Alma Mater Studiorum - Università di Bologna

DEI - Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione

Dottorato di Ricerca in Ingegneria Elettronica, delle Telecomunicazioni e Tecnologie dell'Informazione

XXVIII Ciclo

Settore Concorsuale: 09/F2 - Telecomunicazioni

Settore Scientifico Disciplinare: ING-INF/03

IoT and Smart Cities: Modelling and Experimentation

Tesi di:

Andrea Stajkic

Coordinatore:

Chiar.mo Prof. Ing. **Alessandro Vanelli-Coralli**

Relatori:

Chiar.mo Prof. Ing. **Roberto Verdone**

Dott. Ing. **Chiara Buratti**

Esame anno finale 2016

"By seeking and blundering we learn."

Johann Wolfgang von Goethe

"An experiment is a question which science poses to Nature,
and a measurement is the recording of Nature's answer."

Max Planck

Table of Contents

Abstract

Internet of Things (IoT) is a recent paradigm that envisions a near future, in which the objects of everyday life will communicate with one another and with the users, becoming an integral part of the Internet. The application of the IoT paradigm to an urban context is of particular interest, as it responds to the need to adopt ICT solutions in the city management, thus realizing the *Smart City* concept.

Creating IoT and Smart City platforms poses many issues and challenges. Building suitable solutions that guarantee an interoperability of platform nodes and easy access, requires appropriate tools and approaches that allow to timely understand the effectiveness of solutions. This thesis investigates the above mentioned issues through two methodological approaches: *mathematical modelling* and *experimentation*. On one hand, a mathematical model for multi-hop networks based on semi-Markov chains is presented, allowing to properly capture the behaviour of each node in the network while accounting for the dependencies among all links. On the other hand, a methodology for *spatial downscaling* of testbeds is proposed, implemented, and then exploited for experimental performance evaluation of proprietary but also standardised protocol solutions, considering *smart lighting* and *smart building* scenarios. The proposed downscaling procedure allows to create an indoor well-accessible testbed, such that experimentation conditions and performance on this testbed closely match the typical operating conditions and performance where the final solutions are expected to be deployed.

Preface

Building reliable Internet of Things (IoT) and Smart City solutions and services requires adequate tools and approaches that will allow to understand the effectiveness of solutions before commercial roll-out. The focus of this thesis is on two such approaches: *mathematical modelling* and *experimentation*. The two approaches will be discussed and presented in chapters that follow, along with the motivations supporting the studies performed.

The PhD was performed in a Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI) at the University of Bologna. Entire experimental work presented in this thesis was carried out within the framework of European FP7 project NEWCOM♯ (Network of Excellence in Wireless Communications). Under the umbrella of this project, I participated in the establishment of European Laboratory of Wireless Communications for the Future Internet (EuWIn), whose facilities available at University of Bologna were then used. Furthermore, a part of my experimental work, dealing with smart building systems, was performed within the framework of RIGERS project.

Problem Statement and Approach

The IoT concept aims at making the Internet even more pervasive. Deploying, for instance, surveillance cameras, monitoring sensors, actuators, displays, vehicles, and so on, will foster the development of a number of applications. These applications will use a potentially enormous amount and variety of data generated by such objects to provide new services to citizens, companies, and public administrations. In order to achieve this, an easy access and interaction with a wide variety of devices needs to be guaranteed. One of the most promising technologies that will allow for this goal to be fulfilled are Wireless Sensor Networks (WSNs). However, low sensing ranges result in dense networks and thus it becomes necessary to design and deploy efficient medium access control (MAC) and routing protocols and understand and determine which protocol suits best a given application. To this aim, it is essential to evaluate the protocol performance and compare it with several other solutions. This can be achieved through mathematical modelling of protocols, simulations and experimentation.

This thesis deals with the performance evaluation of:

- MAC protocols through *mathematical modelling* (in Chapter 2);
- Routing protocols through *experimentation* (in Chapter 3);

Recently, gaps between existing facilities and various desired properties and needs for suitable facilities for IoT experimentation were identified. Among these properties, the realism of experimentation environment plays a crucial role. Increased realism implies matching the experimentation conditions as close as possible to the typical operating conditions where the final solutions are expected to be deployed. In this

way, design flaws or imperfections can be detected in earlier phase and evened out, thus reducing the cost of roll out and maturation time. A part of this thesis is dedicated to responding to these needs by proposing and implementing a methodology for *spatial downscaling* of testbeds, i.e. a procedure to identify a subset of nodes of the flexible and large indoor testbed, which will constitute the *downscaled* testbed used to reproduce the real world network deployment.

Structure and Contribution of the Thesis

As previously stated, this thesis approaches the problem of IoT and Smart City paradigm implementation from two different perspectives, elaborated in details in the following chapters.

The aim of this thesis is to provide some general guidelines for the design of systems optimised for specific application-dependent requirements, by considering both theoretical and practical aspects of IoT and Smart City concept implementation. This thesis also gives a deep insight on the performance of several current most promising technologies in the IoT and Smart City framework.

The Chapter 1 introduces the concept of IoT and Smart City, discussing basic concepts, issues and challenges. A more detailed description of reference applications considered in this thesis, i.e. smart lighting and smart building application is given as well. The Chapter 1 also gives an overview of technologies that are considered as possible enablers for the implementation of IoT paradigm. A detailed description of IEEE 802.15.4 standard, as well as of the two solutions that can be implemented on top of it and are considered in this thesis, Zigbee and Low power Wireless Personal Area Networks (6LoWPAN), is provided. Finally, Chapter 1 concludes with a

description of methodologies considered in this thesis. In particular, a mathematical model for MAC protocols for smart lighting systems is introduced, followed by the proposal of spatial downscaling methodology.

Chapter 2 explains in details a novel mathematical approach, combining node-level semi-Markov chains (SMCs) and network-level finite state transition diagram (FSTD), for modeling multi-hop Linear Wireless Sensor Networks (LWSNs) using contention-based MAC protocols. It allows the derivation of per-node and network level performance metrics, such as throughput and energy efficiency. A novelty of the proposed model stands in the fact that it considers the dependencies among all links in the network and analyses both node and network states.

Chapter 3 discusses the empirical approach to IoT and Smart City paradigm implementation. In this chapter the proposed methodology for spatial downscaling is verified and implemented. Once the implementation of the procedure is demonstrated on an example, it is used for the optimisation of various network protocols. In particular, several protocol solutions are proposed, implemented, evaluated and compared with some benchmark solutions. In the second part of the chapter, a smart building application is addressed. Different types of network architectures and protocols will be evaluated under various environmental conditions. Advantages and disadvantages of centralised and distributed solutions will be demonstrated and discussed in details.

The main contribution of this thesis stands in the approaches that were adopted and applied. In both, analytical and experimental phases, novel methodologies are proposed and applied to two reference applications of this thesis.

The work presented in this thesis has lead to publications which are listed in Publications section of the thesis.

Chapter 1

Internet of Things and Smart Cities

This Chapter introduces the topic of IoT and Smart City (SC), discussing the perspectives, challenges and opportunities behind a future Internet. Reference applications considered in this thesis are described in details. In particular, smart lighting and smart building scenarios are considered, as these two scenarios apart from being very broad and widely present in the world of IoT and SC were also the topics of interest for projects that I participated in during my PhD course (Newcom♯ and RIGERS). The two main approaches discussed in this thesis, i.e. mathematical modelling and experimentation are brought up in this chapter. Mathematical model presented in this thesis is put forward. Furthermore, novel empirical tools and methodologies developed and used in this thesis are anticipated and introduced. Finally, the main technologies available for the realisation of IoT and Smart City paradigm (IEEE 802.15.4, Zigbee, 6LoWPAN), with particular attention to MAC and routing protocols that are subjects of this thesis, are described.

1.1 Internet of Things

The IoT is an emergent paradigm evolving around the concept of *things* (objects, cars, etc.), equipped with radio devices and uniquely addressable. The notion of IoT has been recognised by industrial leaders and media as the next wave of innovation, pervading into our daily life [1–3]. The basic idea of this concept is the pervasive presence of a variety of things or objects around us such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc., which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [4].

IoT is a multidisciplinary domain that covers a large number of topics from purely technical issues (e.g., routing protocols, semantic queries), to a mix of technical and societal issues (security, privacy, usability), as well as social and business themes [5].

The IoT enables physical objects to see, hear, think and perform jobs by having them talk together, to share information and to coordinate decisions. The IoT transforms these objects from traditional to smart by exploiting underlying technologies such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, Internet protocols and applications [6]. IoT is the biggest promise of the technology today.

Advances in microelectronics, microelectromechanical systems, and wireless communications allowed for the miniaturization of networked computers as well as sensors and actuators to connect to the physical world. Consequently, new fields that are considered an integral part of the IoT, as depicted in Figure 1.1, were opened:

- WSNs, that leverage low-power radios and multi-hop communication to cover

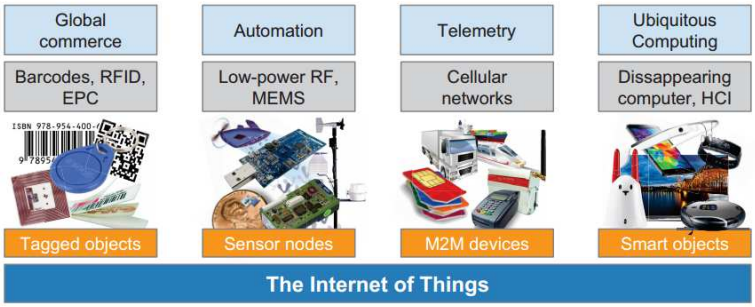

Figure 1.1: The concept of Internet of Things.

large areas with small, inexpensive, autonomous sensor nodes. They enable real-time sensor readings of physical phenomena, e.g., for battlefield surveillance, environmental monitoring, or smart cities.

- Machine-to-machine (M2M), usually uses cellular networks to connect stationary sensors and mobile objects, such as cargo or car fleets, to a central IT system. Besides cellular networks, there are new long-range radio technologies that target machine-to-machine (M2M), for instance LoRA [7], Sigfox [8], and the white space spectrum special interest group Weightless.

- Smart Objects, that are everyday objects endowed with processing and communication capabilities together with sensors and/or actuators. Through the connection with digital services, these objects become smart and can provide human-computer interaction that is woven into our everyday lives.

- Tagged Objects, that are objects equipped low cost tags able to provide identification to RF readers.

All four fields can already be found in the real world. However, most of them form so-called silo applications. These closed vertical systems only fulfill a special task and are hard to integrate with systems from other application domains.

The crucial leap toward a literal Internet of Things was made by adopting the Internet Protocol (IP) as the narrow waist to interconnect physical objects. In 2003, the IoT pioneers Adam Dunkels and Zach Shelby independently showed that native IP support is feasible for the resource-constrained devices used in WSNs and smart objects. With the increasing interest in low-power networks, the IETF chartered a working group4 in 2006 to standardize an adaptation layer for transmitting IP packets over IEEE 802.15.4 [9], the most common low-power radio standard at the time. The resulting IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) specifications are based on the Internet Protocol version 6 (IPv6), which has a modular design, and hence it is better suited for adaptation than its predecessor Internet Internet Protocol version 4 (IPv4).

Both, IEEE 802.15.4 and 6LoWPAN will be discussed in details in the rest of this section, being underlying technologies for the implementation of IoT and Smart City paradigm and solutions widely considered and discussed in this thesis.

It is important to understand the needs and requirements of IoT, in order to be able to properly design and customize the system on all layers, including MAC and routing protocols that are topics of interest in this thesis. To this aim, in the rest of this section, an overview of IoT challenges and applications will be given.

1.1.1 Applications

This section provides an overview of major applications of IoT paradigm. The main application fields considered, and their brief description are given in the following:

- **Design of smart cities.** The IoT can help the design of smart cities e.g., monitoring air quality, discovering emergency routes, efficient street lighting, watering gardens etc.

- **Smart metering and monitoring** The IoT design for smart metering and monitoring will help to get accurate automated meter reading invoice delivery to the customers. The IoT can also be used to design such scheme for wind turbine maintenance and remote monitoring, as well as gas, water and environmental metering and monitoring.

- **Design of smart homes.** The IoT can facilitate the design of smart homes e.g., energy consumption management, interaction with appliances, detecting emergencies, home safety and security etc.

- **Prediction of natural disasters.** The combination of sensors and their autonomous coordination can help to predict the occurrence of land-slides or other natural disasters and to take appropriate actions in advance.

- **Industry applications.** The IoT can find applications in industry e.g., managing a fleet of cars for an organization. The IoT helps to monitor their environmental performance and process the data to determine and pick the one that needs maintenance.

- **Water scarcity monitoring.** The IoT can help to detect the water scarcity at different places. The networks of sensors, might not only monitor long term water interventions such as catchment area management, but may even be used to alert if an upstream event, such as the accidental release of sewage into the stream, might have dangerous implications.

- **Medical applications.** The IoT can also find applications in medical sector for saving lives or improving the quality of life e.g., monitoring health parameters, monitoring activities, support for independent living, monitoring medicines intake etc.

- **Precision agriculture.** A network of different sensors can sense data, perform data processing and inform the farmer through communication infrastructure e.g., SMS about the portion of land that needs particular attention. This may include smart packaging of seeds, fertilizer and pest control mechanisms that respond to specific local conditions and indicate actions. Intelligent farming system will help agronomists to have better understanding of the plant growth models and to have efficient farming practice by having the knowledge of land conditions and climate variability. This will significantly increase the agricultural productivity by avoiding the inappropriate farming conditions.

- **Intelligent transport system design** The Intelligent transportation system will provide efficient transportation control and management using advanced technology of sensors, information and network. The intelligent transportation can have many interesting features such as non-stop electronic highway toll, mobile emergency command and scheduling, transportation law enforcement,

vehicle rules violation monitoring, reducing environmental pollution, anti-theft system, avoiding traffic jams, reporting traffic incidents, smart beaconing, minimizing arrival delays etc.

- **Smart security.** The IoT can also find applications in the field of security and surveillance e.g., area surveillance, tracking of people and assets, infrastructure and equipment maintenance, alarming etc.

1.1.2 Issues and challenges

The IoT can change the shape of the Internet and can offer enormous economic benefits but it also faces many issues and challenges. Some of them are briefly described below [10], [11].

- **A Low Power Communication Stack.** The majority of objects are not able to draw power from the mains, and are battery charged. This means that finding enough energy to power processing and communication is a major challenge. Whilst we are ready to recharge our mobile phones on a daily basis, changing batteries in millions of objects is impractical. Any stack must therefore exhibit a low average power consumption.

- **A Highly Reliable Communication Stack.** Although the Internet is a best-effort transport medium, protocols incorporate error detection, retransmissions and flow control. These techniques are applied at various protocol layers concurrently, which leads to a reliable end-to-end experience, albeit in a rather inefficient way. For the IoT to merge seamlessly into the Internet, it is

necessary to offer the same reliability we are used to on the Internet with the additional requirement that is the highest possible efficiency.

- **An Internet-Enabled Communication Stack.** Enabling another dialect of the Internet has profound implications on the protocol design. The Internet is exhibiting emergent behavior today because communication is bidirectional; it is hence of utmost importance to ensure that communication from objects but also towards objects is facilitated. Furthermore, the explosion of the Internet can arguably be attributed to the ability of any machine around the world to talk to any other machine, all this facilitated by one universal language, IP; it is hence of paramount importance that the IoT is IP enabled. This in turn calls for standardized communication solutions that will be described in details in the following and evaluated in this thesis.

1.2 Smart City

Cities nowadays face complex challenges to meet objectives regarding socio-economic development and quality of life. The concept of smart cities is a response to these challenges. Smart cities and green technology has become one of the most important and promising items in world agenda in preparing for better future [12–15].

The concept of smart cities has attracted considerable attention in the context of urban development policies. Although there is not yet a formal and widely accepted definition of Smart City, the final aim is to make a better use of the public resources, increasing the quality of the services offered to the citizens, while reducing the operational costs of the public administrations. This objective can be pursued by the

deployment of an urban IoT, i.e., a communication infrastructure that provides unified, simple, and economical access to a plethora of public services, thus unleashing potential synergies and increasing transparency to the citizens. An urban IoT, indeed, may bring a number of benefits in the management and optimization of traditional public services, such as transport and parking, lighting, surveillance and maintenance of public areas, preservation of cultural heritage, garbage collection, etc [16].

1.2.1 Applications

A brief overview of Smart City application fields is given in the following.

- **Wireless City.** Base stations or access points, originating from different technologies, will cover the entire city, providing many functions of urban management and service systems for the public, business, foreign visitors, tourists and government agencies. These functions include mobile wireless video surveillance, mobile video conferencing, mobile dispatching emergency response, and emergency telecommunications.

- **Smart Home.** Sensor devices, including radio frequency identification devices, infrared sensors, global positioning system, laser scanners and so on, can be combined with the Internet to form the Internet of Things. Then all the items around us in everyday life can be taken as terminals to be brought into the network, achieving the centralized and remote control. For example, the realization of smart home can be convenient to achieve the intelligent control of lighting and electrical appliances, as well as receive the intelligent notification of home alarm messages.

- **Smart transportation.** According to their needs and traffic situation, every city can take advantage of sensor network and other technical means to change the traditional transport system, and establish the smart traffic management system, including adaptive traffic signal (automatic control of traffic lights according to flow time) control system, urban traffic control system and so on. At this point, the smart traffic management system can achieve the integration of urban planning, construction, management and operations, and provide comprehensive support for other subsystems of smart urban system.

- **Smart Public Service and Construction of Social Management.** In daily life, in order to respond to citizens' complaints, requests for assistance, personal management of social affairs and other aspects, we can establish a social service system, which can cover the intelligent management of the entire city and market operation. Based on this, we can provide a service platform for urban comprehensive planning, emergency response, community management, and turn the government into a one-stop service system. In this case, the government can collect and analyze real-time data in urban areas, providing more rapid and agile service to the public.

- **Smart Medical Treatment.** The Internet of Things, having the great potential to be applied into smart medical treatment, can help hospitals to achieve the smart medical care and intelligent management of medical materials, and support the digital collection, processing, storage, transmission and sharing of internal medical information, as well as equipment, drug, and personnel management. Besides, it can also meet the needs of intelligent equipment supplies

management and supervision of public health, solving so many issues, as for example the weak support of health care platform, the overall low level of medical services and the medical safety hazards.

- **Green City.** With the current technological platforms, we can achieve not only the networking, interoperability and mutual control of various devices and systems, but also the collection, transmission, storage, display and control of audio, video and alarm information for the environmental purposes.

- **Smart Tourism.** Smart tourism should be based on the existing tourism related information and infrastructure, taking advantage of digital information and the Internet of Things to achieve the establishment of a set of solutions, which can fulfill the management and tourism-related tasks, such as tourism online services, management of customer relation and operational area, development of domestic and overseas market, intelligent management system of monitoring, collection of information as well as forecast of tourism development.

1.2.2 Issues and challenges

With the increasing needs of urban management, construction and operation in reasonable planning of urban space and function layout, incident detection, emergency response and public information services, the construction of smart city is facing great difficulties, including the following [13]:

- **Large scale space-time information and efficient services.** Spatial information of smart city comes from a wide variety of sensors, controllers and computing terminals, and is maintained by computers and storage nodes of

different departments, so how to manage and coordinate the equipment with various structures and wide-area distribution is a great challenge for constructing service platform. On the other hand, information on smart city contains not only a large amount of structured data, such as temperature, geographical coordinates and so on, but also a lot of unstructured data, such as pictures, audio and video files. And whether we can store and manage the huge amount of data effectively will directly affect the performance of information services. Finally, smart city is related to intelligent analysis of urban information, decision support, public affairs and many other applications. Besides, a large amount of real-time tasks also need to respond to user requests quickly.

- **Heterogeneous sensor data.** The important basis of developing a smart city appears to be the Internet of Things. But as the demands in sensor platform, observation mechanisms, processes, location information and technical requirements are different, how to build models describing sensor information, including location attribute, observation object, time and status is a difficult technical problem.

- **Intelligent analysis and decision support.** With diverse sources spatio-temporal data should update in real-time. So how to create a unified understanding of data semantics, and extract new knowledge based on specific cycle data and realtime data poses a technical difficulty in establishing a knowledge base of smart city.

- **Sharing policy mechanism and legal protection.** As smart city involves

many sectors and industries, we need to break trade barriers to achieve information sharing and information exchange between many different departments, such as traffic, public security, etc.

1.3 Reference Applications

1.3.1 Smart Lighting System

Equipping infrastructure with sensors that can transmit and receive data creates opportunities for cutting costs and increase of environmental stainability. The existing city infrastructure provides a basis for a large plethora of services, once proper wireless technologies and devices are mounted on top of it. Sensors for monitoring purposes (air quality, traffic, emergency situations etc.) can be deployed on roads, bridges, in tunnels or on top of lamp posts for example (see Fig. 1.2), as well as for smart lighting system purposes [17–19], which is a reference application in this thesis. In fact, in order to support the 20-20-20 directive, the optimization of the street lighting efficiency is an important feature. In particular, this service can optimize the street lamp intensity according to the time of the day, the weather condition, and the presence of people.

