuring the elapsed interval between two events: *i*) the initiator node starts query search, and *ii*) the initiator node receives a positive answer. Unreceived answers are treated as failed queries.

The ABC algorithm significantly reduces the average latency time. The latency gain increases with the distance to the record, as shown in Figures 5.10 – 5.12. As an example, the difference between BCIR* and the FLOOD baseline for $\mathsf{H} = 4$ is similar to the difference between ABC and the FLOOD baseline for $\mathsf{H} = 8$. The ABC algorithm allows the user to find records that are distant from the initiator in a much faster way than BCIR* (which is one of the fastest mechanisms). The improvements on the reduction of latency reveal the contribution given by LBF on the acceleration of query propagation in the record direction, a unique property of ABC, which makes it competitive with FLOOD (the fastest algorithm with no added delay), when records are located up to the 7th hop.

All the topology models reveal similar estimated latency performance. As expected, the latency average values are almost identical for Lattice, Manhattan and Mine scenarios. Figures 5.10 – 5.12 are almost coincident, without significant differences between those topologies. However, a small deviation is perceptible in Figure 5.11 (Manhattan scenarios), due to the higher node density of the Manhattan scenarios, which increases the congestion problems.

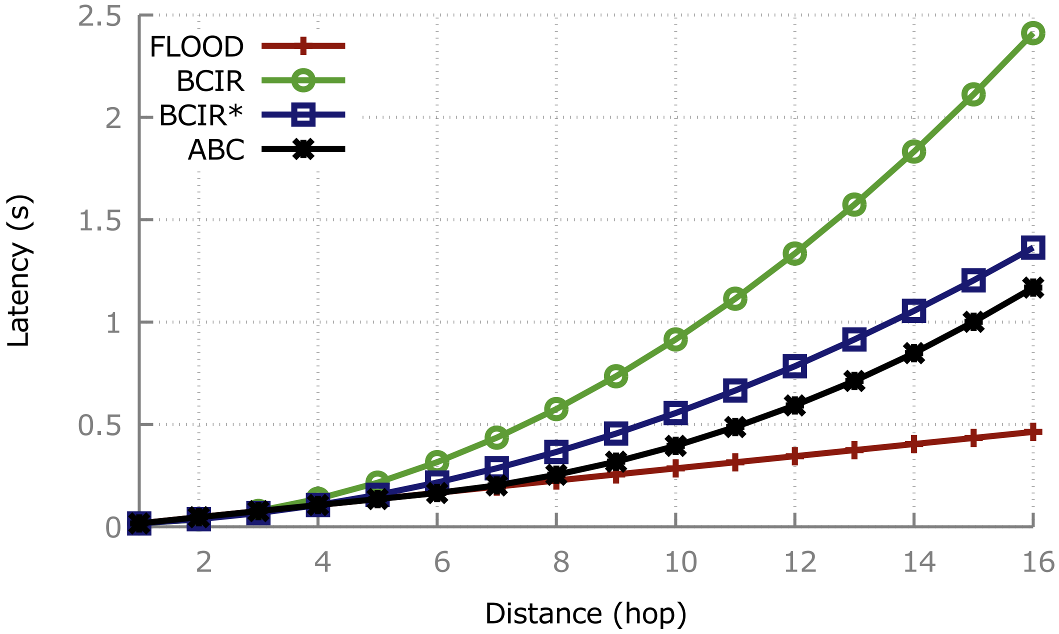


Figure 5.12: Estimated latency for Mine scenarios.

One of most important results is the improvement in latency achieved by the ABC algorithm. As depicted in Figure 5.12, ABC significantly reduces the latency response when compared with BCIR and BCIR*, given its ability to adapt the added delay to an expected record location intuition.

Figures 5.10 – 5.12 show that latency is mainly set by the chosen search algorithm and its relationship with the record distance (hop). Thus, the delay time component introduced by the cancellation algorithms far exceeds the effect caused by distinct topological class models.

Retransmissions Assuming that the number of retransmissions is the dominant component to estimate the algorithm energy efficiency for searching records on multi-hop unstructured wireless networks, the simulations results compare the energy efficiency in FLOOD, BCIR, BCIR* and ABC by counting the number of retransmissions performed by each one until all messages related with the same query cease to propagate.

The network topologies have some influence in the algorithms energy efficiency, as may be observed by the differences between the absolute values depicted in Figures 5.13 – 5.15.

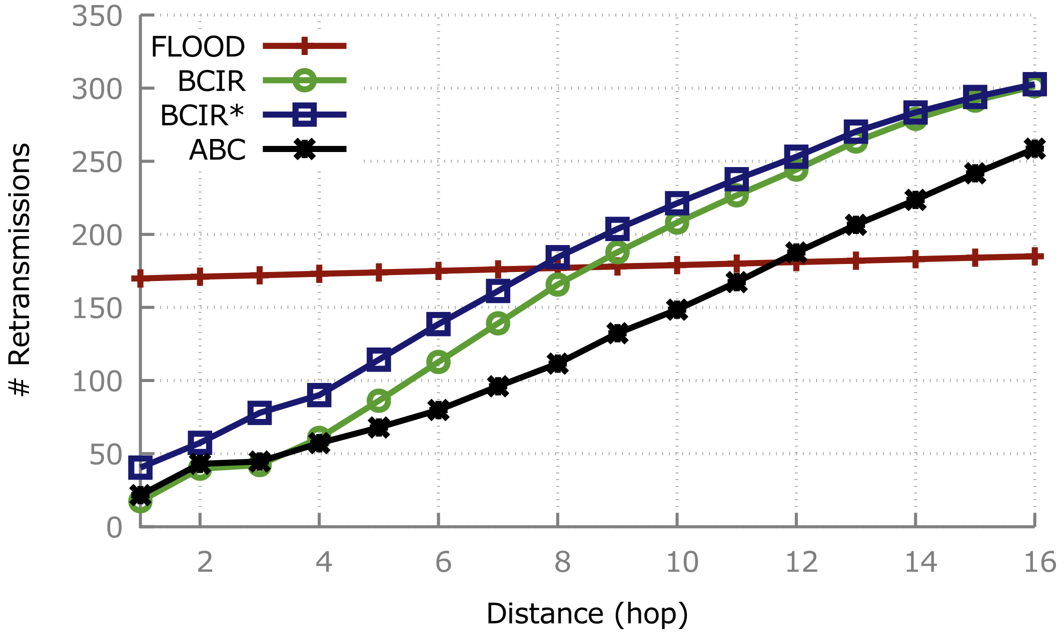


Figure 5.13: Estimated energy efficiency for Lattice topology.

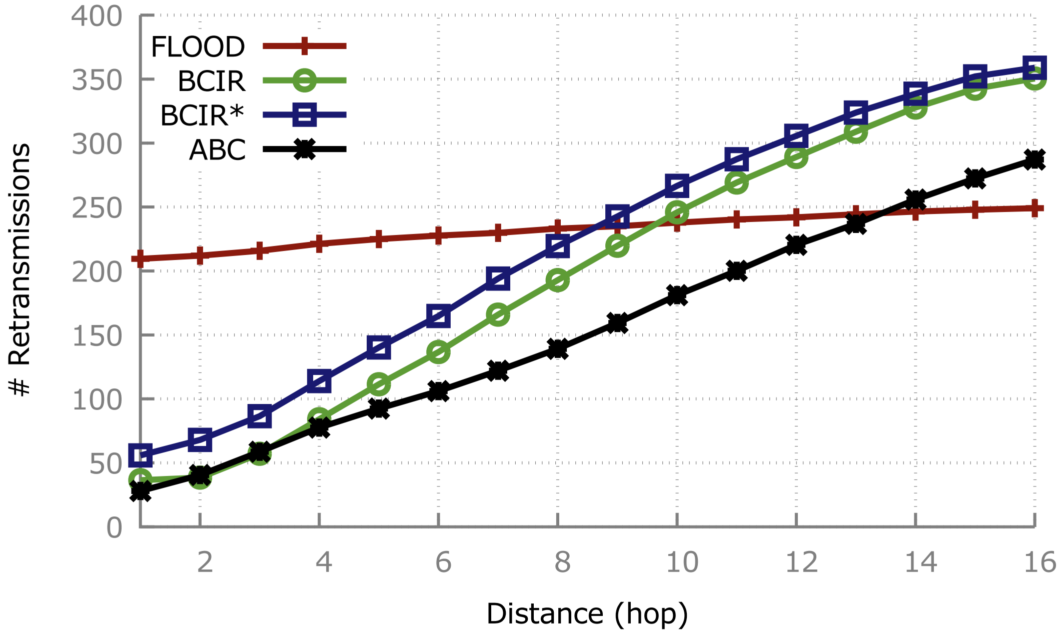


Figure 5.14: Estimated energy efficiency for Manhattan topology.

Although, the relative behaviour among the algorithms (for each topology) is similar.

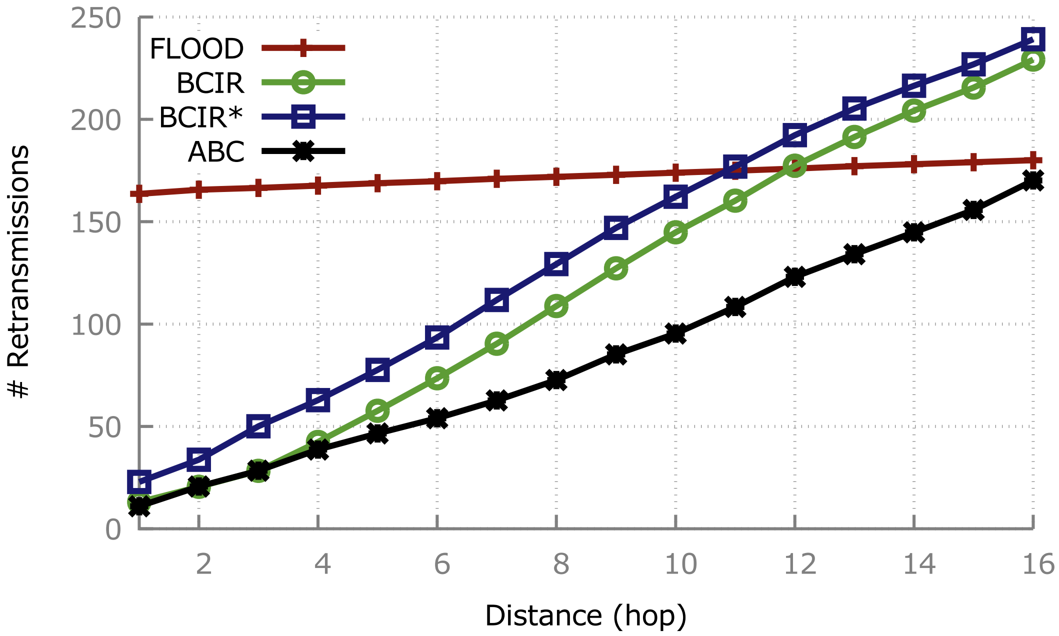


Figure 5.15: Estimated energy efficiency for Mine topology.

The most important result is that ABC significantly reduces the number of retransmissions when compared to other algorithms and on every studied topology, as shown in Figures 5.13 – 5.15. When records may be found in the first hops, the ABC algorithm improves the energy efficiency in a pattern comparable to BCIR (that was the most energy-efficient cancellation protocol).

Results confirm the theoretical predictions given by Eq.(3.19), where the **threshold** (H_{th}) is a hop limit until cancellation mechanisms are useful to apply, *i.e.* consuming less energy than a simple flooding.

For example, considering one of the Lattice scenarios where the diameter is $\mathsf{H}_{\mathsf{max}} = 24$, and therefore the predicted threshold is $\mathsf{H}_{\mathsf{th}} \leq \frac{24}{\sqrt{2}} \leq 17$. In Figure 5.13, ABC crosses the FLOOD baseline for $\mathsf{H} = 12$, which confirms the predicted threshold limit. Beyond the distance limit ($\mathsf{H} = 12$) it is better to disable the cancellation mechanism and let the flooding end by itself. However, it should be remembered that the network diameter may not be known or may not be properly estimated in runtime, thus interfering with the algorithm efficiency.

The difference in performance between Lattice and Manhattan topologies, depicted in Figures 5.13 – 5.14, may be explained by the influence of node distance and congestion.

Table 5.2: Packet drops per algorithm and topology node density.

Searching	**Lattice**	**Manhattan**	**Mine**
FLOOD	$\simeq 19.332\%$	$\simeq 29.790\%$	$\simeq 5.963\%$
BCIR	$\simeq 19.064\%$	$\simeq 29.011\%$	$\simeq 5.644\%$
BCIR*	$\simeq 19.383\%$	$\simeq 29.954\%$	$\simeq 5.807\%$
ABC	$\simeq 13.433\%$	$\simeq 26.980\%$	$\simeq 3.712\%$
Density (nodes/Km^2)	$\simeq 16.50$	50	$\simeq 26.37$ ($\sigma = 0.73$)

Cancellation algorithms (BERS, BERS*, BCIR and BCIR*) evidenced a trade-off between latency and the number of retransmissions, as discussed in Chapter 3. However, a cross-analysis of Figures 5.12 and 5.15 suggests that ABC is able to reduce latency and increase energy efficiency. Comparing their performance with ABC, in cases where records are found at the same distance, ABC offers significant gains in time with faster responses, and with a reduction in the number of retransmissions. The gains increase in line with the distance between the source of the query and the target record location.

5.4.4 Reliability

The reliability inspects the simulation data results for three additional indicators: *i*) proportion of packet drops, *ii*) records found per hop, and *iii*) proportion of unsuccessful queries. The reliability verifies the consistence of the evaluated metrics in the previous section. These additional indicators complement the evaluation process and characterise the randomization introduced in the experiments. Errors influence the performed measurements when using randomization, motivating a confirmation that repeating the experiments in similar conditions will produce the same results.

Table 5.2 summarizes the following metrics for each topology and searching mechanism: *i*) packet drop ratio and *ii*) node density. It should be notice that, density is constant for Lattice and Manhattan scenarios, whereas in the Mine scenario (topologies with variable number of nodes) node density is estimated using the average number of nodes, and given by $\left(\frac{\overline{\mathsf{N}}}{\mathsf{Area}}\right)$. From Table 5.2, regardless of topology network, packet drops for ABC is perfectly distinguishable from the other algorithms. Nevertheless, with respect to BCIR and BCIR* they are very similar to FLOOD. Not surprisingly, the protocol performance is affected by the network topology (notably by the node density).

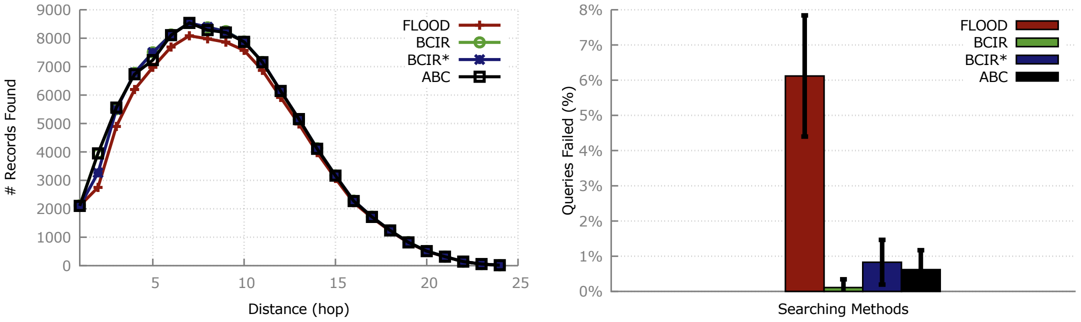
In practice, an approach to compensate frequent losses of connectivity at specific locations, is to increment the pool of nodes covering the target area. The node density is given by $\frac{\#\mathsf{N}}{\mathsf{Area}}$. In Lattice topologies ($\mathsf{N} = 169$) nodes are uniformly distributed along the columns and rows. In the case of Manhattan scenarios and to maintain network connectivity, the number of nodes were increased. In the Manhattan scenarios $\mathsf{N} = 250$ nodes are randomly distributed (in contrast to Lattice) along the streets and taking some precautions to not overdo coverage redundancy. In Mine topology, N is only known at runtime, since it results from the generation of a modified connected random geometric graph, resulting in a variable number of nodes.

Whenever a new topology instance is generated, records are randomly allocated to the network nodes, without any predefined location for each one. To verify the effective record distribution of resources, the *ns-2* logfiles were inspected to check if there was any bias in record location and count the number of unanswered queries. For that, the **Queries Failed (QF)** percentage were computed, using the formula $\mathbf{QF} = \frac{\mathsf{iq}-\mathsf{aq}}{\mathsf{iq}} \times 100\%$, where iq is the number of initiated queries, and aq the number of positive answers received by the initiator.

The existence of packet drops (Table 5.2) does not mean that the queries will fail in the same proportions. The broadcast will explore alternative paths, enabling records to be discovery under such adverse conditions. To compare the number of records found per hop with the percentage of unanswered queries for each topology, 200 queries were held, distributed over a pool of 500 random generated topologies.

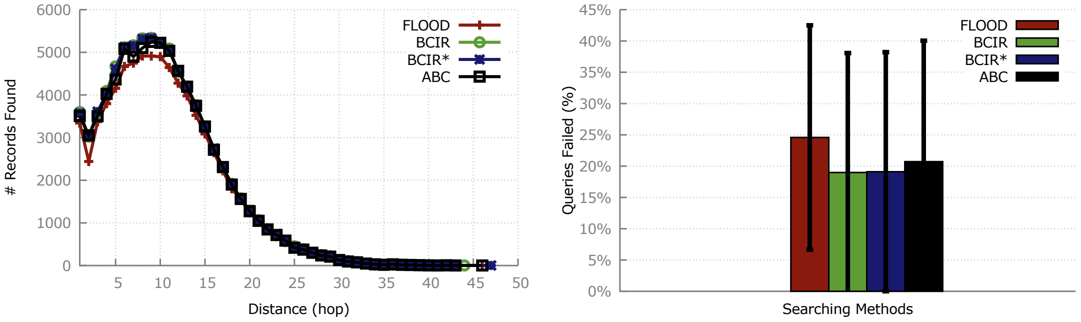
Figures 5.16 – 5.18 depict the number of records successfully found and the proportion of queries failed for each network. Figures displays record distribution as a function of distance (hop), allowing to make an explicit relationship with the distance to the initiator, which is one of the most significant multi-hop network topology parameters. Results are presented side by side suggesting a correlation between the number of records found and the percentage of queries failed.

Figures 5.16a – 5.18a depict the distribution of target records per hop, giving a confirmation that records are uniformly distributed. In a faultless scenario, these curves should be completely coincident. However, the system is not deterministic



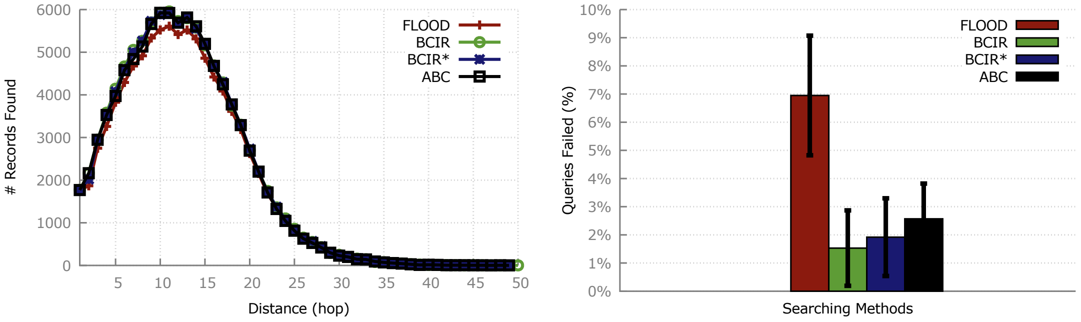
(a) Record distance distribution (b) Proportion of unanswered queries

Figure 5.16: Comparing the number of target records found per hop with the percentage of unanswered queries for Lattice scenarios.



(a) Record distance distribution (b) Proportion of unanswered queries

Figure 5.17: Comparing the number of target records found per hop with the percentage of unanswered queries for Manhattan scenarios.



(a) Record distance distribution (b) Proportion of unanswered queries

Figure 5.18: Comparing the number of target records found per hop with the percentage of unanswered queries for Mine scenarios.

and failures occur, resulting in a slight variation between the different algorithms. The differences are more evident in the case of FLOOD, since it has the lowest latency from all the evaluated algorithms, thus increasing the likelihood of failures.

Considering the Lattice scenarios, when cross-comparing both graphs from Figure 5.16, its leads to suspect that the reason for less records found by flooding (Figure 5.16a) is related to the fact that flooding has the higher percentage of failed queries (Figure 5.16b) relatively to the other algorithms.

The proportion of failed queries is depicted in Figures 5.16b – 5.18b. These figures present the univariate statistical analysis, using its central tendency (average) and dispersion (standard deviation). The limits expressed in the error bars for low density scenarios (Lattice and Mine) result mainly from the randomization effect. In Manhattan topologies that were generated with higher node density, the proportion of failed queries is similar for all algorithms, since higher wireless node density does not benefit radio communications.

BCIR is the algorithm with less failed queries, due the fact that it has the highest latency performance. BCIR added delay creates better conditions (less congestion) to retransmit the answer back to the initiator.

ABC achieves a discovery performance similar to BCIR and BCIR*, with a significant reduction on latency and energy consumption, justifying the effort to implement an adaptive delay approach.

5.5 Discussion

The ABC is a new algorithm for record searching over unstructured networks, that takes into account the wireless sensor nodes scarcity of resources.

Experiences conducted in *ns-2* simulator show that cancellation strategy is feasible and it achieves successful results on search algorithms that are able to stop an ongoing broadcast, considering several realistic simulation scenarios.

Assuming that records are available upon a few hops, which often happens in practice (unless resources are very scarce and far from the initiator), the cancellation approach significantly reduces the number of retransmissions (waste of energy) diverging from starveling flooding approaches.

The evaluation of ABC performed in Section 5.4 showed that a prediction on the location of target records may significantly reduce energy costs for successive discovery queries, while still benefiting from a reduction in latency (end-to-end time). In BCIR and other authors approaches [11, 203], the effort to reduce latency (*e.g.,* by progress from BERS to BERS*) presumes an unwanted penalty in the number of retransmissions. The ABC algorithm overcomes this adversity whenever there is a record location intuition. The evaluation shows that, among broadcast cancellation mechanisms, ABC has the best performance of all the compared algorithms.

ABC stands out from its competitors due to its capability to reduce both latency and the retransmission ratio. ABC abandons the uninformed approach of BERS, BERS*, BCIR and BCIR* to include an informed heuristic, using data collected from recent executed queries.

A strategy to reduce latency is speeding up the searching query. However, an omnidirectional propagation (unknown record location) also retransmits the query in the opposite direction of the target record. Upon the successful discovery, the wrongly explored locations have to retransmit cancellation messages to stop the expansion of the search query. Thus, the retransmission ratio can raise quickly, evidencing a energy-time trade-off that seemed to be unshakable. However, the ABC algorithm may propagate faster in the expected *"right"* direction (gradient-based) without slithering too much for *"wrong"* directions, considerably attenuating what seemed to be an unavoidable trade-off.

The ABC general approach adequately copes with scenarios in which there are no clues about the location of target records. All scenarios used for protocol evaluation (in *ns-2*), distribute records randomly along the network nodes, to ensure that ABC algorithm initiates the searching mechanisms under the very same conditions.

External conditions (*e.g.,* moving objects, power loss) may be responsible for wireless communication failures, that may be attenuated by designing a sensor network with proper redundancy. Scenarios with lower node density show fewer redundancy and are more susceptible to failures. However, excessive density scenarios may also cause communication failures, due to the presence of a large number of collisions events.

The evaluation confirms a correlation between dense scenarios and higher packet drop occurrences, thus indicating that the network node density has a critical role in the number of packet drops.

Average gains achieved by ABC are very consistent (regular) in different network topologies. The performance of ABC is improved by its gradient mechanism (managed with LBFs) representing the intuition for speeding the query propagation towards the predicted record location.

ABC achieves its goals with a small penalty in the message size, resulting from piggybacking LBFs inside exchanged messages ($\mathsf{m_s}$ and $\mathsf{m_c}$). The ABC algorithm operates by taking advantage of the query propagation mechanism to epidemically disseminate the expected direction of the target record location, without increasing the number of retransmitted messages.

The ABC algorithm shows the success of an alternative approach that will allow for a reduction of time (latency) and energy (retransmissions), although it builds on the assumption of some topology stability and the existence of previous concluded searches (for better predictions).

Chapter 6

Conclusions

Recent technological advances paved the way to small size, low cost and battery powered microcontrollers with integrated sensors (*e.g.,* temperature, humidity, light) and wireless communications. Related work contextualization demonstrated that sensor nodes are evolving, becoming more powerful, autonomous, overcoming the limitations of static systems and being able to cope with real world environment dynamics, contributing to turn the *Internet of Things (IoT)* vision into a reality.

A wide variety of applications require that wireless sensor nodes communicate beyond their radio range. To achieve it, nodes are required to have the technological means to organize themselves into multi-hop unstructured networks. In this model, the major building block is the broadcast communication primitive, which is extensively used by data dissemination protocols. However, sensor nodes powered by limited energy sources pose new challenges to the need of keeping systems alive as long as possible. Difficulties rise when sensors are placed in locations with difficult accesses (*e.g.,* underground Mines), where a simple sensor battery replacement may be impossible or impractical. Knowing that radio transmissions are one of the most energy demanding operations, multi-hop communication algorithms must avoid energy consumption from unneeded retransmissions, as reviewed in the initial chapters. The unnecessary retransmissions footprint caused by simple flood-based broadcast algorithms gave us the motivation to look for flooding alternatives and to step up to the challenge of investigating broadcast controlled solutions.