It can be noticed that for all of the aforementioned applications it is common that nodes are deployed along a line and therefore form a particular, linear network topology [20, 21]. Apart from Smart City scenario, the linear type of networks can appear in the broader IoT framework, like in the area of gas/water/oil pipeline control, or border and river environmental monitoring. This calls for dedicated solutions in terms of system design, including MAC and routing protocols, that will be thoroughly

studied throughout this thesis, both through mathematical modelling (Chapter 2) and experimentation (Chapter 3).

Depending on the specific application implemented, data could be generated by just one source in the line, or by all nodes deployed. In both cases the generated data should be transmitted to a given destination node, typically located at the end of the street, acting as a gateway toward the Internet (i.e., 3G gateway).

Depending on the traffic considered, two scenarios are defined and considered in the rest of the thesis:

- Linear Wireless Network (LWN), where only the last node in the line, i.e., the farthest from the destination, generates packets and sends them to the next hop, in order to reach the sink. All other nodes just forward the received packets.

- LWSN, where all nodes in the network can generate a packet. Relay nodes are allowed to send their data only after they receive a packet from the previous node in the line, acting as token. Relay nodes generate and append their payloads to the payload of the last packet received from the previous node in the line. As a result, packet size increases hop by hop.

1.3.2 Smart Building

An important structural element of Smart Cities are buildings - residential or commercial - in which people spend a significant amount of time in their daily lives. Making these buildings smart with IoT technologies will not only improve the quality of life and convenience of citizens in indoor spaces, but also contribute towards more sustainable cities through more efficient utilization of scarce resources such as energy,

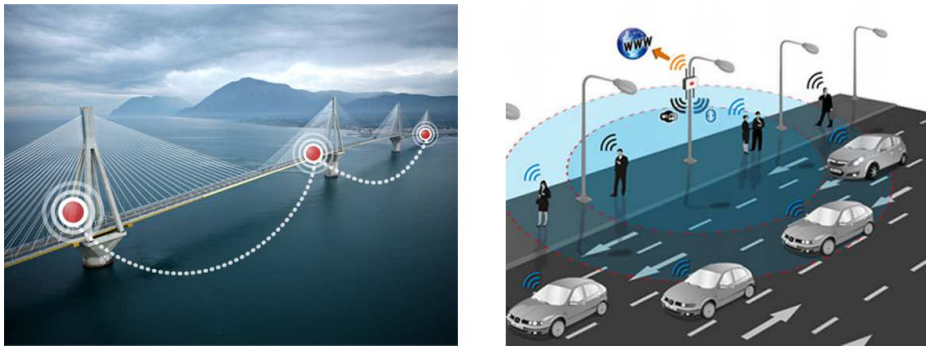

Figure 1.2: Linear wireless network examples.

gas and water. Buildings consume about 42% of electricity which is more than any other assets globally. Smart buildings provide services to building tenants and allow to reduce energy consumption costs. The energy-efficient operation of smart buildings would bring a major impact on many organizations. This can be achieved through the utilization of wireless technologies, as WSNs. Compared with a wired monitoring system, the installation of WSN is more economic and more flexible, and requires less modifications to the exiting buildings' structure, which leads to the realization of metering and control for smart buildings [22]. Wireless sensors and actuators, including electrical meters, thermal, pressure, and illumination sensors and actuators, monitor and control electricity consumptions and different environmental factors related to electricity consumptions in buildings and typically report measurements to a central entity that is a building concentrator. The availability of these monitored data brings opportunities to improve resource management in buildings, through, for example, strategic electricity pricing schemes and intelligent scheduling of electricity consuming activities. As shown in Figure 1.3, in general a smart building consists of:

- Sensors - monitoring and notifying in case of changes;
- Actuators - performing a physical action;
- Controllers - controlling units and devices based on programmed rules set by user;
- Central unit enabling programming of units in the system;
- Interface - the user communication with the system;
- Network - allows communication between the units;
- Smart meter - offers two-way, near or real-time communication between customer and utility company.

In order to enable reliable system functioning and communication between aforementioned elements, different technologies can be deployed. These technologies will be introduced in the following, and evaluated through experimentation in Chapter 3.

1.4 Methodological Approach: Modelling and Experimentation

As previously stated, the focus of this thesis is on the study of MAC and routing protocols deployed in IoT and Smart City applications. To this aim, different approaches can be adopted. The work described in this thesis focuses on two approaches: analytical and empirical. The latter refers to the mathematical model for MAC protocol for smart lighting system that will be introduced in the following and discussed in details

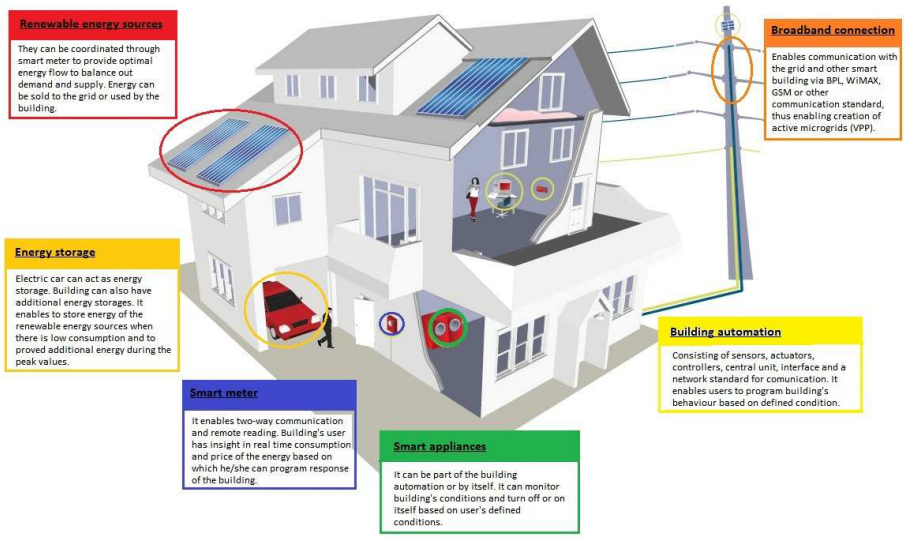

Figure 1.3: Smart building.

in Chapter 2. The former approach refers to the experimental work presented in this thesis in Chapter 3, that focuses on the implementation and performance evaluation of network protocols for smart lighting and smart building systems.

1.4.1 Mathematical Modelling

For what concerns mathematical modelling, a new approach to analytically derive per-node and network performance metrics, by considering both, single node and network behavior is proposed and discussed in details in Chapter 2. The model is based on SMC. SMC are a generalisation of Markov processes that extend the specification of the process by including a state holding-time, that is the time that passes before moving to the next state. Discrete time Markov chain (MC) have a unit

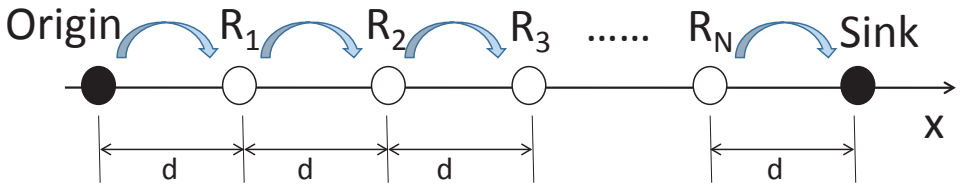

Figure 1.4: Reference scenario.

state holding time [23] and therefore are not suitable for properly describing CSMA-based protocols, where the state permanence time depends on the current status of the node (sensing, transmission, idle, etc.). Summing up, the main contributions of the proposed model presented in this thesis are:

- The proposal of a mathematical model for multi-hop wireless networks considering the dependencies among all links in the network;
- The proposal of a novel approach considering both, node and network status;
- Evaluation of the per-node and network performance metrics.

Multi-hop wireless networks using contention-based MAC protocols, have been studied through state transition diagrams and Markov Chains (MCs) many times in the past (see Section 2.1). One of the problems of previous approaches, is that they model independently the state of the different nodes. On the opposite, the state of each node in a multi-hop communication path, depends on the rest of the network, a fact that can not be handled properly by modelling the individual nodes independently. Using only state transition diagrams at the node-level therefore, brings to approximations. In particular, in a multi-hop LWSN each node is impacted by the other nodes in the line depending on whether: i) the latter have packets in the queue, which is a fact

related to the overall network state; ii) they can "hear" each other through the carrier sensing mechanism (if used); iii) mutual interference generates packet losses.

In this thesis, a novel approach based on the combination of a node-level Semi-Markov Chain (SMC), which models the behaviour of a generic node assuming a given network state, and a network-level Finite State Transition Diagram (FSTD), which accounts for the state of all nodes in the network, is proposed. The methodology allows to precisely consider the mutual dependance among the different links in the multi-hop path, in terms of queue status, interference levels, and ability to sense transmissions made by others.

The approach can be applied to different MAC protocols. As reference cases Slotted Aloha and a type of Carrier Sensing Multiple Access, specifically devised for LWSNs (L-CSMA) [24], scheme are considered in this thesis. Depending on the MAC protocol considered, the model complexity increases when N gets larger. Though unfortunately there is no simple formulation of the approach for a generic value of N, it is shown that derivation of the FSTD for a network with $N + 1$ routers is simple, al long as it is found for the case of N routers. A recursive approach can provide the FSTD for any N.

1.4.2 Experimentation

According to FIRE+ (Future Internet Research and Experimentation, one of the components of the European Commission framework Horizon 2020), experimentally-driven research and innovation is a key mechanism towards advancement in Internet technology[1]. This is particularly true for the many applications of the future Internet

[1]http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/85-ict-11-2014.html

of Things (IoT), like smart public lighting, waste management, handling of parking systems, etc.

However, there are a number of practical issues that arise from the need to test new IoT protocols/applications over *real world deployments*. First, this often requires significant investments in terms of human and financial resources, in order to properly plan, deploy, use and maintain the experimental platforms; small or medium enterprises sometimes cannot afford such an investment. Moreover, real world testbeds are often not fully controllable; this makes the analysis of experimental results very difficult. Finally, protocol optimization requires running separate experiments consecutively under fixed conditions (e.g., in terms of radio environment); this is possible only if the full testbed context can be recorded and replicated, that is not achievable in real world deployments.

Running tests under controlled conditions, as well as reducing the time for experiments, is therefore fundamental, especially for IoT applications and protocols. This can be achieved by using a *controllable testbed*, deployed in an indoor and controlled environment, able to allow reproduction of different real world conditions, and to replicate the experimental context as many times as needed[2].

The work presented in this thesis replies to these needs, by proposing a methodology to identify a subset of nodes of the flexible and large indoor testbed, which will constitute the *downscaled* testbed used to reproduce the real world deployment. The proposed methodology will be described in the following and implemented in Chapter 4.

[2]In the thesis the word "reproduce" is used to describe the action of associating to a real world deployment a downscaled testbed providing similar performance results. The word "replicate" is used for the duplication of the same experimental context on the downscaled testbed, for the purpose of comparison among different protocol/application configurations.

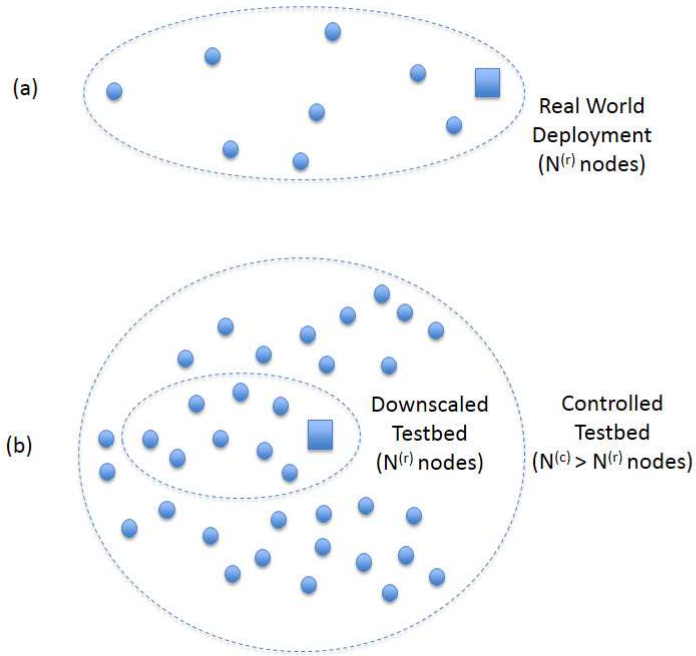

Figure 1.5: Downscaling description.

Figure 1.5 clarifies the concept of downscaling: a real world deployment (a) with $N^{(r)}$ nodes and a coordinator can be reproduced on the downscaled testbed (b) having the same number $N^{(r)}$ of nodes; the latter is part of the controllable testbed that, for the sake of flexibility, is made of a larger number $N^{(c)}$ of nodes, and is compact in space. The selection of the subset of nodes must be such that the performance of the protocols/solutions implemented and tested on the downscaled testbed, will be similar to those that would be measured in the real world deployment. The problem of selecting the best possible subset of nodes is formulated as a 0-1 Linear Program, and solved using a state-of-the-art Mixed Integer Programming solver.

Once the procedure is suitably formalised, it will be implemented and verified in Chapter 4, using two outdoor applications where IEEE 802.15.4-compliant sensor

nodes are deployed over lamp posts in two different locations: i) a parking where we deployed $N^{(r)} = 11$ nodes and ii) a district of a small town near Bologna, where deployed $N^{(r)} = 25$ nodes are deployed. As controllable testbed, a platform developed within the EuWIn laboratory, established at the University of Bologna and described in details in Chapter 4, will be used.

1.5 Enabling Technologies

In this section, an overview of the main technologies used for the implementation of IoT and SC paradigm is given. Emphasis is given to the IEEE 802.15.4 standard, which has been widely accepted as the de facto standard for wireless sensor networks, and in particular to MAC sub-layer as defined by the standard, since in the rest of this thesis it will be shown how MAC protocols can be analytically modelled (in Chapter 2). Moreover, routing protocols that are typically used in the IoT and SC framework will be described in details. The experimental evaluation of these benchmarks, as well as some proprietary routing protocol solutions will be presented in Chapter 3.

1.5.1 IEEE 802.15.4 Standard

Institute of Electrical and Electronics Engineers (IEEE) 802.15.4-2003 (Low Rate wireless personal area network (WPAN)) deals with low data rate but very long battery life (months or even years) and very low complexity. The IEEE 802.15.4 Working Group[3] focuses on the standardisation of the bottom two layers of Open Systems Interconnection (OSI) protocol stack, physical (Layer 1) and data-link (Layer 2) layer. The higher layers are normally specified by industrial consortia such as the ZigBee Alliance[4]. The first edition of the 802.15.4 standard was released in May 2003. Several standardised and proprietary networks (or mesh) layer protocols run over 802.15.4-based networks, including IEEE 802.15.5, ZigBee, 6LoWPAN, WirelessHART, and ISA100.11a.

[3]See also the IEEE 802.15.4 web site: http://www.ieee802.org/15/pub/TG4.html
[4]See also the ZigBee Alliance web site: http://www.zigbee.org

IEEE 802.15.4 wireless technology is a short-range communication system intended to provide applications with relaxed throughput and latency requirements in WPANs. The main field of application of this technology is the implementation of WSNs, that are key underlying technologies in the IoT and SC frameworks.

In the following some technical details related to the physical (PHY) layer and the MAC sublayer as defined in the standard, are reported.

The IEEE 802.15.4 PHY layer operates in three different unlicensed bands (and with different modalities) according to the geographical area where the system is deployed. However, direct sequence spread spectrum (DS-SS) is mandatory everywhere to reduce the interference level in shared unlicensed bands.

PHY layer provides the interface with the physical medium. It is in charge of radio transceiver activation and deactivation, energy detection, link quality, clear channel assessment, channel selection, and transmission and reception of the message packets. Moreover, it is responsible for establishment of the radio frequency (RF) link between two devices, bit modulation and demodulation, synchronization between the transmitter and the receiver, and, finally, for packet level synchronization.

IEEE 802.15.4 specifies a total of 27 half-duplex channels across the three frequency bands, whose channelisation is depicted in Fig. 1.6 and is organized as follows:

- 868 MHz band, used in the European area, implements a cosine-shaped binary phase shift keying (BPSK) modulation format, with DS-SS at chip-rate 300 $\frac{\text{kchip}}{\text{s}}$ (a pseudo-random sequence of 15 chips transmitted in a 25 µs symbol period). Only a single channel with data rate 20 $\frac{\text{kbit}}{\text{s}}$ is available and, with a required minimum −92 dBm RF sensitivity, the ideal transmission range (i.e., without considering wave reflection, diffraction and scattering) is approximatively 1 km;

- 915 MHz band, ranging between 902 and 928 MHz and used in the North American and Pacific area, implements a raised-cosine-shaped BPSK modulation format, with DS-SS at chip-rate 600 $\frac{\text{kchip}}{\text{s}}$ (a pseudo-random sequence of 15 chips is transmitted in a 50 µs symbol period). Ten channels with rate 50 $\frac{\text{kbit}}{\text{s}}$ are available and, with a required minimum −92 dBm RF sensitivity, the ideal transmission range is approximatively 1 km;

- 2.4 GHz industrial scientific medical (ISM) band, which extends from 2400 to 2483.5 MHz and is used worldwide, implements a half-sine-shaped Offset Quadrature Shift Keying (O-QPSK) modulation format, with DS-SS at 2 $\frac{\text{Mchip}}{\text{s}}$ (a pseudo-random sequence of 32 chips is transmitted in a 16 µs symbol period). Sixteen channels with data rate 250 $\frac{\text{kbit}}{\text{s}}$ are available and, with minimum −85 dBm RF sensitivity required, the ideal transmission range is approximatively 220 m.

The ideal transmission range is computed considering that (although any legally acceptable power is permitted) IEEE 802.15.4-compliant devices should be capable of transmitting at −3 dBm. Since the 2.4 GHz band is shared with many other services, the other two available bands can be used as an alternative.

Power consumption is a primary concern, so, to achieve long battery life the energy must be taken continuously at an extremely low rate, or in small amounts at a low power duty cycle: this means that IEEE 802.15.4-compliant devices are active only during a short time. The standard allows some devices to operate with both the transmitter and the receiver inactive for over 99% of time. So, the instantaneous link data rates supported (i.e., 20 $\frac{\text{kbit}}{\text{s}}$, 40 $\frac{\text{kbit}}{\text{s}}$, and 250 $\frac{\text{kbit}}{\text{s}}$) are high with respect to the data throughput in order to minimize device duty cycle.

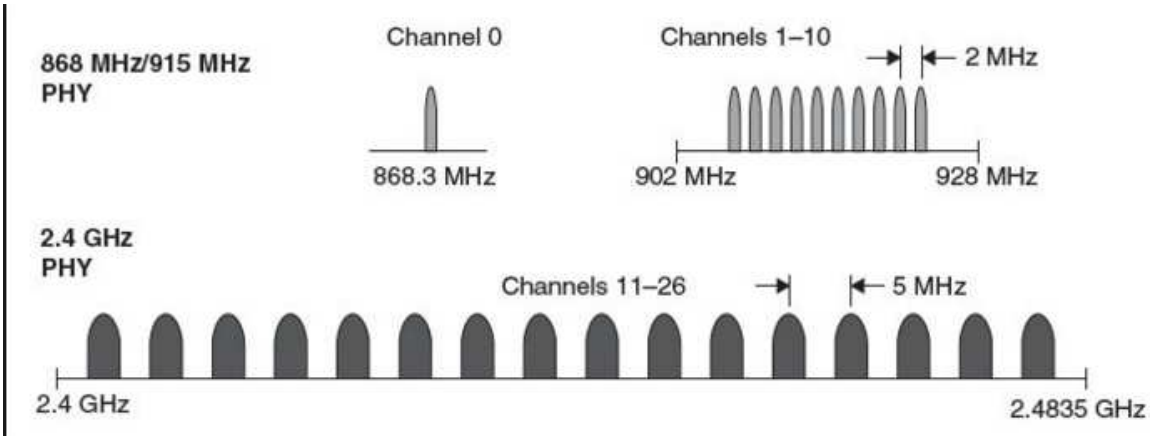

Figure 1.6: Spectrum allocation chart and channelisation for WPAN applications in IEEE 802.15.4 standard.

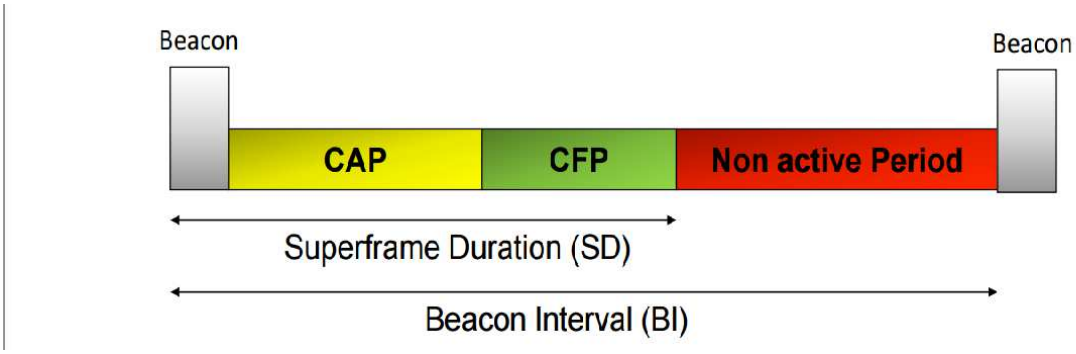

Figure 1.7: IEEE 802.15.4 SF structure.