This thesis proposes new approaches to attenuating the negative impact of flooding algorithms on node energy consumption. Searching mechanisms and location services for unstructured networks are examples of very active research areas exploring energy-efficient and time-efficient multi-hop broadcast algorithms. However, to the extent of our knowledge, no general solution for this problem addressing both energy and time constraints has been found. Centralized solutions are not acceptable for unstructured networks, due to the unreliability of communications that characterises both WSNs and MANETs, which invalidates any attempt to predict or maintain a full view of the system state.

Our main strategy to reduce energy consumption in multi-hop unstructured networks explores distributed mechanisms to limit broadcast propagation, generic designated as *"broadcast cancellation"*. This approach can stop broadcast propagation, contributing to improve the performance of several core services such as location services. Location is one of the most important services in a wide variety of contexts, for example in search algorithms. Searching for items in a dynamic network is a recurring task and a very demanding primitive for sensor-based applications. Our preliminary tests show in practice that, multi-hop broadcast using the cancellation approach provides good indications, outperforming some existing algorithms.

This thesis evaluates and compares some approaches for energy-efficient search mechanisms (*e.g.,* ERS, ERS-TTL, BERS and BERS*). Experience gained during the exploratory work, including the development of a simulation tool designated as Synchronous Round-based Simulator (SRS), described in Chapter 2, exposed the negative impact produced by flooding approaches. Using simulation tools and considering synthetic network topologies, we conclude that the current solutions have severe limitations, detailed in Chapter 3.

We verified that algorithms based on ERS can limit the expansion of flooding, which is a great improvement when compared with pure flooding search mechanisms. However, when the resource is not found within the limited search area, ERS uses multiple flooding attempts with a growing range. Moreover, tuning ERS parameters requires previous knowledge of the network size, which is a very demanding and resource-consuming task if the network topology has to be discovered in run time.

Algorithms based in chasing packets such as BERS and BERS* are very promising, but still need further improvements. In particular, investigating new algorithms which can cope with dynamic topologies (*e.g.,* WSNs, MANETs), independently of having some network topology knowledge (*e.g.,* diameter). Results showed that cancellation mechanism may improve flood-based implementations, and that on average, cancellation algorithms are more efficient than existing alternatives for multi-hop networks.

Thesis outcomes contain two efficient uninformed search mechanisms, *i.e.* BCIR and BCIR*, able to stop an ongoing broadcast using the cancellation strategy. In Chapter 3 we conclude that BCIR and BCIR* successfully overcome the performance characteristics reported by related work. However, the cancellation approach exhibits a trade-off between time and energy. Cancellation parameters may be conveniently adjusted for configuring the performance of the algorithm. BCIR showed an antagonistic relationship between latency and retransmission ratio, when tuning the parameters. Likewise, other approaches (*e.g.,* BERS) evidence this trade-off, also designated as energy-time, which is a peculiar effect of limited broadcast mechanisms.

A combination of informed searching and broadcast cancellation mechanisms posed new challenges, in particular the need to represent the expected location of network items in a memory-efficient manner, compatible with large networks, dynamic topologies and limited sensor memory, that characterises our network model. To address those issues, a new probabilistic data structure was investigated, designated as Linear Bloom Filter (LBF). The LBF contrasts with deterministic approaches that unambiguously represent a given network property (*e.g.,* location).

LBF evaluation performed in Chapter 4 concluded that calibration parameters $(\mathsf{m}, \mathsf{b}, \mathsf{k})$ may be adjusted to delimit the LBF expected error, and to determine boundaries on which LBF is a credible alternative to a deterministic data structure. LBF extends the traditional Bloom filters binary responses by including a confidence parameter that is capable of reflecting a fuzzy membership for each element. Realizing the requirements for WSNs and MANETs informed search mechanisms, the LBF probabilistic nature is designed to privilege local data, attenuating the distant ones which are usually less relevant.

A final contribution is the Adaptive Broadcast Cancellation (ABC) algorithm, presented in Chapter 5. ABC integrates a distributed learning mechanism that adapts the algorithm behaviour to previously conducted queries. It is an hybrid search mechanism that can initiate the search without any previous network location context. However, as messages are propagated through the network, it uses them to acquire an intuition (informed searching) on the most likely location of a given target record. To efficiently manage the collected data, ABC explores the probabilistic nature of the LBF data structure. ABC confirms that an adaptive approach based on an informed strategy may improve the performance of search mechanisms. Adaptive broadcast cancellation achieves very significant reductions in the retransmission ratio and latency, especially when records are available within a few hops, which often happens in practice unless resources are very scarce and thus far (on average) from the initiator.

The algorithms were studied under a theoretical approach (analytical expressions) complemented by evaluation experiments. Experiments conducted on the *ns-2* environment simulator confirmed that broadcast cancellation techniques such as BCIR, BCIR* and ABC significantly reduce the number of retransmissions required for searching target records on unstructured wireless networks.

This thesis further investigates the time needed for a cancellation mechanism to globally stop an ongoing broadcast upon target discovery, and the distance threshold that no longer justifies the cancellation effort, when the network diameter is known, considering the additional traffic it produces.

Finally, results show that, for random scenarios, it is possible to efficiently stop an ongoing broadcast using the cancellation strategy. This thesis balances the trade-off between the gains in the overhead (estimated energy cost) required to implement some broadcast cancellation mechanisms and its impact on latency. Adoption of an informed approach such as the one presented in ABC shows that it is possible to reduce both latency and retransmission rate in comparison with alternatives developed elsewhere, therefore contributing to minimize sensor battery depletion.

6.1 Future Work

Hardware and technological advancements should not be hampered by inefficient flood-based algorithms. There are several solutions for locating an item over a large scale area populated with wireless devices. However, there is always the possibility of improving the search mechanisms, due to their complex nature.

Despite the success revealed by broadcast cancellation approaches, trends for energy-time efficient search algorithms are an ongoing research topic.

Next, some lines for future research, giving continuation to the developed work and including new approaches to be explored.

Improving the performance of Linear Bloom Filter (LBF)

We claim that LBF is a robust, generic, probabilistic data structure that may be applied also on other application domains apart from this thesis focus. The idea of quantifying uncertainty in a set membership contrasts with traditional Bloom filters using a boolean representation.

One aspect for future development is to include an explicit deleting operation that may remove tuple instances from the LBF filter. The LBF should be able to manage record withdrawal whenever a given item is no longer available and search algorithms may abandon the periodical history clean up approach.

Given the lower influence observed by a reduction in the number of `hash(x)` functions (`k`), another future research strategy to improve LBF efficiency may explore the reduction of `k`, more precisely adopting a similar approach to a Cuckoo filter [80].

LBF overestimation may also be improved by evolving the actual linear quantization to a non-linear adaptive quantization approach, that may attend to scalability problems and fault events, which are typical of wireless networks applications. A future approach may explore the quantization levels allocation, assigning more precision to elements with greater confidence index penalizing the precision of the ones with lower confidence.

No data compression is considered in LBF. Adopting an encoding mechanism that reduces the number of bits required to represent the confidence value (`c`) will extend the number of quantization levels, thus improving the quality of filter responses. An acceptable lossy data compression approach may be explored, when the introduced error is inferior to the filter probabilistic component.

Exploring concurrent and clustering-based cancellation

Results show that cancellation is more effective when the searched records may be found in just a few hops away from the initiator. Generically, this occurs more often when record density increases. In high record density areas, the probability of finding multiple records before other nodes are notified by cancellation messages increases, thus multiple cancellation waves may propagate concurrently. A future improvement can emerge from including a mechanism that caches all the discovered records in the intermediate nodes, and uses such information to reduce the communication to a subpath between the initiator and the target node. The expected saving in retransmissions may be very significant since it is expected a lower cost in shorter distance queries. In addition, we expect to improve energy efficiency by deriving new strategies to implement the cancellation mechanism and explore the concurrent query propagations benefits.

Cancellation mechanisms may benefit from a clustering-based strategy, expecting to efficiently manage large sized wireless networks. To support network scalability, nodes may be grouped into disjoint and mostly non-overlapping clusters, reducing the number of retransmissions to cover a given target area. The formation of cluster heads may be opportunistically elected by direct broadcast messages within the same radio range, without increasing retransmissions costs.

In terms of further research, the next obvious step is to confirm how both clustering and concurrent cancellation can affect the energy-time trade-off in a multi-hop wireless network.

Improving cancellation using a gradient approach with extrema propagation

Inspired by the Adaptive Broadcast Cancellation (ABC) protocol achievements regarding the attenuation of the energy-time trade-off, we intend to continue the research for alternative heuristics that control the added delay. Expectations are that cancellation may adapt itself to node faults or expired record events. A suggestion for future development is to explore alternatives to the constant attenuation (`att`) used in the ABC exponential decaying mechanism.

Gradient information used to guide the expanding messages (searching and cancellation) may be improved by exploring other aggregation distributed algorithms, aiming at reducing query latency and overhead. Providing a summary of global system properties of which the expected record location is a good example, may be accomplished by using extrema propagation techniques [27].

Broadcast cancellation and extrema propagation may be combined (both rely on broadcast message exchange) to dynamically compute gradient properties, expecting to efficiently detect promising propagation directions.

Finally, it would be important to confirm the results on the generalised model in real world network deployments, covering other node compositions and network topologies.

Bibliography

[1] Ahmad, N., and Hussain, S. Z. Broadcast expenses controlling techniques in mobile ad-hoc networks: A survey. *Journal of King Saud University - Computer and Information Sciences* (2015). **Cited** on page 137.

[2] Akkaya, K., and Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks 3*, 3 (2005), 325–349. **Cited** on page 6.

[3] Akon, M., Shen, X., Naik, S., Singh, A., and Zhang, Q. An inexpensive unstructured platform for wireless mobile peer-to-peer networks. *Peer-to-Peer Networking and Applications 1*, 1 (2008), 75–90. **Cited** on page 28.

[4] Akyildiz, I. F., and Stuntebeck, E. P. Wireless underground sensor networks: Research challenges. *Ad Hoc Networks 4* (2006). **Cited** on page 3.

[5] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. A survey on sensor networks. *Communications Magazine 40*, 8 (Aug. 2002), 102–114. **Cited** on pages 4 and 25.

[6] Akyildiz, I. F., Wang, X., and Wang, W. Wireless mesh networks: a survey. *Computer Networks 47*, 4 (2005), 445–487. **Cited** on page 25.

[7] Al-Humoud, S. O., Mackenzie, L. M., and Vanderbauwhede, W. Dynamic counter-based broadcast in manets. In *Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks* (New York, NY, USA, 2009), PM2HW2N '09, ACM, pp. 84–88. **Cited** on pages 4 and 34.

[8] Al-Karaki, J., and Kamal, A. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications 11*, 6 (Dec 2004), 6–28. **Cited** on page 7.

[9] Al-Omari, S. A. K., and Sumari, P. An overview of mobile ad hoc networks for the existing protocols and applications. *CoRR abs/1003.3565* (2010). **Cited** on pages 5 and 13.

[10] Al-Rodhaan, M., Mackenzie, L., and Ould-Khaoua, M. A new route discovery algorithm for manets with chase packets. In *22nd Annual UK Performance Engineering Workshop* (2006), pp. 1–8. **Cited** on pages 12, 39 and 72.

[11] Al-Rodhaan, M. A., Mackenzie, L., and Ould-Khaoua, M. Improvement to blocking expanding ring search for manets. In *DCS Technical Report Series* (2008). **Cited** on pages 12, 79 and 169.

[12] AlAamri, H., Abolhasan, M., Wysocki, T., and Lipman, J. On optimising route discovery for multi-interface and power-aware nodes in heterogeneous manets. *International Conference on Wireless and Mobile Communications 0* (2010), 244–249. **Cited** on pages 29 and 68.

[13] Albert, R., and Barabasi, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys. 74* (Jan 2002), 47–97. **Cited** on page 56.

[14] Alemdar, H., and Ersoy, C. Wireless sensor networks for healthcare: A survey. *Computer Networks 54*, 15 (2010), 2688–2710. **Cited** on page 3.

[15] Almeida, P. S., Baquero, C., Preguica, N., and Hutchison, D. Scalable bloom filters. *Information Processing Letters 101*, 6 (2007), 255–261. **Cited** on pages 46, 112, 125 and 131.

[16] Alzaq, H., and Kabadayi, S. Mobile robot comes to the rescue in a wsn. In *IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)* (Sept 2013), pp. 1977–1982. **Cited** on page 4.

[17] Amgoth, T., and Jana, P. K. Energy-aware routing algorithm for wireless sensor networks. *Computers and Electrical Engineering 41* (2015), 357–367. **Cited** on pages 34 and 40.

[18] Anastasi, G., Conti, M., Francesco, M. D., and Passarella, A. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks 7*, 3 (2009), 537–568. **Cited** on page 5.

[19] Anjum, S. S., Noor, R. M., and Anisi, M. H. Survey on manet based communication scenarios for search and rescue operations. In *5th International Conference on IT Convergence and Security (ICITCS)* (Aug 2015), pp. 1–5. **Cited** on page 3.

[20] Avin, C., and Brito, C. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (New York, NY, USA, 2004), IPSN '04, ACM, pp. 277–286. **Cited** on pages 27, 34 and 39.

[21] Aziz, N., and Aziz, K. Managing disaster with wireless sensor networks. In *International Conference on Advanced Communication Technology (ICACT)* (Feb 2011), pp. 202–207. **Cited** on page 4.

[22] Babaei, H., Fathy, M., and Romoozi, M. Modeling and optimizing random walk content discovery protocol over mobile ad-hoc networks. *Performance Evaluation 74* (2014), 18–29. **Cited** on pages 34, 40 and 138.

[23] Baccour, N., Koubaa, A., Mottola, L., Zuniga, M. A., Youssef, H., Boano, C. A., and Alves, M. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transactions on Sensor Networks (TOSN) 8*, 4 (Sept. 2012), 34:1–34:33. **Cited** on page 31.

[24] Balister, P., Bollobas, B., Sarkar, A., and Walters, M. Highly connected random geometric graphs. *Discrete Applied Mathematics 157*, 2 (2009), 309–320. **Cited** on page 58.

[25] Balister, P., Sarkar, A., and Bollobas, B. Percolation, connectivity, coverage and colouring of random geometric graphs. In *Handbook of Large-Scale Random Networks*, B. Bollobas, R. Kozma, and D. Miklas, Eds., vol. 18 of *Bolyai Society Mathematical Studies*. Springer Berlin Heidelberg, 2008, pp. 117–142. **Cited** on page 100.

[26] Banerjee, N., Acharya, A., and Das, S. K. Enabling sip-based sessions in ad hoc networks. *Wireless Networks 13*, 4 (2007), 461–479. **Cited** on page 25.

[27] Baquero, C., Almeida, P., Menezes, R., and Jesus, P. Extrema propagation: Fast distributed estimation of sums and network sizes. *IEEE Transactions on Parallel and Distributed Systems PP*, 99 (2011), 1. **Cited** on page 176.

[28] Barritt, B., Malakooti, B., and null Zhihao Guo. Intelligent multiple-criteria broadcasting in mobile ad-hoc networks. *Annual IEEE Conference on Local Computer Networks 0* (2006), 761–768. **Cited** on page 41.

[29] Basagni, S., Chlamtac, I., Syrotiuk, V. R., and Woodward, B. A. A distance routing effect algorithm for mobility (dream). In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking* (New York, NY, USA, 1998), MobiCom '98, ACM, pp. 76–84. **Cited** on pages 39 and 135.

[30] Basili, V. R., Selby, R. W., and Hutchens, D. H. Experimentation in software engineering. *IEEE Transactions on Software Engineering 12* (July 1986), 733–743. **Cited** on page 15.

[31] Berndtsson, M., Hansson, J., Olsson, B., and Lundell, B. *Thesis Projects: A Guide for Students in Computer Science and Information Systems.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2008. **Cited** on page 21.

[32] Bilskie, M. V., Coggin, D., Hagen, S. C., and Medeiros, S. C. Terrain-driven unstructured mesh development through semi-automatic vertical feature extraction. *Advances in Water Resources 86, Part A* (2015), 102 – 118. **Cited** on page 1.

[33] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications ACM 13*, 7 (July 1970), 422–426. **Cited** on pages 41, 42, 109, 111, 114 and 140.

[34] Boldi, P., and Vigna, S. Mutable strings in java: design, implementation and lightweight text-search algorithms. *Science of Computer Programming 54*, 1 (2005), 3–23. Principles and Practice of Prgramming in Java (PPPJ 2003). **Cited** on page 42.

[35] Bonomi, F., Mitzenmacher, M., Panigrahy, R., Singh, S., and Varghese, G. An improved construction for counting bloom filters. In *Proceedings of the 14th Conference on Annual European Symposium* (London, UK, 2006), ESA'06, Springer-Verlag, pp. 684–695. **Cited** on pages 46 and 112.

[36] Boukerche, A. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks.* Wiley-IEEE Press, 2008. **Cited** on page 2.

[37] Boukerche, A., Turgut, B., Aydin, N., Ahmad, M. Z., Boloni, L., and Turgut, D. Routing protocols in ad hoc networks: A survey. *Computer Networks 55*, 13 (Sept. 2011), 3032–3080. **Cited** on pages 7, 35, 36, 68 and 70.

[38] Boyinbode, O., Le, H., Mbogho, A., Takizawa, M., and Poliah, R. A survey on clustering algorithms for wireless sensor networks. In *13th International Conference on Network-Based Information Systems (NBiS)* (Sept 2010), pp. 358–364. **Cited** on pages 34 and 40.

[39] Bradonjic, M., and Perkins, W. On sharp thresholds in random geometric graphs. *ArXiv e-prints* (Aug. 2013). **Cited** on page 58.

[40] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* (New York, NY, USA, 1998), MobiCom '98, ACM, pp. 85–97. **Cited** on page 32.

[41] Broder, A. Z., and Mitzenmacher, M. Network applications of bloom filters: A survey. *Internet Mathematics 1*, 4 (2003). **Cited** on pages 41 and 45.

[42] Callaway, E., Gorday, P., Hester, L., Gutierrez, J., Naeve, M., Heile, B., and Bahl, V. Home networking with ieee 802.15.4: a developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine 40*, 8 (Aug 2002), 70–77. **Cited** on page 28.

[43] Cartigny, J., and Simplot, D. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *Annual Hawaii International Conference on System Sciences* (Jan 2003), p. 10. **Cited** on pages 34, 39 and 48.

[44] Castelluccia, C., and Mutaf, P. Hash-based dynamic source routing. In *Networking*, N. Mitrou, K. Kontovasilis, G. Rouskas, I. Iliadis, and L. Merakos, Eds., vol. 3042 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 1012–1023. **Cited** on page 42.

[45] Castro, J., and Nasini, S. Mathematical programming approaches for classes of random network problems. *European Journal of Operational Research 245*, 2 (2015), 402–414. **Cited** on page 63.

[46] Cerf, V. G. On (computing) artifacts. *Communications ACM 58*, 9 (Aug. 2015), 7–7. **Cited** on page 12.

[47] Chakeres, I. D., and Belding-Royer, E. M. Aodv routing protocol implementation design. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops* (Washington, DC, USA, 2004), ICDCSW '04, IEEE Computer Society, pp. 698–703. **Cited** on page 47.

[48] Chang, N., and Liu, M. Revisiting the ttl-based controlled flooding search: optimality and randomization. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (New York, NY, USA, 2004), ACM, pp. 85–99. **Cited** on pages 13, 39, 71, 72, 76, 77, 84 and 137.

[49] Chang, R.-S., and Leu, S.-J. Long-lived path routing with received signal strength for ad hoc networks. In *International Symposium on Wireless Pervasive Computing* (Jan 2006), p. 6. **Cited** on page 40.

[50] Chang, Y.-K., Ting, Y.-W., and Wu, S.-C. Power-efficient and path-stable broadcasting scheme for wireless ad hoc networks. *International Conference on Advanced Information Networking and Applications Workshops 2* (2007), 707–712. **Cited** on page 40.

[51] Chau, C.-K. Finite random geometric graphs by circular and square coverage. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (June 2009), pp. 1–8. **Cited** on page 57.

[52] Chazelle, B., Kilian, J., Rubinfeld, R., and Tal, A. The bloomier filter: An efficient data structure for static support lookup tables. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2004), SODA '04, Society for Industrial and Applied Mathematics, pp. 30–39. **Cited** on pages 46, 111 and 112.

[53] Chen, L., Gu, Y., Guo, S., He, T., Shu, Y., Zhang, F., and Chen, J. Group-based discovery in low-duty-cycle mobile sensor networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)* (June 2012), pp. 542–550. **Cited** on page 28.

[54] Chen, Q., Kanhere, S. S., and Hassan, M. Performance analysis of geography-limited broadcasting in multihop wireless networks. *Wireless Communications and Mobile Computing 13*, 15 (2013), 1406–1421. **Cited** on pages 71 and 78.

[55] Chen, X., Ho, Y.-C., Zhang, J., and Bai, H. Probabilistic forwarding (profor) for large scale sensor networks. In *IEEE International Conference on Networking, Sensing and Control* (2006), pp. 963–968. **Cited** on page 39.

[56] Cheng, Z., and Heinzelman, W. B. Flooding strategy for target discovery in wireless networks. *Wireless Networks 11* (September 2005), 607–618. **Cited** on pages 7, 30, 39, 67, 68, 76, 77 and 85.

[57] Chengetanai, G., and O'Reilly, G. Survey on simulation tools for wireless mobile ad hoc networks. In *IEEE International Conference on Electrical, Computer and*

Communication Technologies (ICECCT) (March 2015), pp. 1–7. **Cited** on pages 14, 47 and 150.

[58] Chin, W. H., Fan, Z., and Haines, R. Emerging technologies and research challenges for 5g wireless networks. *IEEE Wireless Communications 21*, 2 (April 2014), 106–112. **Cited** on pages 2 and 28.

[59] Choi, W., Ghidini, G., and Das, S. K. A novel framework for energy-efficient data gathering with random coverage in wireless sensor networks. *ACM Transactions on Sensor Networks 8*, 4 (Sept. 2012), 36:1–36:30. **Cited** on page 34.

[60] Chu, M., Haussecker, H., Zhao, F., Chu, M., Haussecker, H., and Zhao, F. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications 16* (2002). **Cited** on page 137.

[61] Chung, J., Cho, K., Kim, H., Choi, M., Pak, J., Ha, N., Lee, B., and Han, K. An efficient topology configuration scheme for wireless sensor networks. In *Proceedings of the 6th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications* (Stevens Point, Wisconsin, USA, 2007), EHAC'07, World Scientific and Engineering Academy and Society (WSEAS), pp. 145–150. **Cited** on page 153.

[62] Cohen, S., and Matias, Y. Spectral bloom filters. In *ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2003), SIGMOD '03, ACM, pp. 241–252. **Cited** on page 112.

[63] Cordero, J. A., Clausen, T., and Baccelli, E. Mpr+sp: Towards a unified mpr-based manet extension for ospf. *Hawaii International Conference on System Sciences 0* (2011), 1–10. **Cited** on page 4.

[64] Dang, H., and Wu, H. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *IEEE Transactions on Wireless Communications 9*, 6 (june 2010), 1874–1881. **Cited** on pages 34 and 40.

[65] Dargie, W., and Poellabauer, C. *Fundamental of Wireless Sensor Networks: Theory and Practice.* John Wiley and Sons, 2010. **Cited** on page 10.

[66] Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., and Terry, D. Epidemic algorithms for replicated database maintenance. In *Annual ACM Symposium on Principles of Distributed Computing* (New York, NY, USA, 1987), PODC '87, ACM, pp. 1–12. **Cited** on page 35.

[67] Deng, Y., Shen, Y., Zhang, G., and Zhang, X. Development of a new broadcast protocol based on machine learning for wide-ranging manet environments. *International Symposium on Information Science and Engieering 1* (2008), 206–209. **Cited** on page 39.

[68] Denning, P. J. Is computer science science? *Communications ACM 48* (April 2005), 27–31. **Cited** on page 13.

[69] Dias-Gomez, J., Mitsche, D., and Perez, X. Sharp threshold for hamiltonicity of random geometric graphs. *CoRR abs/cs/0607023* (2006). **Cited** on page 57.

[70] Dodig-Crnkovic, G. Scientific methods in computer science. In *Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden* (April 2002). **Cited** on page 12.

[71] Dorogovtsev, S. N., and Mendes, J. F. F. *Evolution of Networks: From Biological Nets to the Internet and WWW (Physics)*. Oxford University Press, Inc., New York, NY, USA, 2003. **Cited** on page 1.

[72] Drabkin, V., Friedman, R., Kliot, G., and Segal, M. On reliable dissemination in wireless ad hoc networks. *IEEE Transactions on Dependable and Secure Computing 8*, 6 (nov.-dec. 2011), 866–882. **Cited** on pages 28, 34 and 41.

[73] Dubois-Ferriere, H., Grossglauser, M., and Vetterli, M. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In *ACM International Symposium on Mobile Ad Hoc Networking* (New York, NY, USA, 2003), MobiHoc '03, ACM, pp. 257–266. **Cited** on pages 36, 40, 41 and 137.

[74] Durresi, A., Paruchuri, V., Iyengar, S., and Kannan, R. Optimized broadcast protocol for sensor networks. *IEEE Transactions on Computers 54*, 8 (aug. 2005), 1013–1024. **Cited** on page 34.

[75] Ellis, C., Miranda, H., and Taiani, F. Count on me: lightweight ad-hoc broadcasting in heterogeneous topologies. In *Workshop on Middleware for Pervasive Mobile and Embedded Computing* (New York, NY, USA, 2009), M-PAC '09, ACM, pp. 1:1–1:6. **Cited** on page 29.