IEEE 802.15.4 defines two different operational MAC modes, namely *beacon-enabled* and *non beacon-enabled*, which correspond to two different channel access mechanisms.

In the non beacon-enabled mode nodes use an unslotted carrier-sense multiple access with collision avoidance (CSMA/CA) protocol to access the channel and transmit their packets [9].

In the beacon-enabled mode [9], instead, the access to the channel is managed

through a superframe (SF), starting with a packet, called *beacon*, transmitted by WPAN network coordinator (NC). The SF may contain an inactive part, allowing nodes to go in sleep mode, whereas the active part is divided into two parts: the Contention Access Period (CAP) and the Contention Free Period (CFP), composed by Guaranteed Time Slots (GTSs), that can be allocated by the NC to specific nodes (see Figure 1.7). The use of GTSs is optional.

In CAP, CSMA/CA channel access algorithm is employed. CSMA/CA Algorithm (Fig. 1.8) is implemented using units of time called Beacon Period (BP) with a duration of 320 μ sec. For each transmission attempt, every node in the network should maintain three variables, namely Backoff Counter (NB), Contention Window (CW), and Backoff Exponent (BE). NB is the number of times the algorithm is required to backoff while attempting the current transmission. It is initialized to 0 and it can assume a maximum value of $\mathrm{NB_{max}}$. CW is the contention window length, whose initial value is equal to 2. It defines the number of BPs where no activity on the channel should be detected before a new transmission can start. BE is the backoff exponent related to the number of BPs a node shall wait before attempting again to sense the channel. It varies between $\mathrm{BE_{min}}$ (initial value) and $\mathrm{BE_{max}}$. Once CAP starts, a node with a packet to transmit will first delay any activity (backoff state) for a number of BPs randomly drawn in the range $\left[0, 2^{\mathrm{BE}-1}\right]$. After this delay, channel sensing is performed for one BP. If the channel is sensed as busy, CW is reset to 2, while NB and BE are increased by 1, ensuring that $\mathrm{BE} \leq \mathrm{BE_{max}}$. If $\mathrm{NB} \leq \mathrm{NB_{max}}$ the node should return in backoff state and wait for another random interval of time. If the channel is assessed as idle, CW is decremented by 1 instead. If CW$>$ 0, the node waits for another BP and then it sounds the channel again, acting as described

before (busy or idle state). The algorithm ends either with the data transmission for CW = 0 or with a failure, when NB $\geq$ NB_{max}, meaning that the node did not succeed in accessing the channel in a maximum number of attempts.

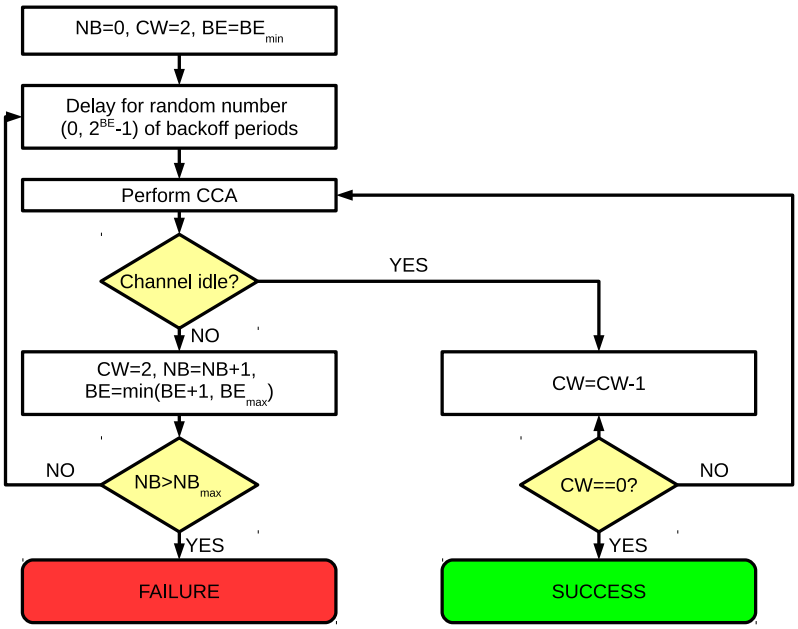

Figure 1.8: IEEE 802.15.4 CSMA/CA algorithm flowchart.

To overcome the limited transmission range, multihop self-organizing network topologies are required. These can be realized taking into account that IEEE 802.15.4 defines two type of devices: the full function device (FFD) and the reduced function device (RFD). The FFD contains the complete set of MAC services and can operate as either a NC or as a simple network device. The RFD contains a reduced set of MAC services and can operate only as a network device.

Two basic topologies are allowed, but not completely described by the standard since definition of higher layers functionalities are out of the scope of IEEE 802.15.4: the star topology, formed around an FFD acting as a NC, which is the only node

allowed to form links with more than one device, and the peer-to-peer topology, where each device is able to form multiple direct links to other devices so that redundant paths are available.

1.5.2 ZigBee Higher Levels Overview

The purpose of the ZigBee Alliance is to univocally describe the ZigBee protocol standard in such a way that interoperability is guaranteed also among devices produced by different companies, provided that each device implements the ZigBee protocol stack.

The ZigBee stack architecture is composed of a set of blocks called layers. Each layer performs a specific set of services for the layer above.

The ZigBee stack architecture is depicted in Fig. 1.9. Given the IEEE 802.15.4 specifications on PHY and MAC layer, the ZigBee Alliance provides the network layer and the framework for the application layer.

The responsibilities of the ZigBee network layer include: mechanisms to join and leave a network, network security, routing, path discovery, one-hop neighbours discovery, neighbour information storage.

The ZigBee application layer consists of the application support sublayer, the application framework, the ZigBee device objects and the manufacturer-defined application objects. The responsibilities of the application support sublayer include: maintaining tables for binding (defined as the ability to match two devices together based on their services and their needs), and forwarding messages between bound devices. The responsibilities of the ZigBee device objects include: defining the role of the device within the network (e.g., WPAN coordinator or end device), initiating

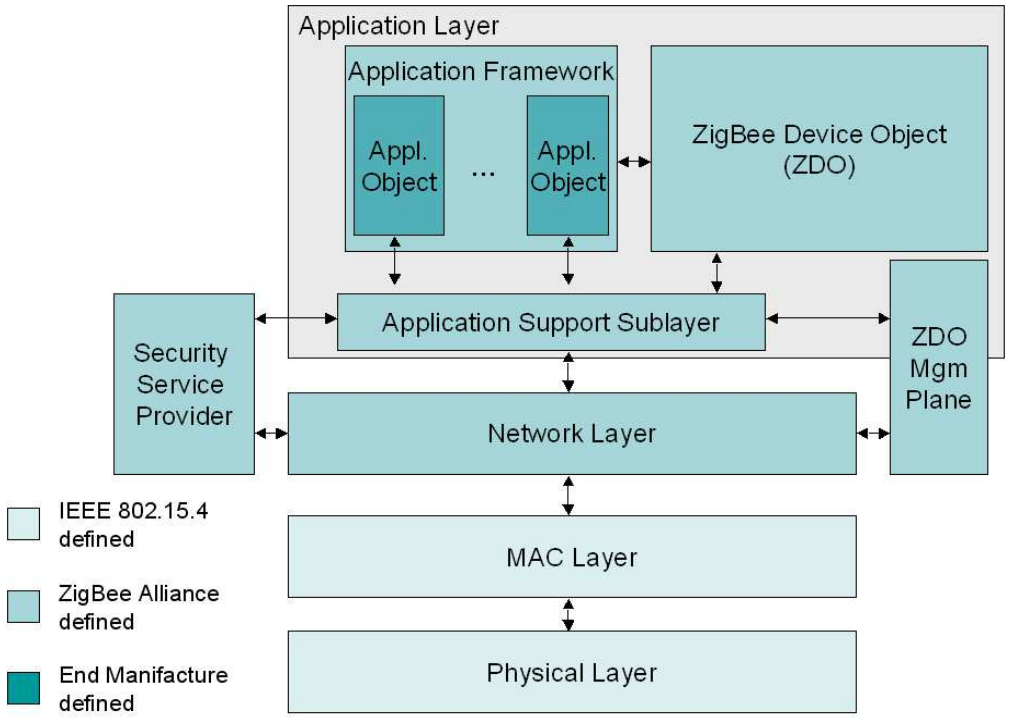

Figure 1.9: A detailed overview of ZigBee stack architecture.

and/or responding to binding requests, establishing secure relationships between network devices, discovering devices in the network, and determining which application services they provide.

ZigBee supports three types of devices: ZigBee routers (ZR), able to perform all tasks described in the IEEE 802.15.4, including forwarding of data; ZigBee Coordinator (ZC), a particular ZR, in charge also of forming the network ZigBee End Devices (ZED) that do not have routing capabilities. Both ZR and ZC correspond to FFD in the IEEE 802.15.4, while ZED corresponds to RFD. In the Zigbee mesh topology generally more than one path, connecting a couple of devices, could be present and, in case of link failures or changing in the environment, paths could be updated on demand. The ZC is responsible for starting the network and collaborates with the ZRs for discovering and maintaining the routes. A ZED cannot perform route discovery and data forwarding.

In order to let nodes compute the link costs to be used during the route discovery process, each node in the network periodically sends Link Status packets in broadcast at one hop. Each node receiving the Link Status packet computes the link cost, being a function of the link quality indicator of the received packet.

1.5.2.1 Zigbee Routing: AODV and Many-to-one

In this section, an overview on Zigbee routing protocols that were evaluated through experimentation (in Chapter 3) is given.

AODV The default routing algorithm of ZigBee is based on Ad hoc on demand distance vector (AODV), which is a pure on-demand route acquisition algorithm. According to Zigbee, a link, l_i, connecting a pair of nodes, is characterized by a given

link cost, given by [25]:

$$C_{l_i} = min\{7, floor(P_{l_i}^{-4})\}, \tag{1.5.1}$$

where $floor(.)$ is a function mapping the real number to the largest previous integer, and P_{l_i} is the probability of packet delivery on the link l_i. According to ZigBee specifications P_{l_i} may be computed through the link quality indicator (LQI) measure, however no specific methods are defined in the standard for the computation of this metric and its implementation is left to the designer. As an example, in the case of the Freescale MC1322x platform, the cost function is a function of the LQI, where the latter is proportional to the received signal strength indicator (RSSI). Therefore, in Zigbee, link costs depend on the success probability computed at link level.

The protocol selects the path, P, connecting a pair of nodes, characterized by the smallest total cost, $C(P)$, given by:

$$C(P) = \sum_{i=1}^{L} C_{l_i}, \tag{1.5.2}$$

being L the number of hops in the path. In case there exists more than one path connecting the same pair of nodes with the same total cost, the path characterized by the lowest value of L, will be selected.

The control packets exchanged in Zigbee networks to find the optimum path are described in the following. The source node broadcasts a route request (RREQ) packet to its neighbors and then intermediate nodes receiving the RREQ rebroadcast it to their neighbors until the RREQ reaches the destination node. During the process of rebroadcasting the RREQ, intermediate nodes record in their route discovery tables the address of the sender from which the first copy of the broadcast packet was received, and the corresponding link cost, computed according to eq. (1.5.1), where

P_{l_i} depends on the LQI with which the RREQ was received. The comparison among path costs of packets with the same RREQ allows choosing the best path and discarding anything worse. Once the destination node receives the RREQ, it responds by unicasting a route reply (RREP) packet back through the selected path. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their routing tables which point to the node from which the RREP came. These forward route entries indicate that the link between the node and the destination is established. Finally, when the RREP reaches the originator, it can send data packet [26].

Many-to-one MTO routing allows to establish a tree topology, rooted at the coordinator. In order to form and maintain the tree, the coordinator periodically sends a MTO Route Request (MTORR) packet in broadcast. Each node receiving a MTO-RR, before retransmitting it, reads the accumulated path cost (i.e., the sum of the costs of the links of the reverse path toward the coordinator) included in the packet, and selects the next hop toward the coordinator. If a node receives several MTO-RRs from different nodes, it elects as a next hop the node characterised by the minimum total path cost to the coordinator. At the end of this MTO-RR transmission, all nodes in the network are aware of the next hop to be used in order to transmit their data to the coordinator. However, if the coordinator wants to know the path to reach a specific node in the network (or a set of nodes, through multi-casting), MTO routing should be combined with Source Routing (SR). After the MTO-RR transmission, once a node has a data packet to be sent to the coordinator, it first sends a Route Record (RREC) packet through the selected path. Each node in the path receiving the RREC packet adds in the relay list field its own address and forwards the new

RREC packet toward the coordinator. The coordinator analyses the RREC packet and stores that information in the Source Route Table. Each time the coordinator has to send a packet to a node, it reads the relay list from this table and sends the packet through the selected path. Even though MTO-RRs are periodically sent by the coordinator and are not generated on-demand (which would make the protocol pro-active), ZigBee saves the re-active feature through the use of Ad hoc AODV protocol [27], when needed. In particular, in case of link failure, AODV is used for discovering a new path toward the destination.

1.5.3 IETF 6LOWPAN Higher Levels Overview

As already well known, in the Internet a packet passes through many different interconnected networks on its way from source to destination. Thus, considering the link layer technology of each traversed network, there is a need for IP-over-X specification to define how to transport IP packets. In many cases, to map the services required by the IP layer on the services provided by the lower layer (i.e, the link layer), the IP-over-X specification can introduce a (sub)layer of its own, often called adaptation layer [28]. Following the same strategy, in the process of shaping the IoT world, the IETF 6LoWPAN working group has started in 2007 to work on specifications for transmitting IPv6 over IEEE 802.15.4 networks [29]. Typically, Low power WPANs are characterized by: small packet size, support for addresses with different lengths, low bandwidth, star and mesh topologies, battery supplied devices, low cost, large number of devices, unknown node positions, high unreliability, and long idle periods. Given the aforementioned features, it is clear that the adoption of IPv6 on top of a Low power WPAN is not straightforward, but poses strong requirements for the

optimization of this adaptation layer. For instance, due to the IPv6 default minimum maximum transmission unit (MTU) size (i.e., 1280 bytes), a no-fragmented IPv6 packet would be too large to fit in an IEEE 802.15.4 frame. Moreover, the overhead due to the 40 bytes long IPv6 header would waste the scarce bandwidth available at the PHY layer. For these reasons, the 6LoWPAN working group has devoted huge efforts for defining an effective adaptation layer in [30]. Further issues encompass the auto-configuration of IPv6 addresses, the compliance with the recommendation on supporting link-layer subnet broadcast in shared networks [31], the reduction of routing and management overhead, the adoption of lightweight application protocols (or novel data encoding techniques), and the support for security mechanisms (i.e., confidentiality and integrity protection, device bootstrapping, key establishment and management). To manage IPv6 packets, allowing link-layer forwarding and fragmentation, 6LoWPAN uses an intermediate adaptation layer between IPv6 and IEEE 802.15.4 MAC levels [30]. Specifically, all 6LoWPAN encapsulated datagrams (that should be transported over IEEE 802.15.4 MAC) are prefixed by a stack of headers, each one identified by a type field. In particular, the header types can be logically grouped in four categories, depending on the function they play in the 6LoWPAN adaptation strategy, as summarized below:

- A no-6LoWPAN Header is used for specifying that the received packet is not compliant to 6LoWPAN specifications and therefore it has to be discarded (this allows the coexistence with other no-6LoWPAN nodes in the same network).

- A Dispatch Header is used to compress an IPv6 header or to manage link-layer multicast/broadcast.

- A Mesh Addressing Header allows IEEE 802.15.4 frames to be forwarded at link-layer, turning single-hop WSNs in multi-hop ones.

- A Fragmentation Header is used when a datagram does not fit within a single IEEE 802.15.4 frame.

It is worth to note that each header may be present or not, depending on the needs. Moreover, headers should appear in a precise order. In the most typical setting, the nodes of the network are connected through multi-hop paths to a small set of root devices, which are usually responsible for data collection and coordination duties. For each of them, a Destination Oriented Directed Acyclic Graph (DODAG) is created by accounting for link costs, node attributes/status information, and an objective function (OF), which maps the optimization requirements of the target scenario. The topology is set-up based on a Rank metric, which encodes the distance of each node with respect to its reference root, as specified by the Objective Function. Regardless the way it is computed, the Rank should monotonically decrease along the DODAG and towards the destination, in accordance to the gradient-based approach. An example of DODAG that can be formed can be seen in Figure 1.10.

The Multipoint-to-Point (MP2P) is the dominant traffic in many Low power and Lossy Network (LLN) applications. It is usually routed towards nodes with some application relevance, such as the LLN gateway to the Internet or to the core of private IP networks. In general, these destinations are the DODAG roots and they act mainly as data collection points for distributed monitoring applications. On the contrary, Point-to-Multipoint (P2MP) data streams can be used for actuation purposes, by means of messages sent from DODAG roots to destination nodes. Finally,

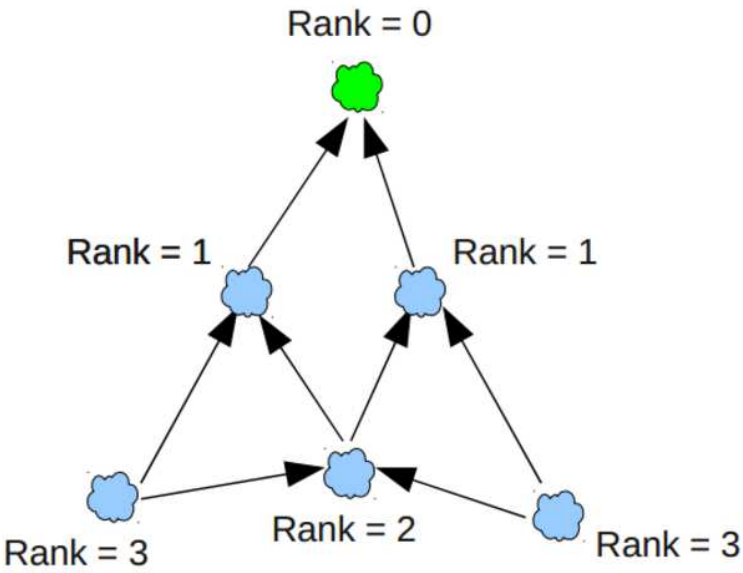

Figure 1.10: An example of RPL DODAG.

Point-to-Point (P2P) traffic is necessary to allow communications between two devices belonging to the same LLN, e.g., a sensor and an actuator. In this case, a packet will flow from the source towards the common ancestor of those two communicating devices; then, downward towards the destination. As a consequence, routing protocol has to discover both upward routes (i.e., from nodes to DODAG roots) in order to enable MP2P and P2P flows, and downward routes (i.e., from DODAG roots to nodes) to support P2MP and P2P traffic. The simplest topology is made by a single DODAG with just one root, e.g., a WSN monitoring a small size area. A more complex scenario is composed of multiple uncoordinated DODAGs with independent roots, that is, the LLN is split in several partitions depending on the needs of the application context. A more sophisticated and flexible configuration could contain a single DODAG with a virtual root that coordinates several LLN root nodes. The main advantage in this case, with respect to the previous one, is the absence of limitations on the parent set selection, given that all nodes belong to the same virtual DODAG, although a stronger coordination is needed among the root nodes. Depending on the application requirements, it is also possible to combine the three examples presented so far, in more complex topologies. The formation of all previously described kind of topologies relies on the routing protocol information dissemination mechanism, which enables a minimal configuration in the nodes and allows them to operate mostly autonomously.

1.5.3.1 6LoWPAN Routing

Routing issues are very challenging for 6LoWPAN, given the low-power and lossy radio-links, the battery supplied nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions should take into account

the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms. An effective solution is being developed by the IETF Routing Over Low power and Lossy (ROLL) networks working group. Recently, it has proposed the leading IPv6 Routing Protocol for Lowpower and Lossy Networks (LLNs), RPL, based on a gradient-based approach [32], [33].

RPL can support a wide variety of different link layers, including those that are constrained, potentially lossy, or typically utilized in conjunction with host or router devices with very limited resources, as in building/home automation, industrial environments, and urban applications. It is able to quickly build up network routes, to distribute routing knowledge among nodes, and to adapt the topology in a very efficient way. For these characteristics, it is suitable also for smart grid communications.

Network devices running the RPL protocol are connected in such a way that no cycles are present. For this purpose, a DODAG, which is routed at a single node, is built. The RPL specification calls this specific node a DODAG root. The graph is constructed by the use of an OF which defines how the routing metric is computed. In other words, the OF specifies how routing constraints and other functions are taken into account during topology construction. In RPL, the OF translates key metrics and constraints into a Rank, which models the node distance from a DODAG root, in order to optimize the network topology in a very flexible way. The Rank may be equal to a simple hop-count distance, may be calculated as a function of the routing metric or it may be calculated with respect to other constraints. Furthermore, the OF allows the selection of a DODAG to join and the identification of a number of peers in that DODAG as parents.

The RPL specification defines four types of control messages for topology maintenance and information exchange. The first one is called DAG Information Object (DIO) and is the main source of routing control information. It may store information like the current Rank of a node, the current RPL Instance, the IPv6 address of the root, etc. The second one is called a Destination Advertisement Object (DAO). It is used to propagate destination information upwards along the DODAG. The third one is named DAG Information Solicitation (DIS) and makes it possible for a node to require DIO messages from a reachable neighbor. The fourth type is a DAO-ACK and is sent by a DAO recipient in response to a DAO message.

In order to implement network formation and management operations, all nodes execute several operations: they send and receive DIOs, containing information about the Rank, the OF, the IDs, and so on. DIO messages are multicasted (periodically) by each node to create the DODAG, thus establishing paths towards the roots. Nodes receiving DIO compute their own Rank, based on the information included in the received DIOs; they join a DODAG and select a set of possible parents in that DODAG among all nodes in the neighborhood; they select the preferred parent among the possible ones. A node receiving a DIO message uses its information to join a new DODAG, or to maintain an existing one, according to the Objective Functions and the Ranks of their neighbors. It can also detect possible routing loops.

Another important aspect of routing the protocols design is the maintenance of the topology. Since most of devices in LLNs are typically battery powered, it is crucial to limit the amount of sent control messages over the network. Many routing protocols broadcast control packets at a fixed time interval which causes energy to be wasted when the network is in a stable condition. Thus, RPL adapts the sending rate

of DIO messages by extending the Trickle algorithm [34]. In a network with stable links the control messages will be rare whereas an environment in which the topology changes frequently will cause RPL to send control information more often.

Chapter 2

Modeling Multi-Hop Networks Using Contention-Based MAC Through Semi-Markov Chains

In this chapter a novel approach to assess the performance of contention-based MAC protocols in Linear Wireless Sensor Networks (LWSNs) is presented. Nodes are considered to be deployed over a straight line and multi-hop transmission is used. The approach is based on the combination of Semi Markov Chains to model the node behaviour, and FSTD to account for all possible interdependencies among the nodes, at the network level. Performance of the MAC protocols is measured in terms of saturated throughput and energy efficiency. The methodology allows accurate performance assessment of different MAC protocols for LWSNs, under a unique framework, a fact that is essential for a fair comparison. The approach is applied to Slotted Aloha and a CSMA-based protocol, as reference cases. It is shown that, as long as the model is used for a network with N sensor nodes, it is easily extended to the case of $N + 1$ nodes.

The rest of the chapter is organised as follows. The next section shows the state of the art in terms of related works; with respect to the latter, the novel contributions

of this work are emphasised, followed by the description of the reference scenario and the model assumptions. The fundamental concept behind the proposed approach, introducing the mathematical model and the formalisation of the node-level SMC and the network-level FSTD is discussed. Then the application of the proposed model to two case studies, Slotted Aloha and L-CSMA, respectively is described. Finally, numerical results and conclusions are reported.

2.1 Related Work

2.1.1 Single-Hop Networks

Theoretical studies of MAC protocols for wireless networks through MCs have become a common practice since the seminal Bianchi's paper [35]. In [36], Authors proposed a MC to evaluate the throughput of 802.15.4 networks with unsaturated downlink and uplink traffic. They considered slotted CSMA with Collision Avoidance (CSMA/CA) in acknowledgement (ACK) mode. However, their analytical model did not match the simulation results very well. The work presented in [37] suggests a MC model for the slotted CSMA/CA, under saturated and unsaturated periodic traffic conditions in a beacon-enabled network, both when ACKs are and are not present. However, the model relies on the assumption that the probability to start sensing the channel is independent among the nodes. Authors in [38] use a more accurate MC model, but they do not consider the dependence of transmission probabilities among nodes when calculating the sensing probabilities. Similarly, [39] proposes a generalized analytical model of the slotted CSMA/CA mechanism of the beacon-enabled IEEE 802.15.4 standard with retry limits, still considering independent probabilities that nodes start

sensing. In [40], a mathematical model for the beacon-enabled mode of the IEEE 802.15.4 MAC protocol is provided, which more precisely follows the MAC procedure defined by the standard taking into account the superframe structure as well; the model however does not scale to multi-hop networks.

Recently, a novel SMC model for the single-hop IEEE 802.11 has been proposed, to mitigate the complexity of wireless local area network (WLAN) performance analysis [23]. The Authors show that the proposed model achieves accurate results with less complexity and computation time with respect to Bianchi's model. In [41], the Authors propose an advanced SMC model that calculates more accurately the network parameters of single-hop WLANs. No application of the approach to multi-hop networks has been proposed.

2.1.2 Multi-Hop Networks

In [42] the Authors study different performance indicators of IEEE 802.15.4 over multi-hop networks. They assume that the probability that a node starts sensing is independent from node to node and therefore, the stationary probability that it attempts sensing the carrier in a randomly chosen time slot is constant for all nodes. Similarly, the works presented in [43], [44] and [45] investigate the performance of the IEEE 802.11 MAC in multi-hop networks, always assuming independent events among different nodes.

The analysis presented in this thesis goes beyond these assumptions, since we take into consideration that the traffic sensed by nodes is different from node to node. With a similar scope, in [46] the Authors model an IEEE 802.15.4 multi-hop network, focusing on the analysis of MAC and wireless channel interaction in the

presence of fading. They take into account the fact that the queue of each node depends also on the traffic generated by the previous node; however they assume that it is independent from link to link. In contrast to that, in the analysis presented here, the queue of the generic node is modelled taking into account the dependence on the traffic generated by all nodes in the network. In [47], the Authors propose a model in which the entire network is modeled using a single MC. However, in order to reduce complexity and improve the applicability of the model, they assume that the steady state probabilities of many states are equal and therefore do not calculate the probability of all possible states.

Summing up, almost all previously cited works have one feature in common: the analysis they propose captures only the node-level behavior. On the other hand, the few works dealing with network-level analyses, do not account properly for all possible interdependencies among nodes.

This work is inspired by previous paper [24]. Nevertheless, there is a significant number of aspects differentiating this work from [24]. The main differences with respect to [24] are: i) a node-level analysis based on SMCs, which could be simply extended to model other contention-based MAC protocols is introduced; ii) the network-level model in order to simplify the analysis is modified; iii) the per-node performance is derived; iv) the energy consumption evaluation is considered.

2.2 Reference Scenario and Assumptions

A Linear Wireless Sensor Network (LWSN), where nodes are deployed over a straight line is considered. Multi-hop transmission is used [20,21]. The network is composed of an origin, N routers (R_n, $n = 1, \ldots, N$) and a sink. The origin generates periodically

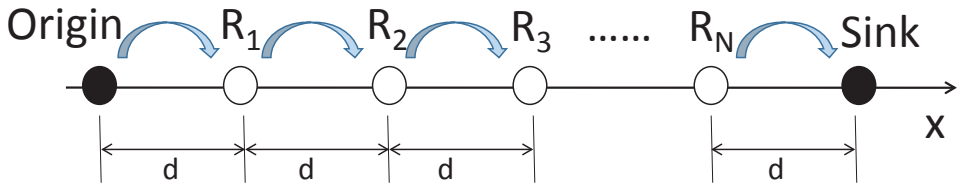

Figure 2.1: Reference scenario.

a data packet, which is transmitted to the next node in the line. Each router forwards it, appending its own payload, till the sink is reached after $N+1$ hops (see Fig. 2.1). LWSNs find application in different fields: e.g. in the area of gas/water/oil pipeline control [48,49], monitoring of rivers [50], and public city lighting systems where sensors are deployed on the lamp posts [51], which is a reference application considered in this thesis. The case where the sensor nodes are equally spaced, separated by d meters, is considered [1].

It is assumed that the network uses a contention-based Medium Access Control (MAC) protocol. The main goal of this chapter is to study network performance in terms of saturation throughput and energy efficiency, as well as to provide per-node metrics.

In the considered scenario, the multi-hop network topology is a natural consequence of the linear distribution of the $N+1$ sensor nodes. Therefore, it is assumed that the routing algorithm has assigned to each node, R_n, the next one in the line, R_{n+1}, as the destination of its own transmission. In the following for the sake of simplicity of notations, the origin is denoted as R_0 and the sink as R_{N+1}. It is assumed

[1]This assumption is introduced for the sake of simplicity, without lack of generality. Removing it, would make the analysis more complex from the notational viewpoint. Moreover, we notice that the assumption is coherent with the application fields of LWSNs that we refer to.

that nodes are time synchronized: time is divided into slots of duration T_{slot} [s].

As in many other papers dealing with contention-based MAC protocols, we are interested in estimating the saturation throughput (this brings to the situation where interference generated in the LWSN is maximum); assuming that nodes work in half duplex mode, when the origin generates a sample and a packet is transmitted, it will refrain from transmitting in the immediate next slot (when the first router might be willing to transmit on its turn), generating a new packet in the immediately following one. Acknowledgements are not used and retransmissions are not considered for the sake of energy efficiency (a goal of most sensor network applications). Under such assumption, the maximum throughput at the sink node is given by one received packet every two slots.

The sink collects the measures sampled by all sensor nodes in the line. For the sake of energy and protocol efficiency, it is assumed that nodes are allowed to send their samples only when they receive a packet from the previous node in the line. In particular, when a router receives a packet, it appends its own payload composed of P bits, before sending it to the next node. The sink will receive data only if successful transmission will happen on all links; thus, we trade-off energy efficiency with received throughput.

Z_n denotes the size of the packet transmitted by router R_n, equal to $Z_n = H + (n+1) \cdot P$, being H the number of bits of the packet header. By denoting as R_b the bit rate, the packet transmission time at the source is equal to $T_\mathrm{p}^{(0)} = Z_0/R_\mathrm{b}$, while the transmission time at the generic node, n, with $n \in \{1, \ldots, N\}$ is: $T_\mathrm{p}^{(n)} = T_\mathrm{p}^{(n-1)} + P/R_\mathrm{b}$. Time slot duration T_{slot} is equal to or larger than $T_\mathrm{p}^{(N)}$ [s].

The dependance of the received power, P_R, on the transmit power, P_T is modelled

according to the following expression:

$$P_{\mathrm{R}} = P_{\mathrm{T}} \cdot k \cdot x^{-\beta} \cdot f \tag{2.2.1}$$

, where k is a constant, x is the distance between the nodes considered, β is the attenuation coefficient, and finally, f is the short-term (fast) random fading component, whose square root is modeled through Rayleigh distribution. The following assumptions are also made:

1. all links are symmetric and power control is used such that the network is fully connected (i.e., each node receives a power equal to the receiver sensitivity, $P_{\mathrm{R}_{min}}$, from the previous hop)[2];

2. a node can detect transmission from a node during the carrier sensing phase (in the following for the sake of simplicity we will say that it "hears" the transmitter) if $P_{\mathrm{R}} \geq P_{\mathrm{S}_{min}}$, being $P_{\mathrm{S}_{min}}$ the sensing threshold;

3. a packet is captured by a receiver if the ratio between the useful and the interfering power is larger than or equal to a capture threshold, α. Only the nearest interferer is accounted for[3] [24]: comparison with simulations demonstrates the negligible impact of this assumption;

4. channel coherence time is much larger than the average time needed for the transmission of one packet from the origin to the sink.

[2]Power control could be simply implemented by listening to the packet forwarded by the next hop in the chain and provided that the node includes the level of transmit power into the packet itself.

[3]Owing to the linearity of the topology, the nearest interferer is also the strongest one with high probability, even in the presence of fading.

In the following $h_{n,m}$ denotes the probability that node n can "hear" node m (and viceversa), given by:

$$h_{n,m} = \mathbb{P}\left\{P_{\mathrm{R}_n}^{(m)} \geq P_{\mathrm{S}_{min}}\right\} = \mathbb{P}\left\{P_{\mathrm{T}}^{(m)}\ k\ d_{n,m}^{-\beta}\ f_{m,n} \geq P_{\mathrm{S}_{min}}\right\}$$

where $P_{\mathrm{R}_n}^{(m)}$ is the power received by node n when node m is transmitting, $f_{m,n}$ is the fading sample between node m (the transmitter) and node n (the receiver) and $d_{n,m}$ is the distance between nodes n and m, which will be expressed in the following as $x_{n,m} \cdot d$, being $x_{n,m}$ an integer. By using eq. (2.2.1) and recalling that the random variable representing fading is negative exponentially distributed, we have

$$\begin{aligned} h_{n,m} &= \mathbb{P}\left\{\frac{f_{m,m+1}}{f_{m,n}} \leq \frac{P_{\mathrm{R}_{min}}}{P_{\mathrm{S}_{min}}}\ x_{n,m}^{-\beta}\right\} \\ &= \int_0^{+\infty}\int_0^{\gamma} e^{-(f_{m,n}+f_{m,m+1})}\ df_{m,n}\ df_{m,m+1} \end{aligned}$$

where $\gamma = \frac{P_{\mathrm{R}_{min}}}{P_{\mathrm{S}_{min}}} x_{n,m}^{-\beta}$. The latter results in $h(x_{n,m}) \triangleq h_{n,m} = \frac{\gamma}{1+\gamma}$. For what concerns the capture effect, we have:

$$c_{n,m} = \mathbb{P}\left\{\frac{C^{(n)}}{I^{(m)}} \geq \alpha\right\} = \mathbb{P}\left\{\frac{f_{m,n+1}}{f_{m,m+1}} \leq \frac{x_{n+1,m}^{\beta}}{\alpha}\right\} \tag{2.2.2}$$

obtained by denoting as $x_{n+1,m} \cdot d$ the distance between the interferer m and the useful receiver, $n+1$ and using the following constraints: $C^{(n)} = P_{\mathrm{R}_{n+1}}^{(n)} = P_{\mathrm{R}_{min}}$ and $I^{(m)} = P_{\mathrm{T}}^{(m)}\ k\ (x_{n+1,m} \cdot d)^{-\beta} f_{m,m+1} = P_{\mathrm{R}_{min}}\ x_{n+1,m}^{-\beta}\ \frac{f_{m,n+1}}{f_{m,m+1}}$. The latter results in $c(x_{m+1,m}) \triangleq c_{n,m} = \frac{\xi}{1+\xi}$, where $\xi = \frac{x_{n+1,m}^{\beta}}{\alpha}$.

2.3 The Proposed Approach

The behaviour of nodes in the multi-hop LWSN will be described through a node-level SMC. The state of a generic node n, from the radio transceiver viewpoint, can be one

of the following: Transmission (T), Listening (L), Idle (I) or Sensing (S). Each of them is characterised by a holding time, denoted as $H_i^{(n)}$ (with i equal to T, L, I or S), that is the amount of time that the node spends in that state. The probability of state i for node n in the SMC (which depends on $H_i^{(n)}$ and the state transition probabilities) is denoted as $\pi_i^{(n)}$. The per-node performance metrics can be computed if $\pi_i^{(n)}$ is known $\forall i$, $\forall n$.

In particular, the state of the generic node n in the SMC depends on the other nodes in the LWSN through: i) the probability of having a packet in the queue (which depends on the previous transmissions in the multi-hop path), $P_a^{(n)}$; and ii) the probability of sensing the channel as busy (if carrier sensing is used), $P_b^{(n)}$. These two probabilities depend on the overall network state; once they are known, we will derive the node states $\pi_i^{(n)}$.

The computation of $P_a^{(n)}$ and $P_b^{(n)}$ is based on a FSTD used to describe the generic state k of the entire network of $N+2$ nodes, whose probability is denoted as $\pi_k^{(net)}$.

The next Sub-Section formalises the node-level SMC for a generic node and MAC algorithm, and provides expressions for $\pi_i^{(n)}$, conditioned to given network states. The following Sub-Section introduces the network-level FSTD. In Sub-Section 2.3.3 the performance metrics is defined, combining node- and network-level considerations.

2.3.1 The Node-Level Semi-Markov Chain

All probabilities discussed in this Sub-Section are conditioned to the network state k; however, for the sake of readability, we will omit the indication of the condition on $\pi_k^{(net)}$.

Semi-Markov processes extend the specification of Markov processes by including

the state holding time, previously defined. In the case of a SMC, state probabilities depend on the stationary probabilities of the corresponding Embedded Markov Chain (EMC), which cannot include self loops (transition from a state to itself) [52]. The EMC is created, moving from the MC describing the node behaviour according to the MAC algorithm used. Therefore, in the following we start from the assumption that the node-level MC is known; we derive the stationary probabilities of the EMC, and then compute the state probabilities of the SMC.

$\Pi_i^{(n)}$ and $\Theta_i^{(n)}$ denote the probabilities of state i in the MC and EMC for node n, respectively. Obviously,

$$\Pi_i^{(n)} = \sum_{\forall j} \Pi_j^{(n)} P_{j,i} \tag{2.3.1}$$

where $P_{j,i}$ is the transition probability from state j to state i of the MC. For previously justified reasons, we first need to transform the MC to an EMC, having state transition probabilities $P_{i,i}^{(e)} = 0, \forall i$. The values of $P_{j,i}^{(e)}$ can be derived by setting: ${P_{j,i}}^{(e)} = 0$, for $j = i$, and ${P_{j,i}}^{(e)} = P_{j,i}/(1 - P_{i,i})$ for $j \neq i$. Consequently, the stationary probability of the state i of the EMC for node n is given by:

$$\Theta_i^{(n)} = \frac{\Pi_i^{(n)}/(1 - P_{i,i})}{\sum_{\forall j} \Pi_j/(1 - P_{j,j})}$$

Finally, we can pass to the SMC and derive the stationary state probabilities for node n, given by [52]:

$$\pi_i^{(n)} = \frac{\Theta_i^{(n)} \mathbb{E}\{H_i^{(n)}\}}{\sum_{\forall j} \Theta_j^{(n)} \mathbb{E}\{H_j^{(n)}\}} \tag{2.3.2}$$

where $\mathbb{E}\{H_i^{(n)}\}$ is the expected value of the holding time of state i for node n .

Let us recall that the transition probabilities of the MC $P_{j,i}$ in Eq. (3) depend on the packet arrival probability, $P_a^{(n)}$, and the busy channel probability, $P_b^{(n)}$, to be extracted from the network-level FSTD.

2.3.2 The Network-Level Finite State Transition Diagram

The generic network state is modelled through a $2 \cdot (N+1)$-dimensional stochastic binary process, composed of two sequences of bits: $\{\mathbf{L}, \mathbf{Q}\}$. $\mathbf{L} = \{L_0 \dots L_N\}$ is a $N+1$ bit sequence representing the status of each link in a given slot, connecting R_n to R_{n+1}, with $n \in \{0, \dots, N\}$; the $(N+1)$-dimensional sequence of bits, $\mathbf{Q} = \{Q_0 \dots Q_N\}$, represents the state of the queue of nodes in the LWSN during the same slot. In particular, $L_n = 1$ represents an active link, that is a transmission occurring on link n (from R_n to R_{n+1}) during the slot, and $L_n = 0$, otherwise. We set $Q_n = 1$ if node R_n has a packet in the queue, because i) a new one has been generated, ii) there is a packet in the queue that was not transmitted since the channel was detected as busy, or iii) there is a packet in the queue that is being transmitted during the current slot.

K denotes the number of possible states in which the network could be, and as $\boldsymbol{\pi}^{(net)}$ the K-dimensional network state probability vector whose elements are the values of $\pi_k^{(net)}$. $\mathbf{P}^{(net)}$ denotes the $K \times K$ matrix of the state transition probabilities.

To find the network state stationary probabilities, ${\pi_k}^{(net)}$, the following system should be solved:

$$\begin{cases} \sum_{\forall k} \pi_k^{(net)} = 1 \\ \boldsymbol{\pi}^{(net)} = \mathbf{P}^{(net)} \cdot \boldsymbol{\pi}^{(net)} \end{cases} \tag{2.3.3}$$

The probabilities $P_a^{(n)}$ and $P_b^{(n)}$ can now be derived, assuming the values of $\pi_k^{(net)}$ are known. The former is obtained from the FSTD, looking at the states where node n has a packet in the queue; therefore $P_a^{(n)} = 1$ if $Q_n = 1$ in $\pi_k^{(net)}$ and zero otherwise. Similarly, $P_b^{(n)} = 1$ if $Q_n = 1$ and $L_n = 0$ in $\pi_k^{(net)}$, and zero otherwise; in fact, the channel for node n is busy if the node has a packet in the queue ($Q_n = 1$), and it does not transmit at the current slot ($L_n = 0$). As is clear from the previous statements,

$P_a^{(n)}$ and $P_b^{(n)}$ depend implicitly on the state k of the network. Therefore, performance metrics should account for all possible network states k (from 1 to K).

2.3.3 Performance Metrics

The final objective of the model is to allow derivation of the per-node and network performance metrics, through combination of the node- and network-level descriptions. For the sake of formal clarity, in this Sub-Section we make explicit the indication of the condition on $\pi_k^{(net)}$ when needed.