[76] Ellis, R. B., Martin, J. L., and Yan, C. Random geometric graph diameter in the unit ball. *Algorithmica 47*, 4 (2007), 421–438. **Cited** on page 57.

[77] Erdos, P., and Renyi, A. On the evolution of random graphs. In *Publication of the mathematical institute of the hungarian academy of sciences* (1960), pp. 17–61. **Cited** on page 55.

[78] Eslami, A., Nekoui, M., Pishro-Nik, H., and Fekri, F. Results on finite wireless sensor networks: Connectivity and coverage. *ACM Transactions on Sensor Networks 9*, 4 (July 2013), 51:1–51:22. **Cited** on pages 26 and 57.

[79] Fall, K., and Varadhan, K. ns notes and documentation, November 2011. **Cited** on page 47.

[80] Fan, B., Andersen, D. G., Kaminsky, M., and Mitzenmacher, M. D. Cuckoo filter: Practically better than bloom. In *ACM International on Conference on Emerging Networking Experiments and Technologies* (New York, NY, USA, 2014), CoNEXT '14, ACM, pp. 75–88. **Cited** on pages 113 and 175.

[81] Fan, L., Cao, P., Almeida, J., and Broder, A. Z. Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking 8*, 3 (June 2000), 281–293. **Cited** on page 111.

[82] Faruque, J., Psounis, K., and Helmy, A. Analysis of gradient-based routing protocols in sensor networks. In *Distributed Computing in Sensor Systems* (Berlin, Heidelberg, 2005), DCOSS'05, Springer-Verlag, pp. 258–275. **Cited** on page 138.

[83] Feeney, L. M. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications 6*, 3 (2001), 239–249. **Cited** on page 51.

[84] Feng, J., Lu, Y.-H., Jung, B., Peroulis, D., and Hu, Y. C. Energy-efficient data dissemination using beamforming in wireless sensor networks. *ACM Transactions on Sensor Networks 9*, 3 (June 2013), 31:1–31:30. **Cited** on page 28.

[85] Ferrari, F., Zimmerling, M., Mottola, L., and Thiele, L. Low-power wireless bus. In *ACM Conference on Embedded Network Sensor Systems* (New York, NY, USA, 2012), SenSys '12, ACM, pp. 1–14. **Cited** on page 24.

[86] Ferrari, F., Zimmerling, M., Thiele, L., and Saukh, O. Efficient network flooding and time synchronization with glossy. In *Information Processing in Sensor Networks (IPSN)* (April 2011), pp. 73–84. **Cited** on pages 13 and 134.

[87] Flajolet, P., Fusy, E., and Gandouet, O. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *International Conference On Analysis Of Algorithms* (2007). **Cited** on page 42.

[88] Font, J. L., Inigo, P., Dominguez, M., Sevillano, J. L., and Amaya, C. Analysis of source code metrics from ns-2 and ns-3 network simulators. *Simulation Modelling Practice and Theory 19*, 5 (2011), 1330–1346. Not cited.

[89] Friedman, R., Kermarrec, A.-M., Miranda, H., and Rodrigues, L. Gossip-based dissemination. In *Middleware for Network Eccentric and Mobile Applications*, B. Garbinato, H. Miranda, and L. Rodrigues, Eds. Springer Berlin Heidelberg, 2009, pp. 169–190. 10.1007/978-3-540-89707-1_8. **Cited** on page 28.

[90] Furno, A., and Zimeo, E. Self-scaling cooperative discovery of service compositions in unstructured p2p networks. *Journal of Parallel and Distributed Computing 74*, 10 (2014), 2994–3025. **Cited** on pages 28 and 34.

[91] Ganesan, D., Cerpa, A., Ye, W., Yu, Y., Zhao, J., and Estrin, D. Networking issues in wireless sensor networks. *Journal of Parallel and Distributed Computing 64*, 7 (2004), 799–814. Computing and Communication in Distributed Sensor Networks. **Cited** on pages 4 and 41.

[92] Garbinato, B., Holzer, A., and Vessaz, F. Six-shot broadcast: A context-aware algorithm for efficient message diffusion in manets. In *Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE* (Berlin, Heidelberg, 2008), OTM '08, Springer-Verlag, pp. 625–638. **Cited** on pages 27 and 40.

[93] Garbinato, B., Holzer, A., and Vessaz, F. Context-aware broadcasting approaches in mobile ad hoc networks. *Computer Networks 54* (May 2010), 1210–1228. **Cited** on pages 40 and 72.

[94] Garbinato, B., Holzer, A., and Vessaz, F. Six-shot multicast: A location-aware strategy for efficient message routing in manets. In *Network Computing and Applications (NCA)* (july 2010), pp. 1–9. **Cited** on pages 40 and 41.

[95] Garcia de Oliveira, V., Rogerio Salvador, M., Takayassu, D., Felix, V., and Fonseca, N. Applying bloom filter addressing in metro ethernet network. In *Advanced Information Networking and Applications Workshops (WAINA)* (April 2010), pp. 42–49. **Cited** on page 42.

[96] Gargano, L., and Hammar, M. Limiting flooding expenses in on-demand source-initiated protocols for mobile wireless networks. In *Parallel and Distributed Processing Symposium* (April 2004), p. 220. **Cited** on pages 12, 71, 73 and 78.

[97] Gerla, M., Kwon, T. J., and Pei, G. On-demand routing in large ad hoc wireless networks with passive clustering. In *Wireless Communications and Networking Conference* (2000), vol. 1, pp. 100–105. **Cited** on page 34.

[98] Goel, A., Rai, S., and Krishnamachari, B. Monotone properties of random geometric graphs have sharp thresholds. *Annals pf Applied Probability 15*, 4 (11 2005), 2535–2552. **Cited** on page 58.

[99] Gowers, W. T. Hypergraph regularity and the multidimensional szemeredi theorem. *ArXiv e-prints* (Oct. 2007). **Cited** on page 55.

[100] Guo, D., He, Y., and Liu, Y. On the feasibility of gradient-based data-centric routing using bloom filters. *IEEE Transactions on Parallel and Distributed Systems 25*, 1 (Jan 2014), 180–190. **Cited** on pages 40, 41, 42, 111, 113, 136, 139 and 160.

[101] Guo, D., Wu, J., Chen, H., Yuan, Y., and Luo, X. The dynamic bloom filters. *IEEE Transactions on Knowledge and Data Engineering, 22*, 1 (Jan 2010), 120–133. **Cited** on page 112.

[102] Guo, S., and Yang, O. Maximizing multicast communication lifetime in wireless mobile ad hoc networks. *IEEE Transactions on Vehicular Technology 57*, 4 (july 2008), 2414–2425. **Cited** on page 29.

[103] Haas, Z. J., Halpern, J. Y., and Li, L. Gossip-based ad hoc routing. *IEEE/ACM Transactions Networks 14* (June 2006), 479–491. **Cited** on page 68.

[104] Haas, Z. J., and Pearlman, M. R. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Transactions on Networking (TON) 9*, 4 (Aug. 2001), 427–438. **Cited** on pages 30, 36, 73 and 77.

[105] Haenggi, M., Andrews, J., Baccelli, F., Dousse, O., and Franceschetti, M. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications, 27*, 7 (September 2009), 1029–1046. **Cited** on pages 14, 57 and 156.

[106] Hammoudeh, M., and Newman, R. Information extraction from sensor networks using the watershed transform algorithm. *Information Fusion 22* (2015), 39–49. **Cited** on page 29.

[107] Han, S. Y., and Lee, D. An adaptive hello messaging scheme for neighbor discovery in on-demand manet routing protocols. *IEEE Communications Letters 17*, 5 (May 2013), 1040–1043. **Cited** on page 138.

[108] Hassan, J., and Jha, S. On the optimization trade-offs of expanding ring search. In *Conference on Distributed Computing* (Berlin, Heidelberg, 2004), IWDC'04, Springer-Verlag, pp. 489–494. **Cited** on pages 76 and 77.

[109] Hassan, J., and Jha, S. Performance analysis of expanding ring search for multi-hop wireless networks. In *Vehicular Technology Conference* (sept. 2004), vol. 5, pp. 3615–3619. **Cited** on page 39.

[110] Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In *Hawaii International Conference on System Sciences* (Jan 2000), p. 10. **Cited** on pages 34, 36, 40 and 41.

[111] Henke, C., Schmoll, C., and Zseby, T. Empirical evaluation of hash functions for multipoint measurements. *SIGCOMM Computer Communication Review 38*, 3 (July 2008), 39–50. **Cited** on page 42.

[112] Hodge, V., O'Keefe, S., Weeks, M., and Moulds, A. Wireless sensor networks for condition monitoring in the railway industry: A survey. *IEEE Transactions on Intelligent Transportation Systems 16*, 3 (June 2015), 1088–1106. **Cited** on page 2.

[113] Holzer, A., Vessaz, F., Pierre, S., and Garbinato, B. Plan-b: Proximity-based lightweight adaptive network broadcasting. In *International Symposium on Network Computing and Applications (NCA)* (aug. 2011), pp. 265–270. **Cited** on pages 27, 40 and 41.

[114] Hossain, M. K., Datta, S., and Edmonds, J. Ad-atma: An efficient mac protocol for wireless sensor and ad hoc networks. *Procedia Computer Science 52* (2015), 484–491. International Conference on Ambient Systems, Networks and Technologies. **Cited** on page 32.

[115] Hosseininezhad, S., Shirazi, G., and Leung, V. Rlab: Reinforcement learning-based adaptive broadcasting for vehicular ad-hoc networks. In *Vehicular Technology Conference* (may 2011), pp. 1–5. **Cited** on page 41.

[116] Huang, H., Hartman, J., and Hurst, T. Data-centric routing in sensor networks using biased walk. In *IEEE Communications Society on Sensor and Ad Hoc Communications and Networks* (sept. 2006), vol. 1, pp. 1–9. **Cited** on page 13.

[117] Huang, K., Zhang, J., Zhang, D., Xie, G., Salamatian, K., Liu, A., and Li, W. A multi-partitioning approach to building fast and accurate counting bloom filters. In *International Symposium on Parallel Distributed Processing (IPDPS)* (May 2013), pp. 1159–1170. **Cited** on page 112.

[118] Huang, Q., Bai, Y., and Chen, L. Efficient lightweight broadcasting protocols for multi-hop ad hoc networks. In *International Symposium on Personal, Indoor and Mobile Radio Communications* (Sept 2006), pp. 1–5. **Cited** on page 4.

[119] hyeon Lee, D., ho Oh, J., and il Jung, J. An enhanced selective forwarding scheme for real-time applications in manets. *Information Science and Applications, International Conference on 0* (2011), 1–11. **Cited** on page 34.

[120] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *International Conference on Mobile Computing and Networking* (New York, NY, USA, 2000), MobiCom '00, ACM, pp. 56–67. **Cited** on pages 138 and 140.

[121] Ishibashi, B., and Boutaba, R. Topology and mobility considerations in mobile ad hoc networks. *Ad Hoc Networks 3*, 6 (Nov. 2005), 762–776. **Cited** on page 6.

[122] Jang, H.-C., Lien, Y.-N., and Tsai, T.-C. Rescue information system for earthquake disasters based on manet emergency communication platform. In *International Conference on Wireless Communications and Mobile Computing* (New York, NY, USA, 2009), IWCMC '09, ACM, pp. 623–627. **Cited** on page 3.

[123] Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., and van Steen, M. Gossip-based peer sampling. *ACM Transactions on Computer Systems (TOCS) 25*, 3 (Aug. 2007). **Cited** on page 35.

[124] Jiang, H., Chen, L., Wu, J., Chen, S., and Leung, H. A reliable and high-bandwidth multihop wireless sensor network for mine tunnel monitoring. *IEEE Sensors Journal 9*, 11 (nov. 2009), 1511–1517. **Cited** on page 3.

[125] Johnson, D. B., Maltz, D. A., and Broch, J. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking* (2001), Addison-Wesley, pp. 139–172. **Cited** on pages 7, 13, 30, 36, 39, 71, 73, 85 and 137.

[126] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences

with zebranet. *SIGOPS Operating Systems Review 36* (October 2002), 96–107. **Cited** on page 40.

[127] Kalmar, A., Vida, R., and Maliosz, M. Context-aware addressing in the internet of things using bloom filters. In *Cognitive Infocommunications (CogInfoCom)* (Dec 2013), pp. 487–492. **Cited** on page 113.

[128] Kang, J. G., Jang, J. W., Yun, C. H., and Lee, Y. W. A network management system for u-city. In *Advanced Communication Technology (ICACT)* (feb. 2011), pp. 279–283. **Cited** on page 24.

[129] Kapoor, H., Mehta, K., Puri, D., and Saxena, S. Survey of various search mechanisms in unstructured peer-to-peer networks. *International Journal of Computer Applications 68*, 6 (April 2013), 21–25. Full text available. **Cited** on page 28.

[130] Karlsson, G., Lenders, V., and May, M. Delay-tolerant broadcasting. *IEEE Transactions on Broadcasting 53*, 1 (march 2007), 369–381. **Cited** on page 32.

[131] Kasamatsu, D., Kawamura, Y., Oki, M., and Shinomiya, N. Broadcasting method based on topology control for fault-tolerant manet. *International Conference on Distributed Computing Systems 0* (2011), 105–110. **Cited** on pages 26 and 72.

[132] Kasamatsu, D., Shinomiya, N., and Ohta, T. A broadcasting method considering battery lifetime and distance between nodes in manet. *International Conference on Distributed Computing Systems 0* (2009), 406–411. **Cited** on page 40.

[133] Kenniche, H., and Ravelomananana, V. Random geometric graphs as model of wireless sensor networks. In *Computer and Automation Engineering (ICCAE)* (feb. 2010), vol. 4, pp. 103–107. **Cited** on pages 86 and 156.

[134] Kermarrec, A.-M., and van Steen, M. Gossiping in distributed systems. *SIGOPS Operating Systems Review 41*, 5 (Oct. 2007), 2–7. **Cited** on page 35.

[135] Khan, M., Askwith, B., Bouhafs, F., and Asim, M. Limitations of simulation tools for large-scale wireless sensor networks. In *Advanced Information Networking and Applications (WAINA)* (march 2011), pp. 820–825. **Cited** on pages 46 and 47.

[136] Kitchenham, B., Al-Khilidar, H., Babar, M. A., Berry, M., Cox, K., Keung, J., Kurniawati, F., Staples, M., Zhang, H., and Zhu, L. Evaluating guidelines for empirical software engineering studies. In *International Symposium on Empirical Software Engineering* (New York, NY, USA, 2006), ISESE '06, ACM, pp. 38–47. **Cited** on page 16.

[137] Ko, Y.-B., and Vaidya, N. Geocasting in mobile ad hoc networks: location-based multicast algorithms. In *Mobile Computing Systems and Applications* (Feb 1999), pp. 101–110. **Cited** on page 135.

[138] Ko, Y.-B., and Vaidya, N. H. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks 6*, 4 (July 2000), 307–321. **Cited** on pages 30, 34, 36, 40, 71, 73, 77 and 135.

[139] Korf, R. E. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence 27*, 1 (Sept. 1985), 97–109. **Cited** on page 38.

[140] Kotz, D., Newport, C., Gray, R. S., Liu, J., Yuan, Y., and Elliott, C. Experimental evaluation of wireless simulation assumptions. In *Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (New York, NY, USA, 2004), MSWiM '04, ACM, pp. 78–82. **Cited** on page 47.

[141] Krishnamachari, B., and Ahn, J. Optimizing data replication for expanding ring-based queries in wireless sensor networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (april 2006), pp. 1–10. **Cited** on page 76.

[142] Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C., and Zhao, B. Oceanstore: An architecture for global-scale persistent storage. *SIGPLAN Notices 35*, 11 (Nov. 2000), 190–201. **Cited** on page 110.

[143] Kuhn, F., Moscibroda, T., and Wattenhofer, R. Initializing newly deployed ad hoc and sensor networks. In *International Conference on Mobile Computing and Networking* (New York, NY, USA, 2004), MobiCom '04, ACM, pp. 260–274. **Cited** on page 34.

[144] Kumar, A., Xu, J., and Zegura, E. Efficient and scalable query routing for unstructured peer-to-peer networks. In *INFOCOM:Computer and Communications Societies* (March 2005), vol. 2, pp. 1162–1173. **Cited** on pages 28, 34 and 114.

[145] Kumar Deva Sarma, H., Rai, P., and Deka, B. Energy efficient communication protocol for wireless sensor networks with mobile node. In *Recent Advances and Innovations in Engineering (ICRAIE)* (May 2014), pp. 1–6. **Cited** on page 34.

[146] Kurkowski, S., Camp, T., and Colagrosso, M. Manet simulation studies: the incredibles. *SIGMOBILE Mobile Computing and Communications Review 9* (October 2005), 50–61. **Cited** on page 134.

[147] Lamport, L. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986. **Cited** on page 21.

[148] Lau, B. C., Ma, E. W., and Chow, T. W. Probabilistic fault detector for wireless sensor network. *Expert Systems with Applications 41*, 8 (2014), 3703–3711. **Cited** on page 5.

[149] Laufer, R. P., Velloso, P. B., and Duarte, O. C. M. B. A generalized bloom filter to secure distributed network applications. *SIGMOBILE Mobile Computing and Communications Review 55*, 8 (June 2011), 1804–1819. **Cited** on page 113.

[150] Levis, P., Lee, N., Welsh, M., and Culler, D. Tossim: Accurate and scalable simulation of entire tinyos applications. In *International Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2003), SenSys '03, ACM, pp. 126–137. **Cited** on page 47.

[151] Levis, P., Patel, N., Culler, D., and Shenker, S. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Symposium on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2004), NSDI'04, USENIX Association, pp. 2–2. **Cited** on pages 35 and 39.

[152] Li, H., Wu, C., Hua, Q.-S., and Lau, F. C. Latency-minimizing data aggregation in wireless sensor networks under physical interference model. *Ad Hoc Networks 12* (2014), 52–68. SCEnarios for ad hoc Network Evaluation Studies (SCENES). **Cited** on page 27.

[153] Li, M., and Liu, Y. Underground coal mine monitoring with wireless sensor networks. *ACM Transactions Sensor Networks 5*, 2 (April 2009), 10:1–10:29. **Cited** on page 135.

[154] Li, X., Nayak, A., and Stojmenovic, I. Exploiting actuator mobility for energy-efficient data collection in delay-tolerant wireless sensor networks. In *International Conference on Networking and Services* (april 2009), pp. 216–221. **Cited** on page 24.

[155] Li, X., Wu, J., and Xu, J. Hint-based routing in wsns using scope decay bloom filters. In *International Workshop on Networking, Architecture, and Storages* (2006), p. 8. **Cited** on pages 34 and 109.

[156] Liarokapis, D., Shahrabi, A., and Komninos, A. Diba: An adaptive broadcasting scheme in mobile ad hoc networks. *Communication Networks and Services Research 0* (2009), 224–231. **Cited** on pages 34 and 40.

[157] Lim, H., Fan, B., Andersen, D. G., and Kaminsky, M. Silt: A memory-efficient, high-performance key-value store. In *ACM Symposium on Operating Systems Principles* (New York, NY, USA, 2011), SOSP '11, ACM, pp. 1–13. **Cited** on page 113.

[158] Lima, R. Broadcast cancellation in unstructured networks. `https://github.com/rmlima/BCUN`, 2017. **Cited** on pages 20, 23 and 150.

[159] Lima, R., Baquero, C., and Miranda, H. Stopping ongoing broadcasts in large manets. In *European Workshop on AppRoaches to MObiquiTous Resilience* (New York, NY, USA, 2012), ARMOR '12, ACM, pp. 4:1–4:5. **Cited** on pages 47 and 73.

[160] Lima, R., Baquero, C., and Miranda, H. Broadcast cancellation in search mechanisms. In *Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2013), SAC '13, ACM, pp. 548–553. **Cited** on pages 39, 88, 133, 140 and 147.

[161] Lima, R., Baquero, C., and Miranda, H. Adaptive broadcast cancellation query mechanism for unstructured networks. In *International Conference on Next Generation Mobile Applications, Services and Technologies* (Sept 2015), pp. 176–181. **Cited** on pages 41 and 118.

[162] Lima, R., Baquero, C., and Miranda, H. Filtro de bloom linear. In *Simposio Nacional de Informatica (INFORUM 2015)* (Covilha, Portugal, Setembro 2015). **Cited** on pages 140 and 141.

[163] Lin, G.-R., Fan, Y.-C., Wang, E. T., Zou, T., and Chen, A. Energy-efficient sensor data acquisition based on periodic patterns. In *Parallel and Distributed Systems (ICPADS)* (Dec 2009), pp. 487–494. **Cited** on page 135.

[164] Lin, H., Lu, M., Milosavljevic, N., Gao, J., and Guibas, L. Composable information gradients in wireless sensor networks. In *Information Processing in Sensor Networks* (April 2008), pp. 121–132. **Cited** on page 136.

[165] Liu, F., and Heijenk, G. Context discovery using attenuated bloom filters in ad-hoc networks. In *Wired/Wireless Internet Communications*, T. Braun, G. Carle, S. Fahmy, and Y. Koucheryavy, Eds., vol. 3970 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 13–25. **Cited** on pages 111 and 113.

[166] Liu, J., Zhao, F., and Petrovic, D. Information-directed routing in ad hoc sensor networks. *IEEE Journal on Selected Areas in Communications 23*, 4 (April 2005), 851–861. **Cited** on page 139.

[167] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys Tutorials 7*, 2 (Apr. 2005), 72–93. **Cited** on page 28.

[168] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. Search and replication in unstructured peer-to-peer networks. In *International Conference on Supercomputing* (New York, NY, USA, 2002), ICS '02, ACM, pp. 84–95. **Cited** on pages 34 and 138.

[169] Lynch, N. A. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996. **Cited** on pages 48, 50 and 51.

[170] Macker, J. P., Dean, J., and Chao, W. Simplified multicast forwarding in mobile ad hoc networks. In *MILITARY COMMUNICATIONS CONFERENCE (MILCOM)* (Oct. 2004), vol. 2, pp. 744–750. **Cited** on page 35.

[171] Mahfoudh, S., and Minet, P. An energy efficient routing based on olsr in wireless ad hoc and sensor networks. In *Advanced Information Networking and Applications* (march 2008), pp. 1253–1259. **Cited** on page 40.

[172] Mahfoudh, S., and Minet, P. Survey of energy efficient strategies in wireless ad hoc and sensor networks. In *International Conference on Networking* (april 2008), pp. 1–7. **Cited** on page 5.

[173] Maraiya, K., Kant, K., and Gupta, N. Application based study on wireless sensor network. *International Journal of Computer Applications 21*, 8 (May 2011), 9–15. Full text available. **Cited** on pages 1 and 24.

[174] Maroti, M. Directed flood-routing framework for wireless sensor networks. In *ACM/IFIP/USENIX International Conference on Middleware* (New York, NY, USA, 2004), Middleware '04, Springer-Verlag New York, Inc., pp. 99–114. **Cited** on page 135.

[175] Matavire, R., and Brown, I. Investigating the use of grounded theory in information systems research. In *South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology* (New York, NY, USA, 2008), SAICSIT '08, ACM, pp. 139–147. **Cited** on page 16.

[176] Maurya, M., and Shukla, S. R. Current wireless sensor nodes (motes): Performance metrics and constraints. *International Journal of Advanced Research in Electronics and Communication Engineering 2*, 1 (2013), 45. **Cited** on page 24.

[177] Meshkova, E., Riihijarvi, J., Petrova, M., and Mahonen, P. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Compututer Networks 52*, 11 (Aug. 2008), 2097–2128. **Cited** on pages 28 and 29.

[178] Miranda, H. *Gossip-Based Data Distribution in Mobile Ad Hoc Networks.* PhD thesis, Universidade de Lisboa, Campo Grande, 1749-016 Lisboa - Portugal, oct 2007. **Cited** on page 150.

[179] Miranda, H., Leggio, S., Rodrigues, L., and Raatikainen, K. A power-aware broadcasting algorithm. In *International Symposium on Personal, Indoor and Mobile Radio Communications* (Sept 2006), pp. 1–5. **Cited** on pages 13, 40 and 41.

[180] Miranda, H., Leggio, S., Rodrigues, L., and Raatikainen, K. An algorithm for dissemination and retrieval of information in wireless ad hoc networks. In *Euro-Par 2007 Parallel Processing*, A.-M. Kermarrec, L. Bouga, and T. Priol, Eds., vol. 4641 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, pp. 891–900. 10.1007/978-3-540-74466-5_96. **Cited** on page 28.

[181] Mishra, S., Singh, J., Dixit, A., and Prakash, S. Modified expanding ring search algorithm for ad-hoc networks. *International Journal of Computer Science and Information Technologies 3.3* (2012), 4067–4070. **Cited** on page 67.

[182] Misra, S., Woungang, I., and Misra, S. C. *Guide to Wireless Ad Hoc Networks*, 1st ed. Springer Publishing Company, Incorporated, 2009. **Cited** on page 1.

[183] Mo, L., He, Y., Liu, Y., Zhao, J., Tang, S.-J., Li, X.-Y., and Dai, G. Canopy closure estimates with greenorbs: Sustainable sensing in the forest. In *ACM Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2009), SenSys '09, ACM, pp. 99–112. **Cited** on page 3.

[184] Mohammed, A., Ould-Khaoua, M., Mackenzie, L., and Abdulai, J.-D. Dynamic probabilistic counter-based broadcasting in mobile ad hoc networks. In *International Conference on Adaptive Science Technology* (jan. 2009), pp. 120–127. **Cited** on pages 34 and 39.

[185] Mohapatra, D., and Suma, S. Survey of location based wireless services. In *International Conference on Personal Wireless Communications* (Jan 2005), pp. 358–362. **Cited** on page 71.