2.3.3.1 Per-Node Metrics

The following per-node performance metrics is considered: i) the probability that the generic node n transmits a packet, denoted as $P_T^{(n)}$; ii) the per-node throughput, that is the number of bits per second correctly received at each hop, denoted as $\Sigma^{(n)}$; iii) the average energy consumption per node, $E^{(n)}$.

i) The probability $P_T^{(n)}$ that node n transmits a packet is equivalent to $\pi_T^{(n)}$ and will be derived easily later for the two MAC algorithms considered in this paper.

ii) Being $P_a^{(n)}$ the probability that node n has a packet in the queue, that is, it received the packet from node $n-1$ correctly, we can express the per-node throughput as

$$\Sigma^{(n)} = \begin{cases} \frac{(n+1)P}{T_{slot}} \sum_{\forall k} P_a^{(n+1)}|_{\pi_k{}^{(net)}} \cdot \pi_k^{(net)} \; [byte/s] & \text{for } n = \{0, 1..N-1\} \\ \frac{(n+1)P}{T_{slot}} \sum_{\forall k} P_a^{(n)}|_{\pi_k{}^{(net)}} \cdot \pi_k^{(net)} \cdot c_{n,m} \; [byte/s] & \text{for } n = N \end{cases} \tag{2.3.4}$$

where $c_{n,m} = 1$ if $L_m = 0$ in $\pi_k^{(net)}$; otherwise (if $L_m \neq 0$), it is given by Eq. (2.2.2). This means that if there is another node (node m) transmitting together with the

last router (node N), the packet of node N is captured with probability $c_{n,m}$; while if no other nodes are transmitting together with router N, the throughput just depends on the probability that the latter has a packet in the queue, $P_a^{(N)}$.

iii) The average energy spent by node n is given by:

$$E^{(n)} = \sum_{\forall i} E_i{}^{(n)} = \sum_{\forall k} \sum_{\forall i} E_i{}^{(n)}|_{\pi_k^{(net)}} \; \pi_k^{(net)} = \sum_{\forall k} \sum_{\forall i} \mathbb{E}\{H_i^{(n)}\} \; \pi_i^{(n)}|_{\pi_k^{(net)}} \; \pi_k^{(net)} \; W_i^{(n)} \tag{2.3.5}$$

where $E_i{}^{(n)}$ and $W_i^{(n)}$ are the energy and power consumed in state i, respectively.

2.3.3.2 Network Metrics

From the viewpoint of the overall LWSN, we are interested into i) the network throughput (which we derive in saturated conditions), and ii) the energy efficiency of the network.

i) The network throughput $\Sigma^{(net)}$ is computed as the throughput of the last router R_N, that is $\Sigma^{(net)} = \Sigma^{(N)}$. The probability to have a packet in the queue, as well as the transmission probability, can be used to derive the relation between the the normalized network throughput $\hat{\Sigma}$ and the normalized offered load G. We define the normalized network throughput as the amount of information successfully received by the destination node per time slot, given by $\hat{\Sigma}^{(net)} = \Sigma^{(net)} T_{slot}$. Similarly, we define the offered load as the total amount of information generated by all nodes in the network, given by:

$$G^{(net)} = \sum_{\forall k} G|_{\pi_k^{(net)}} \pi_k^{(net)} = \sum_{\forall n} \sum_{\forall k} P_T^{(n)}|_{\pi_k^{(net)}} \pi_k^{(net)} \cdot nP \; [bits] \tag{2.3.6}$$

where $P_T^{(n)}$ is computed as the probability that a node n transmits nP bytes of information to the next hop.

ii) The normalized throughput $\hat{\Sigma}$ and the average energy consumed per node, in each state, can be used to compute the energy efficiency, η, defined as:

$$\eta = \frac{\hat{\Sigma}^{(net)}}{\sum_{\forall n} E^{(n)}} = \frac{\hat{\Sigma}^{(net)}}{\bar{E}^{(net)}}[bits/J] \tag{2.3.7}$$

where the numerator represents the amount of information received by the sink, while in the denominator we write the sum of average energies consumed by each node in the network.

Now that the basic principles of the proposed analysis and the performance metrics have been introduced, we apply the model to two case studies: Slotted Aloha and a CSMA-based protocol. The thorough analysis of both case studies is provided in the following Sections.

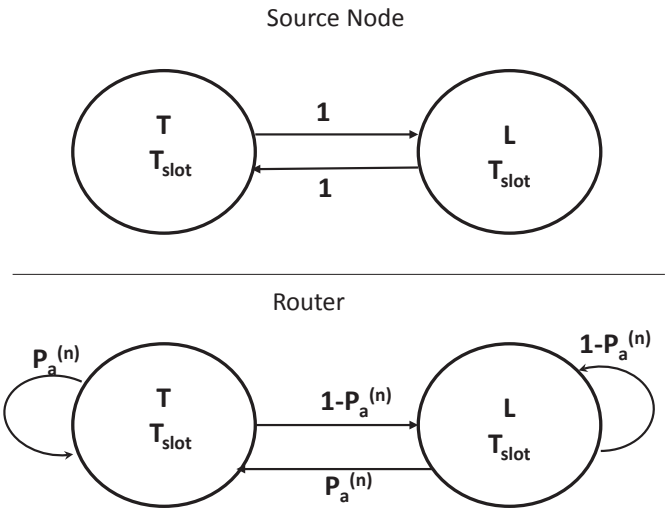

Figure 2.2: SMC for the source and the generic router: Slotted ALOHA.

2.4 Slotted Aloha

2.4.1 Protocol Description

A simple protocol is considered as a first use case: i) slots have duration $T_{slot} = (N+1)\frac{P}{R_b} + \frac{H}{R_b}$; ii) if a node has received a data packet in a slot, it transmits it during the next one, then it goes to Listen state (waiting for other packets).

2.4.2 Node-Level MC

The behavior of the source and of the generic router are described through two (different) state diagrams shown in Figure 2.2. According to this protocol, the holding time of each state is the same, the slot duration; therefore, the state transition diagram is a normal MC (no need to refer to EMCs and SMCs). In this simple case, the

node state stationary probabilities can be expressed as:

$$\pi_T^{(n)} = \begin{cases} P_a^{(0)}\ \pi_L^{(0)} & \text{for } n = 0 \\ \frac{P_a^{(n)}}{1-P_a^{(n)}}\ \pi_L^{(n)} & \text{for } n > 0 \end{cases}$$

$$\pi_L^{(n)} = \begin{cases} \frac{\pi_T^{(0)}}{P_a^{(0)}} & \text{for } n = 0 \\ \frac{1-P_a^{(n)}}{P_a^{(n)}}\ \pi_T^{(n)} & \text{for } n > 0 \end{cases}$$

Solving the system of equations, we obtain:

$$\pi_T^{(n)} = \begin{cases} \frac{P_a^{(0)}}{1+P_a^{(0)}} & \text{for } n = 0 \\ P_a^{(n)} & \text{for } n > 0 \end{cases}$$

$$\pi_L^{(n)} = \begin{cases} \frac{1}{1+P_a^{(0)}} & \text{for } n = 0 \\ 1 - P_a^{(n)} & \text{for } n > 0 \end{cases} \tag{2.4.1}$$

The values of $\pi_T^{(n)}$ and $\pi_L^{(n)}$ depend on $P_a^{(0)}$, as expected; therefore, they depend on the overall network status.

2.4.3 Network-Level FSTD

In the case of Slotted Aloha, the probability to transmit a packet $P_T^{(n)}$ equals the probability to have a packet in the queue $P_a^{(n)}$; therefore, the two sequences of bits describing each state $\{\mathbf{L}, \mathbf{Q}\}$ are equal.

Let us consider first the three-hop network (i.e. $N = 2$; the case with $N = 1$ is trivial, owing to the assumption of half duplex nodes) shown in Figure 2.3. In the initial state **A**, only the source node has a packet to transmit. After packet transmission, which happens with probability 1, the network moves to state **B**. Similarly, the network passes from state **B** to state **C**. From state **C**, where both the source and R_2 transmit their packets, two situations may occur: if the packet transmitted

by the source and interfered by R_2 is captured, the network passes to state **B** with probability c_{02}; otherwise, with probability $\bar{c}_{02}$, the network passes to state **D**. From state **D**, the network goes back to the initial state **A**.

In the same Figure the network with $N = 3$ is shown. In this case, states **A**, **B**, **C** and **D**, though defined through longer bit sequences $\{\mathbf{L}, \mathbf{Q}\}$ where a zero si appended, are still present with the same transition probabilities, except for the transitions from state **C**, which account for events with probability c_{20} and $\bar{c}_{20}$. Two more states are present, **E** and **F**, accounting for cases where the last bit in the sequences $\{\mathbf{L}, \mathbf{Q}\}$ is 1. Moving to the case with $N = 4$ (see Figure 2.4), similar changes happen, with two new states **G** and **H**. The FSTD for $N = 3$ is still present, with changes in the transitions from **E** and **F**.

In general, building the FSTD for a network with $N + 1$ routers is simple when moving from the FSTD with N routers. In fact, each time a router is added in the network, the FSTD changes because the two sequences of bits $\{\mathbf{L}, \mathbf{Q}\}$ get longer by one bit, and two new states are introduced; however, the previous FSTD (for N routers) is contained in the new one (for $N + 1$ routers) with few changes. Let us derive the network state stationary probabilities for the three cases of $N = 2$, 3 and 4, solving Eq. (2.3.3). $N = 2$) The state probabilities vector in this case is $\boldsymbol{\pi}^{(net)} = [\pi_{\mathbf{A}}{}^{(net)}, \pi_{\mathbf{B}}{}^{(net)}, \pi_{\mathbf{C}}{}^{(net)}, \pi_{\mathbf{D}}{}^{(net)}]^T$, and $\mathbf{P}^{(net)}$ is given by:

$$\mathbf{P}^{(net)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & c_{02} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \bar{c}_{02} & 0 \end{bmatrix} \tag{2.4.2}$$

The network state stationary probabilities are found as: $\pi_{\mathbf{A}}{}^{(net)} = \pi_{\mathbf{D}}{}^{(net)} = \frac{1}{4}\bar{c}_{02}$; $\pi_{\mathbf{B}}{}^{(net)} = \pi_{\mathbf{C}}{}^{(net)} = \frac{1}{4}\bar{c}_{02} + \frac{1}{2}c_{02}$;

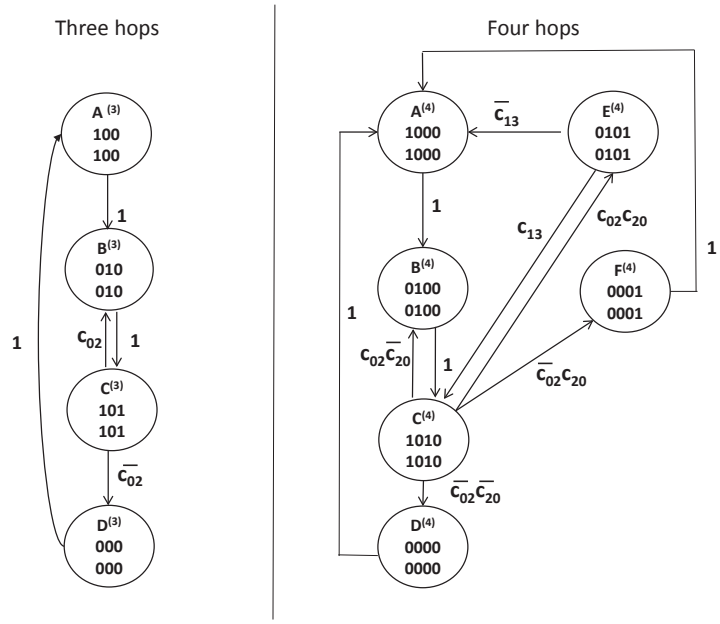

Figure 2.3: Network-level FSTD for a three- and four-hop network: Slotted Aloha.

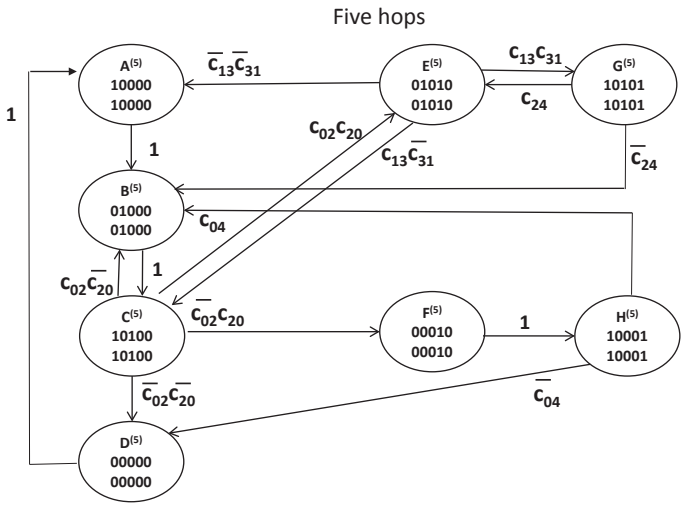

Figure 2.4: Network-level FSTD for a five-hop network: Slotted Aloha.

$N = 3$) In this case, $\mathbf{P}^{(net)}$ is given by:

$$\mathbf{P}^{(net)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & \bar{c}_{13} \\ 1 & 0 & c_{02}\bar{c}_{20} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & c_{13} \\ 0 & 0 & c_{02}c_{20} & 0 & 0 & 0 \\ 0 & 0 & \bar{c}_{02}c_{20} & 0 & 0 & 0 \\ 0 & 0 & \bar{c}_{02}\bar{c}_{20} & 0 & 0 & 0 \end{bmatrix}$$

Once again we derive the network state stationary probabilities; they are reported in Table 2.1 as coefficients of linear equations having variables identified in the first row. $N = 4$)Similarly, for a five hop network $\mathbf{P}^{(net)}$ is given by:

Table 2.1: Network State Stationary Probabilities; Slotted Aloha, $N = 3$.

	$\bar{c}_{02}\bar{c}_{20}$	$c_{02}\bar{c}_{20}$	$c_{02}c_{20}c_{13}$	$c_{02}c_{20}\bar{c}_13$	$\bar{c}_02c_{20}$
${\pi_\mathbf{A}}^{(net)}$	1/4	0	0	1/4	1/4
${\pi_\mathbf{B}}^{(net)}$	1/4	1/2	0	1/4	1/4
${\pi_\mathbf{C}}^{(net)}$	1/4	1/2	1/2	1/4	1/4
${\pi_\mathbf{D}}^{(net)}$	1/4	0	0	0	0
${\pi_\mathbf{E}}^{(net)}$	0	0	1/2	1/4	0
${\pi_\mathbf{F}}^{(net)}$	0	0	0	0	1/4

$$\mathbf{P}^{(net)} = \begin{bmatrix} 0 & 0 & 0 & 0 & \bar{c}_{13}\bar{c}_{31} & 0 & 0 & 1 \\ 1 & 0 & c_{02}\bar{c}_{20} & c_{04} & 0 & 0 & c_{24} & c_{04} \\ 0 & 1 & 0 & 0 & c_{13}\bar{c}_{31} & 0 & 0 & 0 \\ 0 & 0 & c_{02}c_{20} & 0 & 0 & 0 & c_{24} & 0 \\ 0 & 0 & 0 & 0 & c_{13}c_{31} & 0 & 0 & 0 \\ 0 & 0 & \bar{c}_{02}c_{20} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{c}_{02}\bar{c}_{20} & \bar{c}_{04} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{c}_{13}c_{31} & 1 & 0 & 0 \end{bmatrix} \tag{2.4.3}$$

while the network state stationary probabilities are reported in Table 2.2. Starting from the network state stationary probabilities (for the three cases of $N = 2$, 3

Table 2.2: Network State Stationary Probabilities; Slotted Aloha, $N = 4$.

	$c_{02}c_{20}$	$c_{02}\bar{c}_{20}$	$c_{02}c_{03}c_{13}c_{31}$	$c_{13}c_{31}c_{24}$	$c_{02}c_{20}\bar{c}_{13}c_{31}$	$\bar{c}_{02}c_{20}c_{04}$	$\bar{c}_{02}c_{20}\bar{c}_{04}$	$c_{02}c_{20}\bar{c}_{13}c_{31}\bar{c}_{04}$	$\bar{c}_{02}\bar{c}_{20}$
$\pi_{\mathbf{A}}{}^{(net)}$	1/4	0	0	0	0	0	1/6	1/6	1/4
$\pi_{\mathbf{B}}{}^{(net)}$	1/4	1/2	1/4	0	1/4	1/4	1/6	1/6	1/4
$\pi_{\mathbf{C}}{}^{(net)}$	1/4	1/2	1/4	0	1/4	1/4	1/6	1/6	1/4
$\pi_{\mathbf{D}}{}^{(net)}$	0	0	0	0	0	0	1/6	1/6	1/4
$\pi_{\mathbf{E}}{}^{(net)}$	1/4	0	1/4	1/2	1/4	0	0	1/6	0
$\pi_{\mathbf{F}}{}^{(net)}$	0	0	0	0	0	1/4	1/6	0	0
$\pi_{\mathbf{G}}{}^{(net)}$	0	0	1/4	1/2	0	0	0	0	0
$\pi_{\mathbf{H}}{}^{(net)}$	0	0	0	0	1/4	1/4	1/6	1/6	0

and 4), we can determine the network state dependant parameters $P_T^{(n)}$ and $\Sigma^{(n)}$, conditioned to $\pi_k^{(net)}$; they are given in Table 2.3. Combining them with the node-level analysis, we compute the transmission probabilities $P_T^{(n)}$, throughput $\Sigma^{(n)}$ and the energy consumption $E^{(n)}$.

2.4.4 Performance Metrics

We can now use the network state dependant parameters to derive the coefficients needed for the node state stationary probabilities computation according to Eq. (2.4.1). The node state stationary probabilities, given in Table 2.4 are obtained. Finally, we replace the previously reported network and node state stationary probabilities into Eq.s (2.3.5), (2.3.6) and (2.3.7), to compute the average energy consumption, normalized offered load and throughput as well as energy efficiency, that are presented in the numerical results section.

Table 2.3: Parameters for Slotted Aloha.

	$P_T^{(n)}\|_{\pi_k^{(net)}}$											
	3 hop			4 hop				5 hop				
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
$\pi_{\mathbf{A}}{}^{(net)}$	1	0	0	1	0	0	0	1	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	0	1	0	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	1	0	1	1	0	1	0	1	0	1	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	NA	NA	NA	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	0	0	0	0	1	0	1	0	1	0	1	0
$\pi_{\mathbf{F}}{}^{(net)}$	NA	NA	NA	0	0	0	1	0	0	0	1	0
$\pi_{\mathbf{G}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1	0	1	0	1
$\pi_{\mathbf{H}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1	0	0	0	1
	$\Sigma^{(n)}\|_{\pi_k^{(net)}}$											
	3 hop			4 hop				5 hop				
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
$\pi_{\mathbf{A}}{}^{(net)}$	0	0	0	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	1	0	0	1	0	0	0	1	0	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	0	1	c_{20}	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	NA	NA	NA	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	0	0	0	1	0	1	c_{31}	1	0	1	0	0
$\pi_{\mathbf{F}}{}^{(net)}$	NA	NA	NA	0	0	0	1	0	0	1	0	0
$\pi_{\mathbf{G}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	0	1	0	1	$c_{40}c_{42}$
$\pi_{\mathbf{H}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	0	0	0	1	c_{40}

Table 2.4: Node state stationary probabilities for Slotted Aloha.

	$\pi_T\|_{\pi_k^{(net)}}$											
	3 hop			4 hop				5 hop				
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
$\pi_{\mathbf{A}}{}^{(net)}$	1/2	0	0	1/2	0	0	0	1/2	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	0	1	0	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	1/2	0	1	1/2	0	1	0	1/2	0	1	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	NA	NA	NA	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	0	0	0	0	1	0	1	0	1	0	1	0
$\pi_{\mathbf{F}}{}^{(net)}$	NA	NA	NA	0	0	0	1	0	0	0	1	0
$\pi_{\mathbf{G}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1/2	0	1	0	1
$\pi_{\mathbf{H}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1/2	0	0	0	1
	$\pi_L\|_{\pi_k^{(net)}}$											
	3 hop			4 hop				5 hop				
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
$\pi_{\mathbf{A}}{}^{(net)}$	1/2	1	1	1/2	1	1	1	1/2	1	1	1	1
$\pi_{\mathbf{B}}{}^{(net)}$	1	0	1	1	0	1	1	1	0	1	1	1
$\pi_{\mathbf{C}}{}^{(net)}$	1/2	1	0	1/2	1	0	1	1/2	0	1	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	NA	NA	NA	1	1	1	1	1	1	1	1	1
$\pi_{\mathbf{E}}{}^{(net)}$	1	1	1	1	0	1	0	1	0	1	0	1
$\pi_{\mathbf{F}}{}^{(net)}$	NA	NA	NA	1	1	1	0	1	1	1	0	1
$\pi_{\mathbf{G}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1/2	1	0	1	0
$\pi_{\mathbf{H}}{}^{(net)}$	NA	NA	NA	NA	NA	NA	NA	1/2	1	1	1	0

2.5 Case Study: CSMA-based protocol

2.5.1 Protocol Description

A protocol specifically devised for LWSNs denoted as L-CSMA [24] is considered. The main idea behind it, is to take advantage of the linear topology and to assign different levels of priority in the access to the channel; routers closer to the destination have higher priority. Nodes sense the channel for intervals of time of different durations: the shorter it is, the higher will be the priority in the access to the channel. To this aim, each slot is split into two parts (see Fig, 2 in [24]): i) first the transmitter senses the radio channel for an interval $T_{\mathrm{s}}^{(n)}$; we set $T_{\mathrm{s}}^{(n)} = (N - n + 1) \cdot T$, where T (the minimum sensing duration, which is applied by the last router in the line, $n = N$) is set equal to P/R_{b}, that is the time needed to transmit the minimum payload; ii) then, if the channel is sensed as free the packet is transmitted to the next node in the line; otherwise the node switches to receiver mode. At the next slot the node tries again to access the channel. Therefore we have: $T_{slot} = T_s^{(n)} + T_p^{(n)}$ equal to $(N + 2) \cdot T + H/R_{\mathrm{b}}$, whatever is n. Retransmissions are not considered, for reasons already mentioned[4].