[186] Moore, E. F. The shortest path through a maze. *Proceedings of an International Symposium on the Theory of Switching* (April 1957), 285–292. **Cited** on page 38.

[187] Moreno, C. B., Almeida, P. S., and Cardoso, J. Probabilistic estimation of network size and diameter. In *Fourth Latin-American Symposium on Dependable Computing - LADC* (Joao Pessoa, Paraiba, Brazil, September 2009), IEEE, IEEE, pp. 33–40. **Cited** on pages 26 and 35.

[188] Morshed, M., Ko, F., Lim, D., Rahman, M., Mazumder, M., and Ghosh, J. Performance evaluation of dsdv and aodv routing protocols in mobile ad-hoc networks. In *New Trends in Information Science and Service Science (NISS)* (may 2010), pp. 399–403. **Cited** on page 48.

[189] Mottola, L., and Picco, G. Muster: Adaptive energy-aware multisink routing in wireless sensor networks. *IEEE Transactions on Mobile Computing 10*, 12 (Dec 2011), 1694–1709. **Cited** on page 134.

[190] Mottola, L., and Picco, G. P. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Computers Surveys (CSUR) 43*, 3 (April 2011), 19:1–19:51. **Cited** on page 2.

[191] N., Y. B., T., M., and Chiplunkar, N. N. Enhancing the performance of manets by monitoring the energy consumption and use of mobile relays. *International Association of Computer Science and Information Technology 0* (2009), 85–89. **Cited** on page 35.

[192] Nagaraju, A., and Srinadas, R. A strategy to reduce the control packet load of aodv using weighted rough set model for manet. *Int. Arab J. Inf. Technol. 8*, 1 (2011), 108–116. **Cited** on page 138.

[193] Naruephiphat, W., and Charnsripinyo, C. Routing algorithm for balancing network lifetime and reliable packet delivery in mobile ad hoc networks. *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing 0* (2009), 257–262. **Cited** on page 40.

[194] Nath, S., Ekambaram, V. N., Kumar, A., and Kumar, P. V. Theory and algorithms for hop-count-based localization with random geometric graph models of dense sensor networks. *ACM Transactions Sensor Networks 8*, 4 (Sept. 2012), 35:1–35:38. **Cited** on page 58.

[195] Newell, A., and Simon, H. A. Computer science as empirical inquiry: symbols and search. *Communications ACM 19* (March 1976), 113–126. **Cited** on page 24.

[196] Ngai, E. C.-H., and Gunningberg, P. Quality-of-information-aware data collection for mobile sensor networks. *Pervasive and Mobile Computing 11* (2014), 203–215. **Cited** on page 34.

[197] Okdem, S., and Karaboga, D. Routing in wireless sensor networks using an ant colony optimization (aco) router chip. *Sensors 9*, 2 (2009), 909. **Cited** on page 77.

[198] Ouksel, A., and Lundquist, D. Reducing redundant data transmissions in wireless ad hoc networks: Comparing aggregation and filtering. *Wireless Networks 21*, 7 (Oct. 2015), 2155–2168. **Cited** on pages 30 and 35.

[199] Outay, F. O., Veque, V. V., and Bouallegue, R. B. An efficient performance comparison of two cross-layer service discovery designs in manet: Bf-sd-zrp versus sd-olsr. *IEEE Conference on Local Computer Networks 0* (2010), 827–834. **Cited** on page 37.

[200] Ovalle-Martinez, F. J., Nayak, A., Stojmenovic, I., Carle, J., and Simplot-Ryl, D. Area-based beaconless reliable broadcasting in sensor networks. *International Journal of Sensor Networks 1* (September 2006), 20–33. **Cited** on page 40.

[201] Pal, S., Bhattacharyya, D., Tomar, G., and hoon Kim, T. Wireless sensor networks and its routing protocols: A comparative study. In *Computational Intelligence and Communication Networks (CICN)* (Nov 2010), pp. 314–319. **Cited** on page 35.

[202] Palsberg, J., and Baxter, S. J. Teaching reviewing to graduate students. *Communications ACM 45* (December 2002), 22–24. **Cited** on page 16.

[203] Park, I., Kim, J., and Pu, I. Blocking expanding ring search algorithm for efficient energy consumption in mobile ad hoc networks. In *Wireless On-demand Network Systems and Services* (Les Menuires (France), Jan. 2006), INRIA, INSA Lyon, IFIP, Alcatel, pp. 191–195. http://citi.insa-lyon.fr/wons2006/index.html. **Cited** on pages 12, 13, 39, 67, 72, 79, 85, 140, 147 and 169.

[204] Park, I., and Pu, I. Energy efficient expanding ring search. In *International Conference on Modelling Simulation* (march 2007), pp. 198–199. **Cited** on pages 13 and 37.

[205] Park, V. D., and Corson, M. S. A highly adaptive distributed routing algorithm for mobile wireless networks. In *IEEE Computer and Communications Societies* (Washington, DC, USA, 1997), INFOCOM '97, IEEE Computer Society, p. 1405. **Cited** on page 36.

[206] Pathan, A., Lee, H.-W., and Hong, C. S. Security in wireless sensor networks: issues and challenges. In *Advanced Communication Technology (ICACT)* (Feb 2006), vol. 2, p. 1048. **Cited** on page 29.

[207] Penrose, M. D. The longest edge of the random minimal spanning tree. *Annals of Applied Probability 7*, 2 (05 1997), 340–361. **Cited** on page 57.

[208] Perera, C., Liu, C., and Jayawardena, S. The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing, PP*, 99 (2015), 1–1. **Cited** on pages 2 and 37.

[209] Perich, F., Undercoffer, J., Kagal, L., Joshi, A., Finin, T., and Yesha, Y. In reputation we believe: query processing in mobile ad-hoc networks. In *Mobile and Ubiquitous Systems: Networking and Services* (aug. 2004), pp. 326–334. **Cited** on page 7.

[210] Perkins, C. E., and Bhagwat, P. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Communications Architectures, Protocols and Applications* (New York, NY, USA, 1994), SIGCOMM '94, ACM, pp. 234–244. **Cited** on pages 34, 36 and 40.

[211] Perkins, C. E., and Royer, E. M. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications* (New Orleans, LA, USA, Feb. 1999), pp. 90–100. **Cited** on pages 13, 30, 34, 36, 73, 77 and 138.

[212] Pham, C. Communication performance of low-resource sensor motes for data-intensive applications. In *Wireless Days (WD)* (Nov 2013), pp. 1–8. **Cited** on page 24.

[213] Pham, N. D., and Choo, H. Efficient location-aided flooding scheme initiated by receiver for manets. In *Ubiquitous Information Management and Communication* (New York, NY, USA, 2009), ICUIMC '09, ACM, pp. 194–199. **Cited** on pages 34 and 78.

[214] Polastre, J., Szewczyk, R., and Culler, D. Telos: Enabling ultra-low power wireless research. In *International Symposium on Information Processing in Sensor Networks* (Piscataway, NJ, USA, 2005), IPSN '05, IEEE Press. **Cited** on page 24.

[215] Pottie, G. J., and Kaiser, W. J. Wireless integrated network sensors. *Communications ACM 43*, 5 (May 2000), 51–58. **Cited** on page 5.

[216] Preethi, S., and Ramachandran, B. Energy efficient routing protocols for mobile adhoc networks. In *Emerging Trends in Networks and Computer Communications (ET-NCC)* (april 2011), pp. 136–141. **Cited** on pages 48 and 72.

[217] Pu, I., Kim, J., and Shen, Y. Energy-time efficiency of two routing strategies with chase packets in expanding ring search. In *International Conference on Telecommunications (ICT)* (april 2010), pp. 742–747. **Cited** on pages 12, 35, 68 and 108.

[218] Pu, I., and Shen, Y. Enhanced blocking expanding ring search in mobile ad hoc networks. In *New Technologies, Mobility and Security (NTMS)* (dec. 2009), pp. 1–5. **Cited** on pages 67, 79, 83, 85, 92 and 99.

[219] Pu, I., and Shen, Y. A framework for chase strategies in recent energy or time efficient route discovery protocols for manets. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (june 2011), pp. 1–4. **Cited** on pages 12 and 39.

[220] Pu, I. M., Stamate, D., and Shen, Y. Improving time-efficiency in blocking expanding ring search for mobile ad hoc networks. *Journal of Discrete Algorithms 24* (Jan. 2014), 59–67. **Cited** on pages 13, 27 and 108.

[221] Puccinelli, D., Giordano, S., Zuniga, M., and Marron, P. J. Broadcast-free collection protocol. In *ACM Conference on Embedded Network Sensor Systems* (New York, NY, USA, 2012), SenSys '12, ACM, pp. 29–42. **Cited** on page 29.

[222] Putze, F., Sanders, P., and Singler, J. Cache-, hash-, and space-efficient bloom filters. *Journal of Experimental Algorithmics (JEA) 14* (Jan. 2010), 4:4.4–4:4.18. **Cited** on page 42.

[223] Qiao, Y., Li, T., and Chen, S. Fast bloom filters and their generalization. *IEEE Transactions on Parallel and Distributed Systems 25*, 1 (Jan 2014), 93–103. **Cited** on pages 46 and 112.

[224] Rafiq, M., Kumar, M., and Ahmad, S. Comparative analysis of energy efficient routing protocols in manet. *International Journal of Computer Applications 82*, 5 (November 2013), 1–6. Full text available. **Cited** on page 32.

[225] Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine 19*, 2 (Mar 2002), 40–50. **Cited** on page 5.

[226] Rahman, M. A., Pakstas, A., and Wang, F. Z. Network modelling and simulation tools. *Simulation Modelling Practice and Theory 17*, 6 (2009), 1011–1031. **Cited** on page 46.

[227] Rhea, S., and Kubiatowicz, J. Probabilistic location and routing. In *Annual Joint Conference of the IEEE Computer and Communications Societies* (2002), vol. 3, pp. 1248–1257. **Cited** on pages 34, 46, 113 and 139.

[228] Rogers, P., and Abu-Ghazaleh, N. Robustness of network-wide broadcasts in manets. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference 0* (2005), 185. **Cited** on page 30.

[229] Rottenstreich, O., Kanizo, Y., and Keslassy, I. The variable-increment counting bloom filter. *IEEE/ACM Transactions on Networking 22*, 4 (Aug 2014), 1092–1105. **Cited** on page 112.

[230] Rout, A., Sethi, S., and Mishra, D. Optimized ant based routing protocol for manet. In *International Conference on Communication, Computing and Security* (New York, NY, USA, 2011), ICCCS '11, ACM, pp. 21–25. **Cited** on pages 40 and 77.

[231] Ruiz, P., and Bouvry, P. Survey on broadcast algorithms for mobile ad hoc networks. *ACM Computing Surveys (CSUR) 48*, 1 (July 2015), 8:1–8:35. **Cited** on page 5.

[232] Sabet, M., and Naji, H. R. A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks. *AEU - International Journal of Electronics and Communications* (2015). **Cited** on pages 34 and 40.

[233] Savvaki, S., Tsagkatakis, G., Panousopoulou, A., and Tsakalides, P. Application of matrix completion on water treatment data. In *International Workshop on Cyber-Physical Systems for Smart Water Networks* (New York, NY, USA, 2015), CySWater'15, ACM, pp. 3:1–3:6. **Cited** on page 3.

[234] Schilling, B. Qualitative comparison of network simulation tools. *Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart* (2005). **Cited** on page 47.

[235] Schneider, J., and Wattenhofer, R. Coloring unstructured wireless multi-hop networks. In *ACM Symposium on Principles of Distributed Computing* (New York, NY, USA, 2009), PODC '09, ACM, pp. 210–219. **Cited** on page 39.

[236] SCHNITZER, S., MIRANDA, H., AND KOLDEHOFE, B. Content routing algorithms to support publish/subscribe in mobile ad hoc networks. In *Local Computer Networks (LCN 2012 Workshops)* (Oct. 2012), pp. 1053–1060. **Cited** on page 28.

[237] SETTLE, R., AND SETTLE, M. *The story of the Pony Express*. Foulsham, 1955. **Cited** on page 7.

[238] SGORA, A., VERGADOS, D. J., AND VERGADOS, D. D. A survey of tdma scheduling schemes in wireless multihop networks. *ACM Computing Surveys (CSUR) 47*, 3 (Apr. 2015), 53:1–53:39. **Cited** on page 32.

[239] SHAH, S., KHANDRE, A., SHIROLE, M., AND BHOLE, G. Performance evaluation of ad hoc routing protocols using ns2 simulation. In *Mobile and Pervasive Computing (CoMPC)* (2008). **Cited** on page 48.

[240] SHAMOUN, S., COHEN, R., SARNE, D., AND MILLER, G. Combined ttl-based search algorithm. In *Ad Hoc Networking Workshop (MED-HOC-NET)* (June 2015), pp. 1–8. **Cited** on pages 39, 71 and 84.

[241] SHAMOUN, S., AND SARNE, D. Two-sided expanding ring search. In *Communication Systems and Networks (COMSNETS)* (Jan 2014), pp. 1–8. **Cited** on pages 13 and 77.

[242] SHAMOUN, S., SARNE, D., AND GOLDFEDER, S. Elastic ring search for ad hoc networks. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, I. Stojmenovic, Z. Cheng, and S. Guo, Eds., vol. 131 of *Lecture Notes of the Institute for Computer Sciences*. Springer International Publishing, 2014, pp. 564–575. **Cited** on page 77.

[243] SHANG, Y. Exponential random geometric graph process models for mobile wireless networks. In *Cyber-Enabled Distributed Computing and Knowledge Discovery* (oct. 2009), pp. 56–61. **Cited** on pages 57, 86 and 100.

[244] SHEN, H., LI, Z., AND CHEN, K. A scalable and mobility-resilient data search system for large-scale mobile wireless networks. *IEEE Transactions on Parallel and Distributed Systems 25*, 5 (May 2014), 1124–1134. **Cited** on pages 27 and 40.

[245] SHETE, P., AND AWALE, R. N. Rss-gossipaodv: Received signal strength based gossip flooding mechanism for aodv. In *IBM Collaborative Academia Research Exchange Conference (I-CARE)* (New York, NY, USA, 2014), I-CARE 2014, ACM, pp. 14:1–14:5. **Cited** on page 40.

[246] SHINTRE, A., AND SONDUR, S. Improved blocking expanding ring search (i-bers) protocol for energy efficient routing in manet. In *Recent Advances and Innovations in Engineering (ICRAIE)* (May 2014), pp. 1–6. **Cited** on pages 12 and 79.

[247] SINGH, C., VYAS, O., AND TIWARI, M. A survey of simulation in sensor networks. In *Computational Intelligence for Modelling Control Automation* (Dec 2008), pp. 867–872. **Cited** on page 47.

[248] Sjoberg, D. I. K., Dyba, T., and Jorgensen, M. The future of empirical methods in software engineering research. In *2007 Future of Software Engineering* (Washington, DC, USA, 2007), FOSE '07, IEEE Computer Society, pp. 358–378. **Cited** on page 16.

[249] Sobeih, A., Chen, W.-P., Hou, J. C., Kung, L.-C., Li, N., Lim, H., Tyan, H.-Y., and Zhang, H. J-sim: A simulation environment for wireless sensor networks. In *Annual Symposium on Simulation* (Washington, DC, USA, 2005), ANSS '05, IEEE Computer Society, pp. 175–187. **Cited** on page 47.

[250] Stathopoulos, T., Kapur, R., Estrin, D., Heidemann, J., and Zhang, L. Application-based collision avoidance in wireless sensor networks. In *International Conference on Local Computer Networks* (Nov 2004), pp. 506–514. **Cited** on page 31.

[251] Sudha, M. N., and Valarmathi, M. Collision control extended pattern medium access protocol in wireless sensor network. *Computers and Electrical Engineering 39*, 6 (2013), 1846–1853. Special Issue on Wireless Systems: Modeling, Monitoring, Transmission, Performance Evaluation and Optimization. **Cited** on page 31.

[252] Suman, S., and Balkrishan. Enhanced aodv for wireless networks. In *Advance Computing Conference (IACC)* (feb. 2010), pp. 246–249. **Cited** on pages 30 and 48.

[253] Sun, M.-T., and Lai, T.-H. Location aided broadcast in wireless ad hoc network systems. In *Wireless Communications and Networking Conference* (mar 2002), vol. 2, pp. 597–602. **Cited** on pages 34, 40, 71 and 78.

[254] Sy, D., Chen, R., and Bao, L. Odar: On-demand anonymous routing in ad hoc networks. In *Mobile Adhoc and Sensor Systems (MASS)* (Oct 2006), pp. 267–276. **Cited** on page 29.

[255] Tarkoma, S., Rothenberg, C., and Lagerspetz, E. Theory and practice of bloom filters for distributed systems. *IEEE Communications Surveys Tutorials 14*, 1 (First 2012), 131–155. **Cited** on pages 42, 45, 110, 111, 113 and 122.

[256] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks 8*, 2/3 (Mar. 2002), 153–167. **Cited** on pages 33, 40 and 68.

[257] Tudose, D., Voinescu, A., Petrareanu, M., Bucur, A., Loghin, D., Bostan, A., and Tapus, N. Home automation design using 6lowpan wireless sensor networks. In *Distributed Computing in Sensor Systems and Workshops (DCOSS)* (June 2011), pp. 1–6. **Cited** on page 2.

[258] Tyagi, S., and Chauhan, R. Performance analysis of proactive and reactive routing protocols for ad hoc networks. *International Journal of Computer Applications 1*, 14 (February 2010), 27–30. Published By Foundation of Computer Science. **Cited** on pages 36 and 50.

[259] Varaprasad, G. Power aware and signal strength based routing algorithm for mobile ad hoc networks. *International Conference on Communication Systems and Network Technologies 0* (2011), 131–134. **Cited** on page 51.

[260] Varga, A., and Hornig, R. An overview of the omnet++ simulation environment. In *Simulation Tools and Techniques for Communications, Networks and Systems* (ICST, Brussels, Belgium, Belgium, 2008), Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 60:1–60:10. **Cited** on page 47.

[261] Vural, S., Navaratnam, P., Wang, N., and Tafazolli, R. Asynchronous clustering of multihop wireless sensor networks. In *IEEE International Conference on Communications (ICC)* (June 2014), pp. 472–477. **Cited** on pages 34 and 40.

[262] Wan, Z., Ren, K., and Gu, M. Usor: An unobservable secure on-demand routing protocol for mobile ad hoc networks. *Wireless Communications 11*, 5 (May 2012), 1922–1932. **Cited** on page 29.

[263] Wang, H.-P., and Cui, L. An enhanced aodv for mobile ad hoc network. In *Machine Learning and Cybernetics* (july 2008), vol. 2, pp. 1135–1140. **Cited** on page 30.

[264] Wang, J., and Medidi, S. Density-first ad-hoc routing protocol for manet. In *Communication Systems Software and Middleware and Workshops* (jan. 2008), pp. 528–535. **Cited** on page 29.

[265] Wang, Y., Wu, H., and Tzeng, N.-F. Cross-layer protocol design and optimization for delay/fault-tolerant mobile sensor networks (dft-msn's). *IEEE Journal on Selected Areas in Communications 26*, 5 (june 2008), 809–819. **Cited** on page 37.

[266] Webster, F. *Theories of the Information Society*. International Library of Sociology. Taylor and Francis, 2014. **Cited** on page 1.

[267] Weingartner, E., vom Lehn, H., and Wehrle, K. A performance comparison of recent network simulators. In *IEEE International Conference on Communications* (June 2009), pp. 1–5. **Cited** on pages 14, 47 and 150.

[268] Weiser, M. Ubiquitous computing. In *Computer*, vol. 26. IEEE Computer Society Press, Los Alamitos, CA, USA, Oct. 1993, pp. 71–72. **Cited** on page 2.

[269] Wenqi, C., and Zhao, X. Multi-hop routing for wireless network in underground mines. In *Wearable Computing Systems (APWCS)* (april 2010), pp. 337–340. **Cited** on page 9.

[270] Williams, B., and Camp, T. Comparison of broadcasting techniques for mobile ad hoc networks. In *ACM International Symposium on Mobile Ad-hoc Networking and Computing* (New York, NY, USA, 2002), MobiHoc '02, ACM, pp. 194–205. **Cited** on page 137.

[271] Winstanley, C., Ramdhany, R., Taiani, F., Porter, B., and Miranda, H. Pampa in the wild: a real-life evaluation of a lightweight ad-hoc broadcasting family. *Journal of Internet Services and Applications 5*, 1 (2014). **Cited** on page 40.

[272] Wu, Z., Song, H., Jiang, S., and Xu, X. Ant-based energy aware disjoint multipath routing algorithm in manets. *International Conference on Multimedia and Ubiquitous Engineering 0* (2007), 674–679. **Cited** on pages 39 and 77.

[273] Yang, B., and Garcia-Molina, H. Improving search in peer-to-peer networks. In *Distributed Computing Systems* (2002), pp. 5–14. **Cited** on page 34.

[274] Yang, C.-C., Chen, J.-Y., Mai, Y.-T., and Yu, C.-C. Load-based power saving in ieee 802.16j multi-hop relay networks. *Wireless Personal Communications: An International Journal 77*, 3 (Aug. 2014), 1885–1903. **Cited** on page 7.

[275] Yang, T., Xie, G., Duan, R., Sun, X., and Salamatian, K. Towards practical use of bloom filter based ip lookup in operational network. In *Network Operations and Management Symposium (NOMS)* (May 2014), pp. 1–4. **Cited** on page 112.

[276] Yassein, M. B., Al-Dubai, A., Khaoua, M. O., and Al-jarrah, O. M. New adaptive counter based broadcast using neighborhood information in manets. *International Parallel and Distributed Processing Symposium 0* (2009), 1–7. **Cited** on page 34.

[277] Yassein, M. B., Nimer, S. F., and Al-Dubai, A. Y. A new dynamic counter-based broadcasting scheme for mobile ad hoc networks. *Simulation Modelling Practice and Theory 19*, 1 (2011), 553–563. Modeling and Performance Analysis of Networking and Collaborative Systems. **Cited** on page 30.

[278] Yassin, M., and Rachid, E. A survey of positioning techniques and location based services in wireless networks. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES)* (Feb 2015), pp. 1–5. **Cited** on pages 9 and 71.

[279] Ye, F., Zhong, G., Lu, S., and Zhang, L. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *Wirel. Netw. 11*, 3 (May 2005), 285–298. **Cited** on pages 40 and 139.

[280] Yick, J., Mukherjee, B., and Ghosal, D. Wireless sensor network survey. *Computer Networks 52*, 12 (2008), 2292–2330. **Cited** on pages 4 and 25.

[281] Yihui, Z., Xirong, Q., Wendong, W., Xiangyang, G., and Jian, M. N3s-olsr: Node-status self-sensing optimized link-state routing protocols for manet. *International Conference on Communications and Mobile Computing 3* (2010), 288–292. **Cited** on pages 40 and 41.

[282] Yu, C. W. Computing subgraph probability of random geometric graphs with applications in quantitative analysis of ad hoc networks. *IEEE Journal on Selected Areas in Communications 27*, 7 (Sept. 2009), 1056–1065. **Cited** on pages 26 and 48.

[283] Yu, D., Hua, Q.-S., Wang, Y., Yu, J., and Lau, F. Efficient distributed multiple-message broadcasting in unstructured wireless networks. *INFOCOM, Proceedings IEEE* (April 2013), 2427–2435. **Cited** on page 34.

[284] Zadeh, L. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353. **Cited** on page 115.

[285] Zelkowitz, M. V. An update to experimental models for validating computer technology. *Journal of Systems and Software 82* (March 2009), 373–376. **Cited** on page 16.

[286] Zeng, X., Bagrodia, R., and Gerla, M. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Proceedings of the Twelfth Workshop on Parallel and Distributed Simulation* (Washington, DC, USA, 1998), PADS '98, IEEE Computer Society, pp. 154–161. **Cited** on page 47.

[287] Zhang, H., and Jiang, Z.-P. On reducing broadcast expenses in ad hoc route discovery. In *Distributed Computing Systems Workshops* (June 2005), pp. 946–952. **Cited** on page 78.

[288] Zhang, X., Kunz, T., Li, L., and Yang, O. An energy-efficient broadcast protocol in manets: Design and evaluation. *Annual Conference on Communication Networks and Services Research, 0* (2010), 199–206. **Cited** on page 68.

[289] Zhang, Y., and Fromherz, M. Constrained flooding: a robust and efficient routing framework for wireless sensor networks. In *Advanced Information Networking and Applications (AINA)* (April 2006), vol. 1, p. 6. **Cited** on pages 34 and 39.

[290] Zheng, C., Sicker, D., and Chen, L. Self-organized context-aware hybrid manets. In *Wireless On-demand Network Systems and Services (WONS)* (March 2013), pp. 128–130. **Cited** on page 40.

[291] Zhong, X., and zhong Xu, C. Online energy efficient packet scheduling with delay constraints in wireless networks. In *Conference on Computer Communications* (April 2008). **Cited** on page 32.

[292] Zhu, J., and Wang, X. Model and protocol for energy-efficient routing over mobile ad hoc networks. *IEEE Transactions on Mobile Computing 10*, 11 (Nov 2011), 1546–1557. **Cited** on page 5.

[293] Zimmermann, H. Osi reference model - the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications 28*, 4 (April 1980), 425–432. **Cited** on page 152.

[294] Zuo, J., Dong, C., Ng, S. X., Yang, L.-L., and Hanzo, L. Cross-layer aided energy-efficient routing design for ad hoc networks. *Communications Surveys Tutorials 17*, 3 (thirdquarter 2015), 1214–1238. **Cited** on page 37.