2.5.2 Node-Level SMC

The SMC describing the behavior of the source node in saturated conditions is straightforward and is shown in Figure 2.5. As far as routers are concerned, since they

[4]Moreover, in [24] it is shown that with L-CSMA the network throughput does not improve by using retransmissions, while the energy consumption increases.

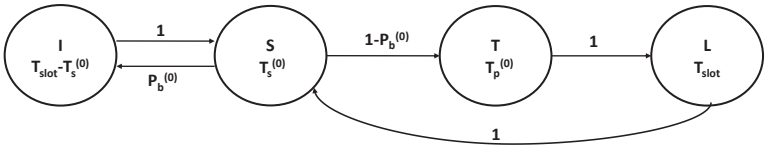

Figure 2.5: SMC for the source node: L-CSMA.

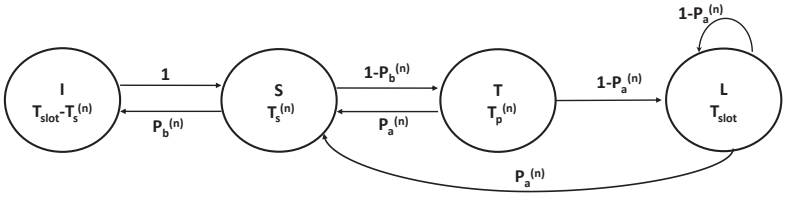

Figure 2.6: MC for a generic router: L-CSMA.

remain in Listen state until a new packet arrives from the previous router, they cannot be described directly through a SMC; we first model the behavior of the generic router through a MC (see Figure 2.6) and then we transform it to the corresponding EMC, shown in Figure 2.7. Since we consider the three MCs reported in Figs.

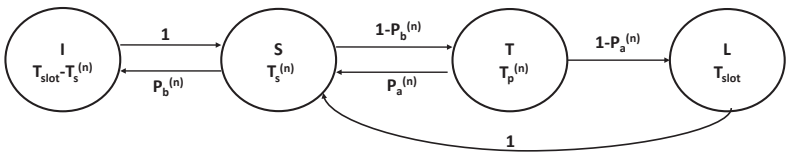

Figure 2.7: EMC for a generic router: L-CSMA.

2.5-2.7 self-explainatory, we do not describe them. Let us only note that Figure 2.7 is general, regardless of the network size (i.e. of N). Now, we first compute the

stationary probabilities of states in the EMC. Having in mind Eq. (2.3.1), we find:

$$\Theta_S^{(n)} = \begin{cases} \Theta_L^{(0)} + \Theta_I^{(0)} & \text{for } n = 0 \\ \Theta_I^{(n)} + P_a^{(n)}\,\Theta_T^{(n)} + \Theta_L^{(n)} & \text{for } n > 0 \end{cases}$$

$$\Theta_T^{(n)} = \begin{cases} (1 - P_b^{(0)})\,\Theta_S^{(0)} & \text{for } n = 0 \\ P_b^{(n)}\,\Theta_S^{(n)} & \text{for } n > 0 \end{cases}$$

$$\Theta_L^{(n)} = \begin{cases} (1 - P_b^{(0)})\,\Theta_T^{(0)} & \text{for } n = 0 \\ (1 - P_a^{(n)})\,\Theta_T^{(n)} & \text{for } n > 0 \end{cases}$$

$$\Theta_I^{(n)} = \begin{cases} P_b^{(0)}\,\Theta_S^{(0)} & \text{for } n = 0 \\ \Theta_S^{(n)}\,P_b^{(n)} & \text{for } n > 0 \end{cases} \tag{2.5.1}$$

The average state holding times are $\mathbb{E}\{H_S^{(n)}\} = T_s^{(n)}$, $\mathbb{E}\{H_T^{(n)}\} = T_p^{(n)}$, $\mathbb{E}\{H_I^{(n)}\} = T_{slot} - T_s^{(n)}$ for all nodes, while $\mathbb{E}\{H_L^{(n)}\} = T_{slot}$ for the source ($n = 0$) and $\mathbb{E}\{H_L^{(n)}\} = \frac{T_{slot}}{P_a^{(n)}}$ for the routers.

As a consequence, the stationary probabilities for each state of the SMC are given by (see Eq. (2.3.2)):

$$\pi_S^{(n)} = \begin{cases} \frac{1}{2-P_b^{(0)}} \frac{T_s^{(0)}}{T_{slot}} & \text{for } n = 0 \\ \frac{P_a^{(n)}}{C^{(n)}} \frac{T_s^{(n)}}{T_{slot}} & \text{for } n > 0 \end{cases}$$

$$\pi_T^{(n)} = \begin{cases} \frac{1-P_b^{(0)}}{2-P_b^{(0)}} \frac{T_p^{(0)}}{T_{slot}} & \text{for } n = 0 \\ \frac{(1-P_b^{(n)})P_a^{(n)}}{C^{(n)}} \frac{T_p^{(n)}}{T_{slot}} & \text{for } n > 0 \end{cases}$$

$$\pi_L^{(n)} = \begin{cases} \frac{1-P_b^{(0)}}{2-P_b^{(0)}}, & \text{for } n = 0 \\ \frac{(1-P_b^{(n)})(1-P_a^{(n)})}{C^{(n)}} & \text{for } n > 0 \end{cases}$$

$$\pi_I^{(n)} = \begin{cases} \frac{P_b^{(0)}}{2-P_b^{(0)}} \frac{(T_{slot}-T_s^{(0)})}{T_{slot}} & \text{for } n = 0 \\ \frac{P_a^{(n)}P_b^{(n)}}{C^{(n)}} \frac{(T_{slot}-T_s^{(n)})}{T_{slot}} & \text{for } n > 0 \end{cases} \tag{2.5.2}$$

where $C^{(n)} = 1 - P_b^{(n)} + P_a^{(n)}P_b^{(n)}$.

2.5.3 Network-Level FSTD

The above states depend on the overall network state, to be derived in this Sub-Section. We consider here the specific case of a 3-hop network ($N = 2$), leaving to the reader the generalisation to larger values of N; the latter should follow the same considerations made for Slotted Aloha. The appendix reports the cases with $N = 3$ and 4.

In Figure 2.8 the FSTD for $N = 2$ is shown. From state **A** the network enters

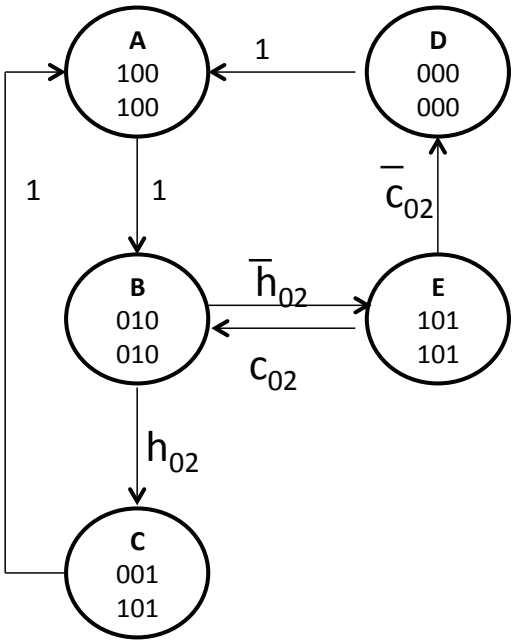

Figure 2.8: Network finite state transition diagram for a 3-hop network: L-CSMA.

state **B** with probability one, as there is only one packet generated. In state **B**, R_1 transmits a packet while the source has a packet in the queue but remains silent for one slot. From state **B**, two transitions may occur: 1) if the source can hear R_2, the network will enter state **C**, since the source has lower priority w.r.t. R_2 and will therefore remain silent, while R_2 will access the channel. During this slot a packet

will be transmitted with success to node R_3 and the network will come back to the initial state **A**; 2) otherwise the network will enter state **E**, where both the source and R_2 will transmit, and a collision occurs. Then, if the packet sent by the source is captured by R_1, the network goes back to **B**, otherwise it moves to state **D**, where no transmissions occur, as the packet sent by the source was lost and, according to the assumptions, the source will wait for one slot.

Therefore, the state probability vector is $\boldsymbol{\pi}^{(net)} = [\pi_{\mathbf{A}}{}^{(net)}, \pi_{\mathbf{B}}{}^{(net)}, \pi_{\mathbf{C}}{}^{(net)}, \pi_{\mathbf{D}}{}^{(net)}, \pi_{\mathbf{E}}{}^{(net)}]^T$, where $[.]^{\mathrm{T}}$ denotes the transpose. The matrix of the state transition probabilities, is given by:

$$\mathbf{P}^{(net)} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & c_{02} \\ 0 & h_{02} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{c}_{02} \\ 0 & \bar{h}_{02} & 0 & 0 & 0 \end{bmatrix} \tag{2.5.3}$$

By solving (2.3.3), we derive:

$$\begin{aligned} \pi_{\mathbf{A}}^{(net)} &= 1/3\ h_{02} + 1/4\ \bar{h}_{02}\bar{c}_{02} \\ \pi_{\mathbf{B}}^{(net)} &= 1/3\ h_{02} + 1/2\ \bar{h}_{02}c_{02} + 1/4\ \bar{h}_{02}\bar{c}_{02} \\ \pi_{\mathbf{C}}^{(net)} &= 1/3\ h_{02} \\ \pi_{\mathbf{D}}^{(net)} &= 1/4\ \bar{h}_{02}\bar{c}_{02} \\ \pi_{\mathbf{E}}^{(net)} &= 1/4\ \bar{h}_{02}\bar{c}_{02} + 1/2 \cdot \bar{h}_{02}c_{02} \end{aligned} \tag{2.5.4}$$

Given the network state stationary probabilities, we can now proceed with the derivation of $P_a^{(n)}$ and $P_b^{(n)}$ as described in Sub-Section IV-B. Since they will be used for the purpose of determining the performance metrics as introduced in Sub-Section IV-C, we make the condition on network states explicit again, as it was made there.

Table 2.5 reports $P_a^{(n)}|_{\pi_k^{(net)}}$ and $P_b^{(n)}|_{\pi_k^{(net)}}$ for $n = 1, 2, 3$ ($P_a^{(0)}|_{\pi_k^{(net)}}$ is not required by the model, being implicitly equal to one).

Table 2.5: Parameters setting for the 3-hop scenario.

	$P_a\|_{\pi_k^{(net)}}$		$P_b\|_{\pi_k^{(net)}}$		
	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$
${\pi_\mathbf{A}}^{(net)}$	0	0	0	0	0
${\pi_\mathbf{B}}^{(net)}$	1	0	0	0	0
${\pi_\mathbf{C}}^{(net)}$	0	1	1	0	0
${\pi_\mathbf{D}}^{(net)}$	0	0	0	0	0
${\pi_\mathbf{E}}^{(net)}$	0	1	0	0	0

2.5.4 Performance Metrics

As previously stated, the probability that node n transmits when the network is in the k-th state, can simply be derived by neglecting the temporal component in $\pi_T^{(n)}$ in Eq. (2.5.2). Therefore:

$$P_T^{(n)} = \begin{cases} \frac{1-P_b^{(0)}}{2-P_b^{(0)}} & \text{for } n = 0 \\ \frac{(1-P_b^{(n)})P_a^{(n)}}{(1-P_b^{(n)})+P_a^{(n)}P_b^{(n)}} & \text{for } n > 0 \end{cases} \tag{2.5.5}$$

By using $P_T^{(n)}$ given by Eq. (2.5.5) and the coefficients from Table 2.5, we compute the probability ${P_\mathbf{T}}^{(n)}$ of each node:

$$P_T^{(0)} = {\pi_\mathbf{A}}^{(net)} + {\pi_\mathbf{E}}^{(net)};$$

$$P_T^{(1)} = {\pi_\mathbf{B}}^{(net)};$$

$$P_T^{(2)} = {\pi_\mathbf{C}}^{(net)} + {\pi_\mathbf{E}}^{(net)};$$

Finally, for the 3-hop network we have that, according to Eq. (3.17) and Table 2.5, the throughput $\Sigma^{(n)}$ is:

$$\Sigma^{(0)} = {\pi_{\mathbf{B}}}^{(net)};$$

$$\Sigma^{(1)} = {\pi_{\mathbf{C}}}^{(net)} + {\pi_{\mathbf{E}}}^{(net)};$$

$$\Sigma^{(2)} = {\pi_{\mathbf{C}}}^{(net)} + {\pi_{\mathbf{E}}}^{(net)} c_{20};$$

We can now derive the coefficients needed according to Eq. (2.5.2) to compute the node state stationary probabilities. They are given in Table 2.6. Obtained values, along with network state stationary probabilities are used to compute node state stationary probabilities. The latter will be used to compute energy consumption, the normalized offered load and energy efficiency, by replacing $\pi_i^{(n)}$ into Eq.s (2.3.5), (2.3.6) and (2.3.7), respectively, along with the corresponding network dependant parameters.

Table 2.6: Node state stationary probabilities for 3-hop scenario.

	$\pi_S\|_{\pi_k^{(net)}}$			$\pi_T\|_{\pi_k^{(net)}}$			$\pi_L\|_{\pi_k^{(net)}}$			$\pi_I\|_{\pi_k^{(net)}}$		
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$
${\pi_{\mathbf{A}}}^{(net)}$	1/2	0	0	1/2	0	0	1/2	1	1	0	0	0
${\pi_{\mathbf{B}}}^{(net)}$	1/2	1	0	1/2	1	0	1/2	1	1	0	0	0
${\pi_{\mathbf{C}}}^{(net)}$	1	0	1	0	0	1	0	1	0	0	0	0
${\pi_{\mathbf{D}}}^{(net)}$	1/2	0	0	1/2	0	0	1/2	1	1	0	0	0
${\pi_{\mathbf{E}}}^{(net)}$	1/2	0	1	1/2	0	1	1/2	0	0	0	0	0

Four-Hop Network

The analysis of the four-hop network is provided in the following, performed analogously to the three-hop network analysis. The finite state transition diagram is reported in Figure 2.9. The state probabilities vector in this case is $\boldsymbol{\pi}^{(net)} =$

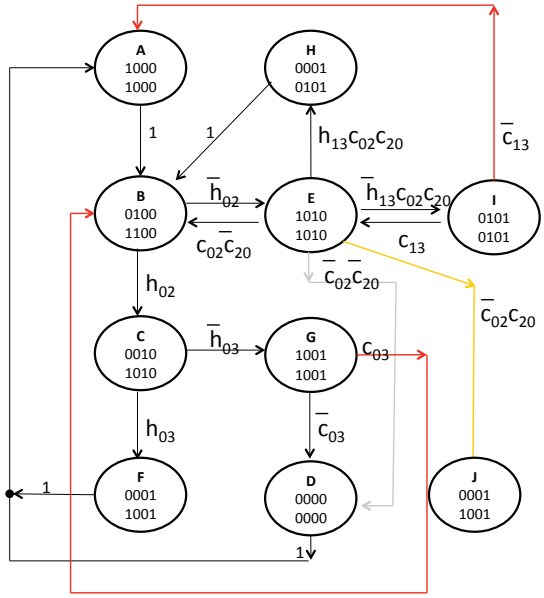

Figure 2.9: Finite state transition diagram for a 4-hop network.

$[\pi_{\mathbf{A}}{}^{(net)}, \pi_{\mathbf{B}}{}^{(net)}, \pi_{\mathbf{C}}{}^{(net)}, \pi_{\mathbf{D}}{}^{(net)}, \pi_{\mathbf{E}}{}^{(net)}, \pi_{\mathbf{F}}{}^{(net)}, \pi_{\mathbf{G}}{}^{(net)}, \pi_{\mathbf{H}}{}^{(net)}, \pi_{\mathbf{I}}{}^{(net)}, \pi_{\mathbf{J}}{}^{(net)}]^T$, and $\mathbf{P}^{(net)}$ is given by:

$$\mathbf{P}^{(net)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\bar{c}_{13} & 1 \\ 1 & 0 & 0 & 0 & c_{02}\bar{c}_{20} & 0c_{03} & 1 & 0 & 0 \\ 0h_{02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{c}_{02}\bar{c}_{20} & 0\bar{c}_{03} & 0 & 0 & 0 \\ 0\bar{h}_{02} & 0 & 0 & 0 & 0 & 0 & 0 & 0c_{13} & 0 \\ 0 & 0 & h_{03} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{h}_{03} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{02}c_{20}h_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{02}c_{20}\bar{h}_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{c}_{02}c_{20} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.5.6}$$

To find the network state probabilities, the system (2.3.3) is solved, obtaining the state probabilities given in the Table 2.7. They are reported in tabular form for

Table 2.7: Network State Stationary Probabilities.

	$h_{02}h_{03}$	$h_{02}\bar{h}_{03}c_{03}$	$h_{02}\bar{h}_{03}\bar{c}_{03}$	$\bar{h}_{02}h_{13}c_{02}c_{20}$	$\bar{h}_{02}c_{02}c_{20}$	$\bar{h}_{02}\bar{h}_{13}c_{02}c_{20}c_{13}$	$\bar{h}_{02}\bar{h}_{13}c_{02}c_{20}\bar{c}_{13}$	$\bar{h}_{02}\bar{c}_{02}\bar{c}_{20}$	$\bar{h}_{02}\bar{c}_{02}c_{20}$
${\pi_{\mathbf{A}}}^{(net)}$	1/4	1/5	0	0	0	0	1/4	1/4	1/4
${\pi_{\mathbf{B}}}^{(net)}$	1/4	1/4	1/5	1/3	1/2	0	1/4	1/4	1/4
${\pi_{\mathbf{C}}}^{(net)}$	1/4	1/4	1/5	0	0	0	0	0	0
${\pi_{\mathbf{D}}}^{(net)}$	0	0	1/5	0	0	0	0	1/4	0
${\pi_{\mathbf{E}}}^{(net)}$	0	0	0	1/3	1/2	1/2	1/4	1/4	1/4
${\pi_{\mathbf{F}}}^{(net)}$	1/4	0	0	0	0	0	0	0	0
${\pi_{\mathbf{G}}}^{(net)}$	0	1/4	1/5	0	0	0	0	0	0
${\pi_{\mathbf{H}}}^{(net)}$	0	0	0	1/3	0	0	0	0	0
${\pi_{\mathbf{I}}}^{(net)}$	1	0	1	0	0	0	0	0	1
${\pi_{\mathbf{J}}}^{(net)}$	0	0	0	0	0	1/2	1/4	0	0

the sake of brevity and simplicity. From the Table 2.7, ${\pi_A}^{(net)}$, for example, can be extrapolated as: ${\pi_A}^{(net)} = \frac{1}{4}h_{02}h_{03} + \frac{1}{5}h_{02}\bar{h}_{03}c_{03} + \frac{1}{4}\bar{h}_{02}\bar{h}_{13}c_{02}c_{20}\bar{c}_{13} + \frac{1}{4}\bar{h}_{02}\bar{c}_{02}\bar{c}_{20} + \frac{1}{4}\bar{h}_{02}\bar{c}_{02}c_{20}$. Similarly, in the rest of the chapter, all the parameters given in this kind of tabular form can be derived in the equivalent way. Applying the same procedure previously described, P_a and P_b that are reported in Table 2.8 are obtained. These values are then, similarly as in the three hop network case, used to compute $\pi_i^{(n)}$ and therefore energy consumption, as well as $P_T^{(n)}$ and $\Sigma^{(n)}$, reported in numerical results section. Node state stationary probabilities are obtained in the same way as in the three-hop case, using the coefficients derived and reported in Table 2.9 as well as previously computed network state stationary probabilities.

Five-Hop Network

The Figure 2.10 shows the finite state transition diagram for the 5-hop network when the L-CSMA protocol is used, while in Table 2.10 its solution is provided. Table 2.11 reports P_a, P_{tx}, P_b and Σ, computed for this case. The detailed computation is omitted for the sake of legibility.

Table 2.8: Parameters setting for the four-hop scenario.

	$P_a\|_{\pi_k^{(net)}}$				$P_b\|_{\pi_k^{(net)}}$			
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
$\pi_{\mathbf{A}}{}^{(net)}$	NA	0	0	0	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	NA	1	0	0	0	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	NA	0	1	0	1	0	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	NA	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	NA	0	1	0	0	0	0	0
$\pi_{\mathbf{F}}{}^{(net)}$	NA	0	0	1	1	0	0	0
$\pi_{\mathbf{G}}{}^{(net)}$	NA	0	0	1	0	0	0	0
$\pi_{\mathbf{H}}{}^{(net)}$	NA	1	0	1	0	1	0	0
$\pi_{\mathbf{I}}{}^{(net)}$	NA	1	0	1	0	0	0	0
$\pi_{\mathbf{J}}{}^{(net)}$	NA	0	0	1	0	0	0	0
	$P_T\|_{\pi_k^{(net)}}$				$\Sigma^{(n)}\|_{\pi_k^{(net)}}$			
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
$\pi_{\mathbf{A}}{}^{(net)}$	1	0	0	0	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	0	1	0	0	1	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	0	0	1	0	0	1	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	0	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	1	0	1	0	0	1	0	0
$\pi_{\mathbf{F}}{}^{(net)}$	0	0	0	1	0	0	1	1
$\pi_{\mathbf{G}}{}^{(net)}$	1	0	0	1	0	0	1	c_{30}
$\pi_{\mathbf{H}}{}^{(net)}$	0	0	0	1	1	0	1	1
$\pi_{\mathbf{I}}{}^{(net)}$	0	1	0	1	1	0	1	c_{31}
$\pi_{\mathbf{J}}{}^{(net)}$	0	0	0	1	0	0	1	1

Table 2.9: Node state stationary probabilities for 4-hop scenario.

	$\pi_S\|_{\pi_k^{(net)}}$				$\pi_T\|_{\pi_k^{(net)}}$				$\pi_L\|_{\pi_k^{(net)}}$				$\pi_I\|_{\pi_k^{(net)}}$			
	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
$\pi_{\mathbf{A}}{}^{(net)}$	1/2	0	0	0	1/2	0	0	0	1/2	1	1	1	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	1/2	1	0	0	1/2	1	0	0	1/2	0	1	1	0	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	1	0	1	0	1/2	0	1	0	1	1	0	1	1/2	0	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	1/2	0	0	0	1/2	0	0	0	1/2	1	1	1	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	1/2	0	1	0	1/2	0	1	0	1/2	1	0	1	0	0	0	0
$\pi_{\mathbf{F}}{}^{(net)}$	1	0	0	1	1/2	0	0	1	1	1	1	0	1/2	0	0	0
$\pi_{\mathbf{G}}{}^{(net)}$	1/2	0	0	1	1/2	0	0	1	1/2	1	1	0	0	0	0	0
$\pi_{\mathbf{H}}{}^{(net)}$	1/2	1	0	1	1/2	0	0	1	1/2	0	1	0	0	1/2	0	0
$\pi_{\mathbf{I}}{}^{(net)}$	1/2	1	0	1	1/2	1	0	1	1/2	0	1	0	0	0	0	0
$\pi_{\mathbf{J}}{}^{(net)}$	1/2	0	0	1	1/2	0	0	1	1/2	1	1	0	0	0	0	0

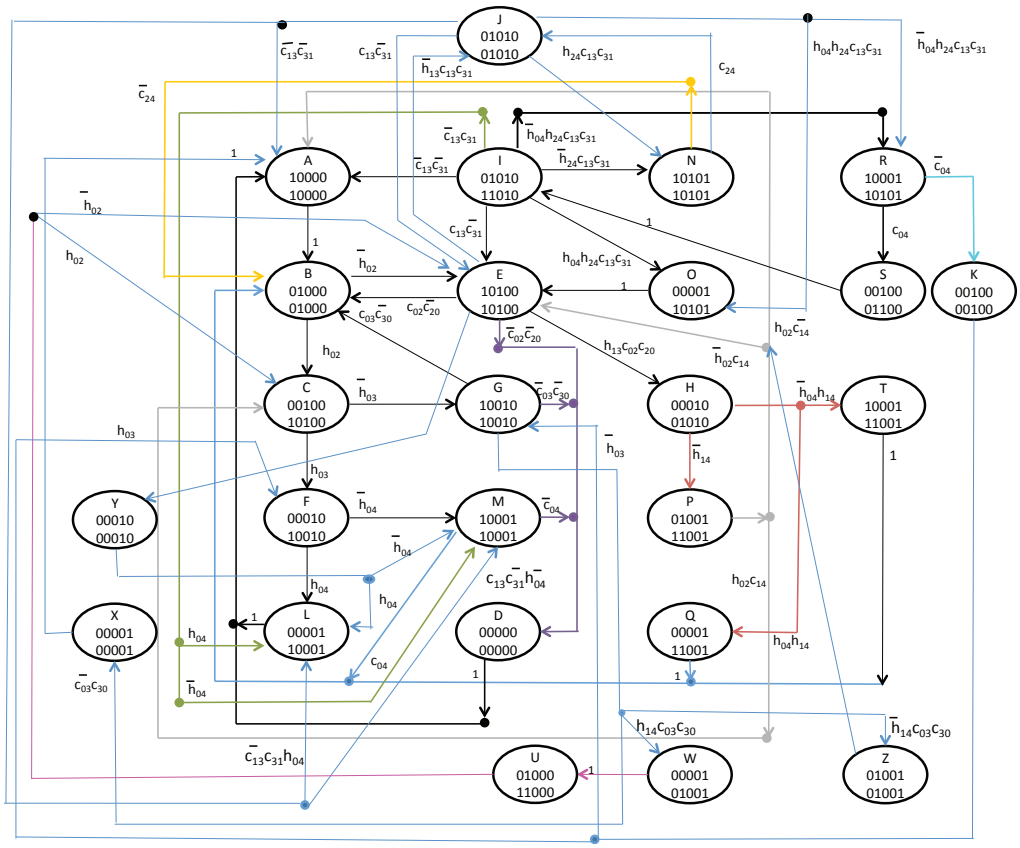

Figure 2.10: Network finite state transition diagram for a 5-hop network.

Table 2.10: L-CSMA: Network state stationary probabilities for a 5-hop network.

	π_A	π_B	π_C	π_D	π_E	π_F	π_G	π_H	π_I	π_J	π_K	π_L	π_M	π_N	π_O	π_P	π_Q	π_R	π_S	π_T	π_U	π_W	π_X	π_Y	π_Z
$h_{02}h_{03}h_{04}$	1/5	1/5	1/5	0	0	1/5	0	0	0	0	0	1/5	0	0	0	0	0	0	0	0	0	0	0	0	0
$h_{02}h_{03}\bar{h}_{04}c_{04}$	0	1/4	1/4	0	0	1/4	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0	0	0	0	0	0
$h_{02}h_{03}\bar{h}_{04}\bar{c}_{04}$	1/6	1/6	1/6	1/6	0	1/6	0	0	0	0	0	0	1/6	0	0	0	0	0	0	0	0	0	0	0	0
$h_{02}\bar{h}_{03}c_{03}c_{30}h_{14}$	0	1/4	1/4	0	0	0	1/4	0	0	0	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0	0
$h_{02}\bar{h}_{03}c_{03}c_{30}\bar{h}_{14}c_{14}$	0	0	1/3	0	0	0	1/3	0	0	0	0	0	0	0	0	1/3	0	0	0	0	0	0	0	0	0
$h_{02}\bar{h}_{03}c_{03}c_{30}\bar{h}_{14}\bar{c}_{14}$	1/5	1/5	1/5	0	0	0	1/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$h_{02}\bar{h}_{03}\bar{c}_{03}c30$	1/5	1/5	1/5	0	0	0	1/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/5	0	0
$\bar{h}_{02}c_{02}c_{20}h_{13}\bar{h}_{04}h_{14}c04$	0	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0	0	0	1/4	1/4	1/4	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}h_{13}\bar{h}_{04}h_{14}\bar{c}_{04}$	0	1/4	0	0	1/4	0	0	1/4	0	0	0	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}h_{13}h04h_{14}$	0	1/4	0	0	1/4	0	0	1/4	0	0	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}h_{13}\bar{h}_{14}c_{14}$	0	0	0	0	1/3	0	0	1/3	0	0	0	0	0	0	0	1/3	0	0	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}h_{13}\bar{h}_{14}\bar{c}_{14}$	1/5	1/5	0	0	1/5	0	0	1/5	0	0	0	0	0	0	0	1/5	0	0	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}\bar{h}_{24}\bar{c}_{24}$	0	1/4	0	0	1/4	0	0	0	0	1/4	0	0	0	1/4	0	0	0	0	0	0	0	0	0	0	1/4
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}\bar{h}_{24}c_{24}$	0	0	0	0	0	0	0	0	0	1/2	0	0	0	1/2	0	0	0	0	0	0	0	0	0	0	1/2
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}c_{04}$	0	0	0	0	0	0	0	0	1/3	0	0	0	0	0	0	0	0	1/3	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{c}_{04}h_{03}$	1/9	1/9	0	1/9	1/9	1/9	0	0	0	1/9	1/9	0	1/9	0	0	0	0	1/9	0	0	0	0	0	0	1/9
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}\bar{c}_{04}c_{03}\bar{c}_{30}$	0	1/6	0	0	1/6	0	1/6	0	0	1/6	1/6	0	0	0	0	0	0	1/6	0	0	0	0	0	0	1/6
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}\bar{c}_{04}\bar{c}_{03}\bar{c}_{30}$	1/8	1/8	0	1/8	1/8	0	1/8	0	0	1/8	1/8	0	0	0	0	0	0	1/8	0	0	0	0	0	0	1/8
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}\bar{h}_{14}c_{03}c_{30}c_{14}\bar{c}_{04}$	0	0	0	0	1/6	0	1/6	0	1/6	0	1/6	0	0	0	0	0	0	1/6	0	0	0	0	0	0	0
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}\bar{h}_{14}c_{03}c_{30}\bar{c}_{14}\bar{c}_{04}$	1/8	1/8	0	0	1/8	0	1/8	0	0	1/8	1/8	0	0	0	0	0	0	1/8	0	0	0	0	0	0	1/8
$h02c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}h_{14}c_{03}c_{30}\bar{c}_{04}$	0	1/7	0	0	1/7	0	1/7	0	0	1/7	1/7	0	0	0	0	0	0	1/7	0	0	0	1/7	0	0	1/7
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{24}\bar{h}_{04}\bar{h}_{03}\bar{c}_{03}c_{30}\bar{c}_{04}$	1/8	1/8	0	0	1/8	0	1/8	0	0	1/8	1/8	0	0	0	0	0	0	1/8	0	0	0	0	1/8	0	1/8
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}\bar{c}_{13}c_{31}h_{04}$	1/5	1/5	0	0	1/5	0	0	0	0	1/5	0	1/5	0	0	0	0	0	0	0	0	0	0	0	0	1/5
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}\bar{c}_{13}c_{31}\bar{h}_{04}c_{04}$	0	1/4	0	0	1/4	0	0	0	0	1/4	0	0	1/4	0	0	0	0	0	0	0	0	0	0	0	1/4
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}\bar{c}_{13}c_{31}\bar{h}_{04}\bar{c}_{04}$	1/6	1/6	0	1/6	1/6	0	0	0	0	1/6	0	0	1/6	0	0	0	0	0	0	0	0	0	0	0	1/6
$\bar{h}_{02}c_{02}c_{20}\bar{h}_{13}c_{13}c_{31}h_{04}h_{24}$	0	0	0	0	1/3	0	0	0	0	1/3	0	0	0	0	1/3	0	0	0	0	0	0	0	0	0	1/3
$\bar{h}_{02}\bar{c}_{02}c_{20}h_{04}$	1/5	1/5	0	0	1/5	1/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/5	0	0
$\bar{h}_{02}\bar{c}_{02}c_{20}\bar{h}_{04}c_{04}$	0	1/4	0	0	1/4	0	0	0	0	0	0	0	1/4	0	0	0	0	0	0	0	0	0	0	1/4	0
$\bar{h}_{02}\bar{c}_{02}c_{20}\bar{h}_{04}\bar{c}_{04}$	1/6	1/6	0	1/6	1/6	0	0	0	0	0	0	0	1/6	0	0	0	0	0	0	0	0	0	0	1/6	0
$h_{02}\bar{h}_{03}\bar{c}_{03}\bar{c}_{30}$	1/5	1/5	1/5	1/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$h_{02}\bar{h}_{03}c_{03}\bar{c}_{30}$	0	1/3	1/3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{h}_{02}c02c_{20}\bar{h}_{13}\bar{c}_{13}\bar{c}_{31}$	1/4	1/4	0	0	1/4	0	0	0	0	1/4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/4
$\bar{h}_{02}\bar{c}_{02}c_{20}\bar{h}_{13}c_{13}\bar{c}_{31}$	0	0	0	0	1/2	0	0	0	0	1/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/2
$\bar{h}_{02}c_{02}\bar{c}_{20}$	0	1/2	0	0	1/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{h}_{02}\bar{c}_{02}\bar{c}_{20}$	1/4	1/4	0	1/4	1/4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.11: Parameters setting for the five-hop scenario.

	$P_a\|_{\pi_k^{(net)}}$				$P_b\|_{\pi_k^{(net)}}$					$P_T\|_{\pi_k^{(net)}}$					$\Sigma\|_{\pi_k^{(net)}}$				
	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_0$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
$\pi_{\mathbf{A}}{}^{(net)}$	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{B}}{}^{(net)}$	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
$\pi_{\mathbf{C}}{}^{(net)}$	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{D}}{}^{(net)}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\pi_{\mathbf{E}}{}^{(net)}$	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{F}}{}^{(net)}$	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
$\pi_{\mathbf{G}}{}^{(net)}$	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0
$\pi_{\mathbf{H}}{}^{(net)}$	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0
$\pi_{\mathbf{I}}{}^{(net)}$	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0
$\pi_{\mathbf{J}}{}^{(net)}$	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0
$\pi_{\mathbf{K}}{}^{(net)}$	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
$\pi_{\mathbf{L}}{}^{(net)}$	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
$\pi_{\mathbf{M}}{}^{(net)}$	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	c_{40}
$\pi_{\mathbf{N}}{}^{(net)}$	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	$c_{40}c_{42}$
$\pi_{\mathbf{O}}{}^{(net)}$	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	1	1
$\pi_{\mathbf{P}}{}^{(net)}$	1	0	0	1	1	0	0	0	0	0	1	0	0	1	1	0	0	1	c_{42}
$\pi_{\mathbf{Q}}{}^{(net)}$	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1
$\pi_{\mathbf{R}}{}^{(net)}$	0	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1	0	1	c_{40}
$\pi_{\mathbf{S}}{}^{(net)}$	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0
$\pi_{\mathbf{T}}{}^{(net)}$	1	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	c_{40}
$\pi_{\mathbf{U}}{}^{(net)}$	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0
$\pi_{\mathbf{W}}{}^{(net)}$	1	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1
$\pi_{\mathbf{X}}{}^{(net)}$	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
$\pi_{\mathbf{Y}}{}^{(net)}$	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
$\pi_{\mathbf{Z}}{}^{(net)}$	1	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	c_{41}

2.6 Numerical Results

In this Section, numerical results achieved through the approach proposed are discussed. They are compared to simulation results for the sake of validation purposes (though our approach does not include approximations). The simulator used is proprietary; it was formerly validated through comparison with ns-3 [24]. Simulations were reproducing the Simulation results have been obtained by averaging over 10,000 packets generated by the origin.

In the following, if not otherwise specified, we set $R_{\mathrm{b}} = 250$ kbit/s, $k = -40$ dB, $\beta = 3$, $\alpha = 6$ dB, $H = 19$ bytes, $P_{\mathrm{R}_{min}} = -90$ dBm and $P_{\mathrm{S}_{min}} = -95$ dBm. The information regarding power consumption of nodes was taken from Texas Instruments CC2530 datasheets (assuming 3.3 V voltage supply) [53], that is, a widely used platform for wireless sensor networks: $W_S = 89.1[mW], W_T = 105.6[mW], W_R = 77.55[mW], W_I = 24.75[mW]$. Let us consider the per-node metrics first. The payload size is set to $P = 20$ bytes; so the expected values of $\Sigma^{(n)}$ will be upper bounded by $P(n+1)/T_{slot} = 20(n+1)/T_{slot}$.

Fig. 2.11 shows the transmission probability for the different routers in a LWSN with N =2, 3 and 4 (three, four and five hops) using Slotted Aloha. Regardless of N, R_0 (the source) has constant ${P_T}^{(0)} = 0.5$ since it only depends on the saturated packet generation process. On the opposite, the routers have smaller values of transmission probability, owing to the usuccessful transmissions happening in the line, which reduce the amount of traffic generated by routers closer to the sink. In the case of $N = 2$, R_1 is always successful because during its transmission the source and R_2 are idle; therefore, R_1 and R_2 generate the same amount of traffic. On the other hand, for

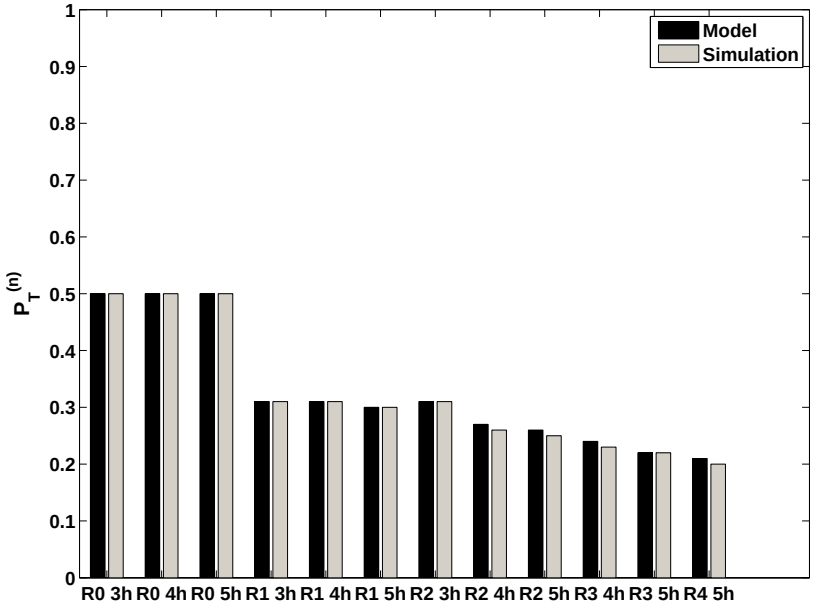

Figure 2.11: Slotted Aloha: per-node transmission probability $P_T^{(n)}$.

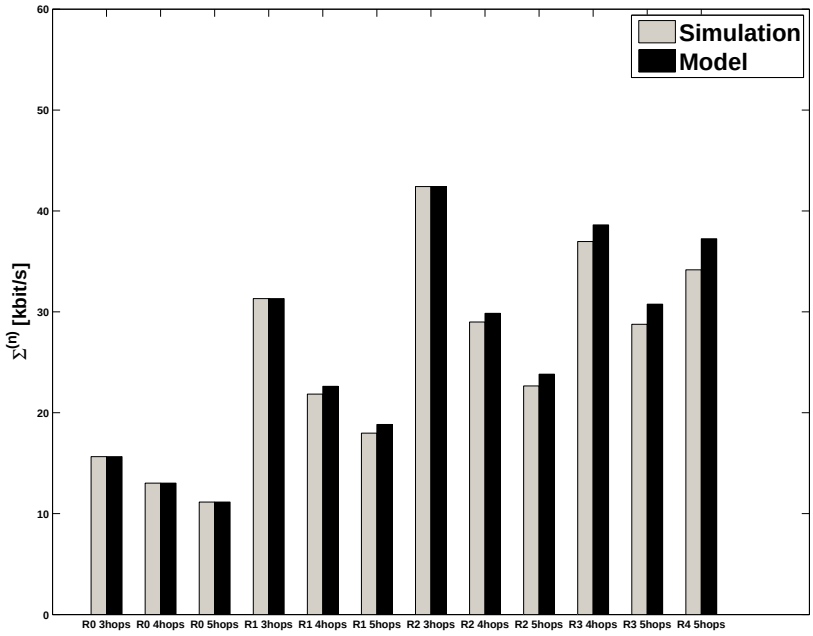

Figure 2.12: Slotted Aloha: Per-node throughput.

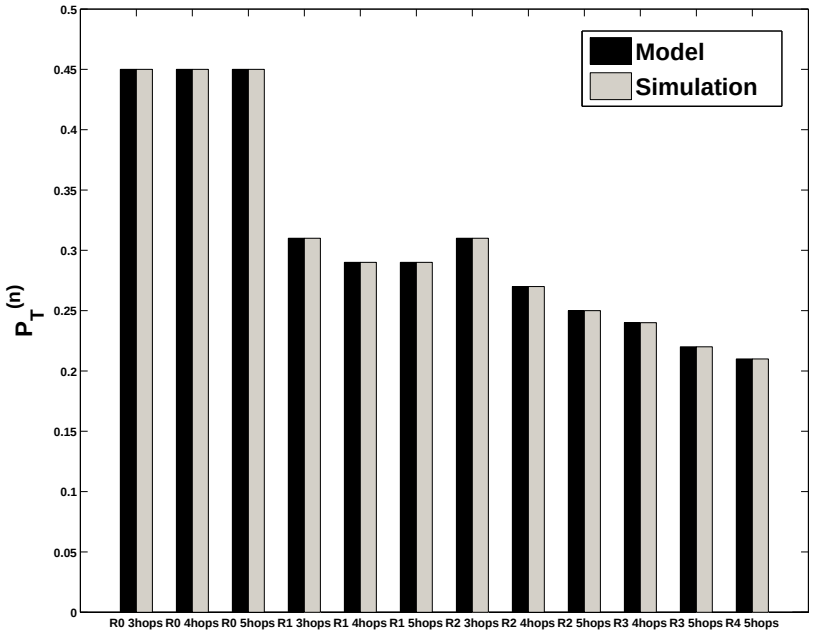

Figure 2.13: L-CSMA: per-node transmission probability $P_T^{(n)}$.

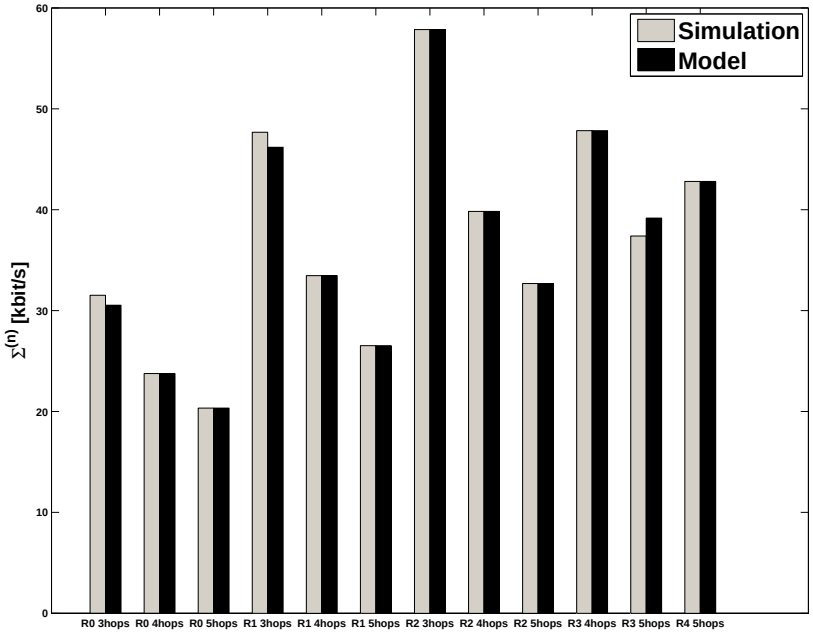

Figure 2.14: L-CSMA: Per-node throughput $\Sigma^{(n)}$.

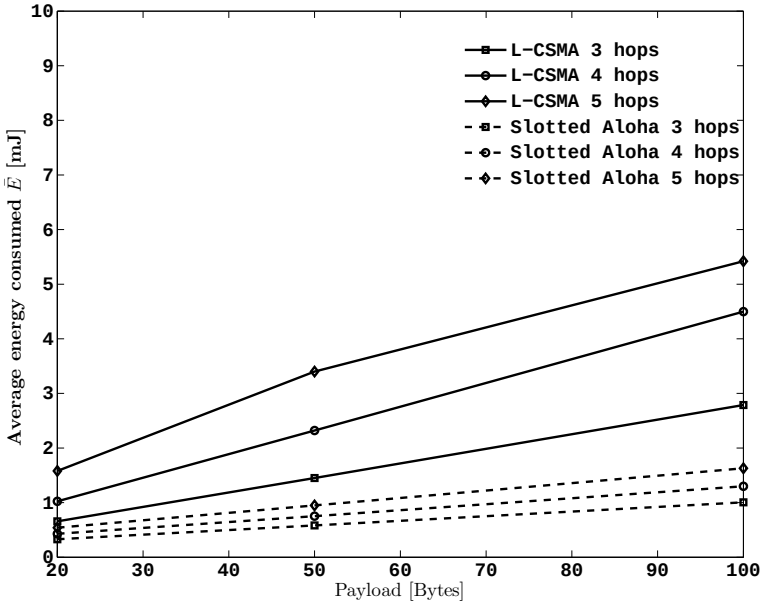

Figure 2.15: Average energy consumption for L-CSMA and Slotted Aloha.

larger N, R_1 transmissions can be affected by the interference of routers closer to the sink, and therefore R_2 has lower transmission rate than R_1. Fig. 2.12 shows the per-node throughput with N =2, 3 and 4 for Slotted Aloha. As expected, it decreases for larger N, increases along the path due to the data aggregation imposed. The value of $\Sigma^{(n)}$.

Similar considerations regarding ${P_T}^{(n)}$ and $\Sigma^{(n)}$ can be made for L-CSMA observing Fig. 2.13. The transmission probability of the source in this case is lower than 0.5, owing to the non zero value of $P_b^{(0)}$ (see Eq. (2.5.5)). Moreover, it should be noted that due to the nature of L-CSMA protocol (nodes closer to the destination have priority in transmission, while nodes farther from the destination will be inhibited in transmission), no critical load can be observed.

In all figures shown above, results obtained through the model fit perfectly with simulations. Let us consider now network performance metrics; we report results

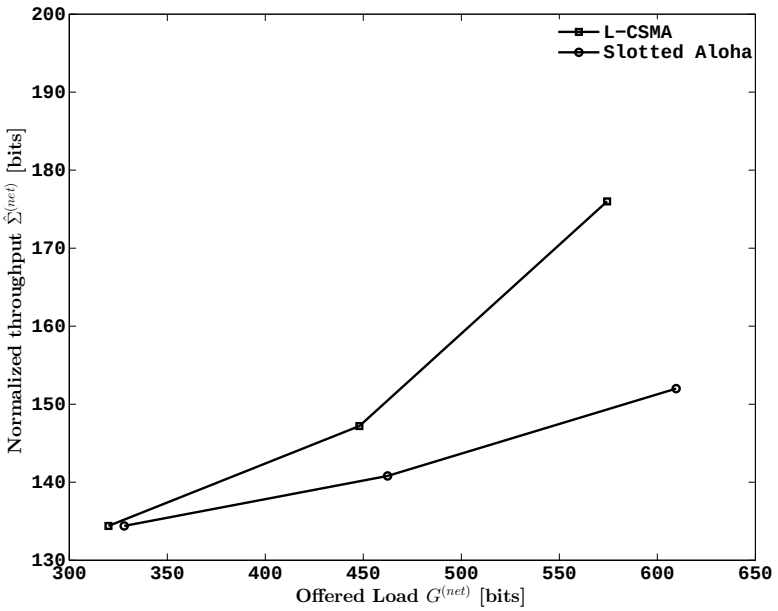

Figure 2.16: Normalized throughput $\hat{\Sigma}^{(net)}$ versus offered load $G^{(net)}$ for L-CSMA and Slotted Aloha.

regarding the energy consumption and network throughput, obtained through the proposed model. Figure 2.15 shows the average energy consumed, averaged among all nodes, given by: $\bar{E} = \sum_{\forall n} E^{(n)}$ for both L-CSMA and Slotted Aloha. As expected, the energy consumed increases when increasing the number of nodes in the network for both protocols. Furthermore, the average energy increases with the payload size for both protocols, since the transmission takes more time. In the case of L-CSMA, this is also due to the fact that when we increase the number of nodes, sensing durations increase as well. As a consequence, it can be seen that L-CSMA performs worse that Slotted Aloha. As previously mentioned, higher number of nodes implies longer slot duration, according to L-CSMA. For the sake of fair comparison, we fixed the same slot duration for Slotted Aloha as well.

Figure 2.16 demonstrates how the proposed model can be used to estimate the

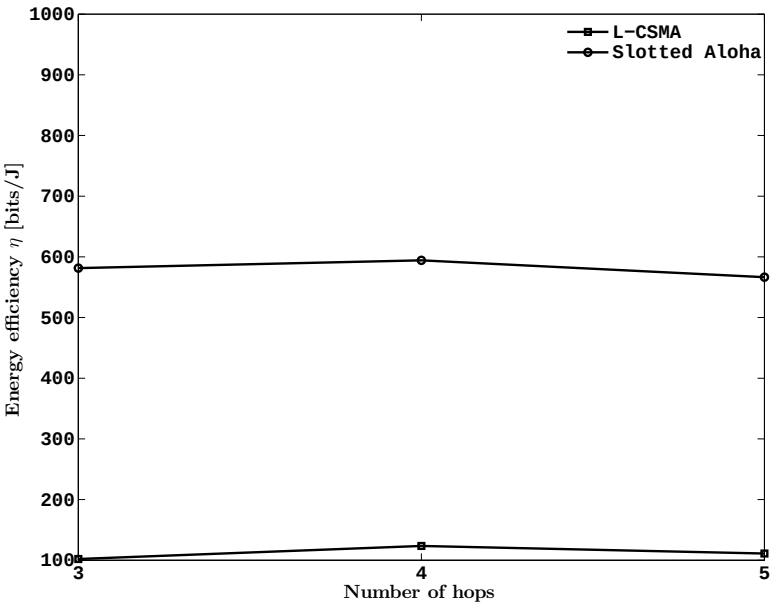

Figure 2.17: Energy Efficiency η versus number of hops for L-CSMA and Slotted Aloha.

performance of the protocols in terms of normalised network throughput, $\hat{\Sigma}^{(net)}$ versus offered load, $G^{(net)}$, where it can be seen that L-CSMA outperforms Slotted Aloha.

Finally, Figure 2.17 shows the energy efficiency computed according to Eq. (2.3.7) for both Slotted Aloha and L-CSMA, considering three, four and five hop network. It can be seen that Slotted Aloha outperforms L-CSMA, since we neglect the energy spent in Listen and Idle states on one hand, and on the other the transmission probability is very similar for both L-CSMA and Slotted Aloha. More importantly, it can be noted that the proposed model can serve as a valuable tool for the protocol performance estimation; Figures 2.16 and 2.17 clearly demonstrate advantages and disadvantages of both protocols considered in this paper.

2.7 Conclusions

In this chapter a novel mathematical approach, combining node-level SMCs and network-level FSTD, for modeling multi-hop LWSNs using contention-based MAC protocols was presented. It allows the derivation of per-node and network level performance metrics, such as throughput and energy efficiency. The application of the approach to Slotted Aloha and a CSMA-based protocol, validated the methodology devised. The extension to other MAC protocols is straightforward.

Though the model complexity increases with the number of routers in the network, the approach is scalable, since derivation of the FSTD for $N+1$ routers is not a difficult task, as long as it is known for N.

The approach allows accurate performance assessment of different MAC protocols for LWSNs, under a unique framework, a fact that is essential for a fair comparison.

Finally, it should be noted that the proposed model or some of its' components could be used as a tool in the downscaling procedure, since information related to connectivity levels among nodes (as probability to receive the packet, the probability to hear transmissions of other nodes, etc.) could be extracted from it. In fact, in the following Chapter, where the downscaling methodology is described in details, the possibility to use mathematical models will be discussed.

Chapter 3

Testing protocols for the IoT and Smart City applications

This chapter describes the experimental work performed during the PhD course. The focus is on smart lighting and smart building systems. All experiments were performed on testbeds available in the laboratory at the University of Bologna, established under the umbrella of a European project, Newcom♯, that is described in details in the first part of this chapter. The second part is dedicated to the description, implementation and validation of the downscaling procedure that allows to suitably select nodes of a testbed and reproduce real world networks. Once the proposed methodology is validated, in the last part of the chapter it is applied to:

- Optimisation of routing protocols for smart lighting systems (Section 3.4);
- Performance evaluation of routing protocols for smart building application (Section 3.5).

In both cases the downscaling methodology was applied to suitably select nodes in order to reproduce realistic operating conditions. Nevertheless, for the sake of conciseness and simplicity, the description of the procedure is omitted for those cases,

once the example of implementation considering sample applications is presented in details.

3.1 Newcom♯ Project

NEWCOM♯ (Network of Excellence in Wireless Communications [1],) is a project funded under the umbrella of the 7th Framework Program of the European Commission (FP7-ICT-318306). NEWCOM♯ pursued long-term, interdisciplinary research on the most advanced aspects of wireless communications like finding the ultimate limits of communication networks, opportunistic and cooperative communications, or energy and bandwidth-efficient communications and networking. The project consisted of both, theoretical and experimental research tracks. Within the experimental track, a EuWIn laboratory was established. EuWIn aims at developing fundamental research through experimentations. The laboratory is composed of three sites, targeting separate technologies and fields of experimentation: the laboratories of the research center CTTC of Barcelona (Spain), of CNIT/University of Bologna (Italy), and of the EURECOM institute of Sophia-Antipolis (France). The three institutions had developed experimental facilities in the context of other projects in the past years, and have committed to make them available to other Newcom# partners, through an integrated and open framework. The three EuWIn sites cover aspects related to radio interfaces (mainly based on MIMO and PHY-layer algorithms) and localization techniques, at CTTC, flexible radio technologies over MIMO platforms, at CNRS/EURECOM and IoT and Smart City applications, at CNIT/UniBO. Experimental work presented in this thesis was performed at EuWIn Bologna premises.

[1] See the website: http://www.newcom-project.eu/

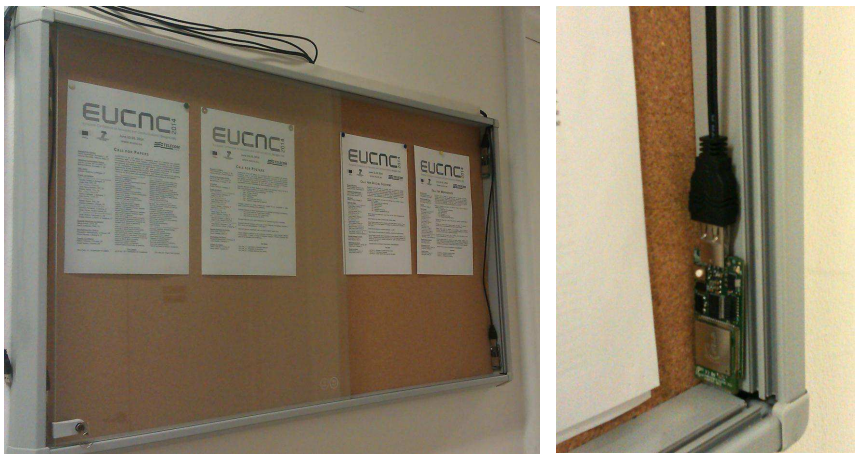

Figure 3.1: Flextop deployment at the University of Bologna.

In the following, a description of facilities available at University of Bologna and in particular of the testbed that was used, will be given.

The EuWIn site in Bologna offers facilities for testing and benchmarking radio network technologies for the future developments of the Internet of Things. In particular, the lab provides more than 100 wireless nodes implementing different types of radio interfaces, distributed according to different platforms:

- Flexible Topology Testbed (Flextop): 53 nodes in fixed positions, distributed along the corridor at the University of Bologna (see Figure 3.2);
- Data Sensing and Processing Testbed (DataSens): 50 wireless nodes equipped with luminosity and temperature sensors and 50 mobile radio nodes (shown in Figure 3.3) for testing delay tolerant routing techniques;
- Localization Testbed: 50 Ultra-Wide Band nodes to test localization algorithms.

The work presented in this thesis refers to experiments performed mainly on Flextop platform as well as on DataSens mobile nodes. Both, Flextop and DataSens nodes

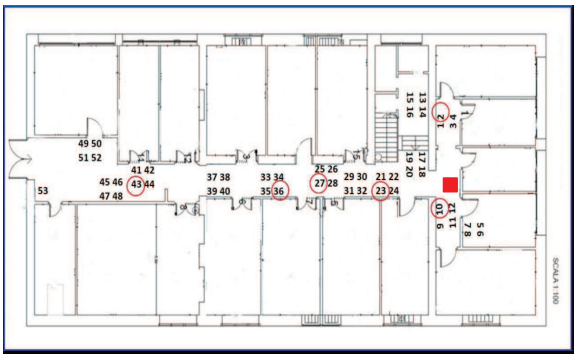

Figure 3.2: Flextop nodes map.

are based on Texas Instruments CC2530 system on chip [54] . Texas Instruments (TI) CC2530 are IEEE 802.15.4 compliant (see Section 1.5.1 for details). The transmit power of the devices ranges from 0 to 15 dBm, while the receiver sensitivity is equal to -92 dBm. Regarding the MAC layer, experiments presented in this thesis refer to the non beacon-enabled mode, employing a carrier sense multiple access with CSMA/CA (see Section 1.5.1). On top of this, different network protocols were deployed and tested, as it will be described in the sequel.

The main objectives of the Flextop platform are: i) to test and fairly compare different routing protocols; ii) to certify simulators implementing IEEE 802.15.4-based networks. More specifically, all the network protocols that can be used over IEEE 802.15.4, as ZigBee, 6lowPAN, or proprietary solutions, could be tested on EuWIn.

The Flextop platform consists of 53 nodes, located in notice boards along a corridor at the University of Bologna. Thirteen notice boards, with four nodes per notice board, are deployed in fixed positions (see Fig. 3.4). Node 53, at the end of the corridor, acts as the coordinator of the network.

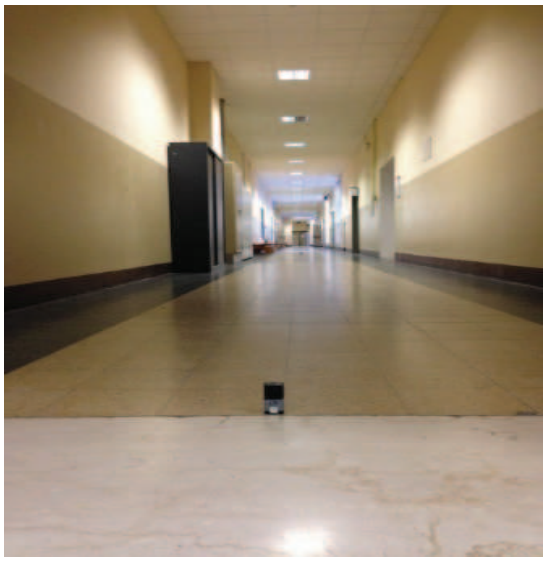

Figure 3.3: DataSens nodes.

The main strength of Flextop is that the experimental environment is stable for the total duration of the experiment, thus making the results replicable, based on the following: i) nodes are at fixed and known positions; ii) channel gains between each pair of nodes are measured at the beginning of each test, as it will be described later; iii) experiments are performed during the night, when nobody is present, avoiding uncontrollable channel fluctuations. Since the experimental environment is well controllable and known a priori, fair comparison among results of tests performed at different time instances is possible.

In order to emulate different environmental conditions and applications, a part of experiments was performed using the DataSens mobile nodes, shown in Figure 3.3. The only difference in hardware features with respect to Flextop devices stands in the fact that DataSens nodes are equipped with batteries.

3.2 The downscaling methodology

In this section the downscaling methodology that is proposed is presented. The aim is to reproduce a real world deployment, characterised by a given number of nodes located in given positions and transmitting at a given power level, on an indoor controllable testbed. Reproducibility refers to the degree of agreement between measurements or observations conducted on separate specimens in different locations. Therefore, the aim of this procedure is to identify the proper subset of nodes of the controllable testbed, their respective locations and the level of transmit power they have to use, such that results obtained through experiments running over the downscaled testbed are in agreement with those observed in the real world deployment.

The procedure is based on the following premise: if we are able to reproduce the channel gains between each nodes pair in the network, we will be able to obtain the same network performance with a good approximation. In the following, channel gains will be used to define the *connectivity level* between two nodes, that is the probability that the data transmitted by one node is correctly received by the other (and viceversa). The level of connectivity among nodes has a strong impact on: i) the topologies formed in the network, that is the set of paths connecting transmitters and the respective receivers; ii) the number of nodes each node can "hear", that is the set of nodes that are not hidden to the given one and with which it will not interfere; iii) the level of interference possibly generated by each node on the other nodes in the network. All the above mentioned items strongly affect the performance of the network, therefore reproducing the connectivity brings to reproducing the network performance, with large probability. This does not exclude the possibility to define

another metric or criteria apart from *connectivity level*, that will then be translated into objective function. Nevertheless, once defined the suitable metric, the sequence of steps proposed by this methodology is the same.

It is important to underline that the proposed downscaling approach refers to downscaling in terms of space and that it is possible when the controllable testbed provides the following degrees of freedom: i) a number of available nodes, $N^{(c)}$, larger than the number $N^{(r)}$ of nodes in the real world deployment, so that a large number of options for the subset selection is possible; ii) the possibility to set different levels of transmit power (or any other parameter affecting the connectivity among nodes) to be used by the nodes in the downscaled testbed. With reference to the point i) above, in Sec. 3.3.2.6 the impact of having different ratios $N^{(c)}/N^{(r)}$ on the precision of the reproducibility is shown: results demonstrate that if the number of nodes in the controllable testbed is two or three times that of the real deployment, a very good fitting in terms of performance can be achieved.

It is also important to underline that the downscaling methodology presented in this thesis could be applied to whatever an hardware platform, provided that the two platforms used in the real deployment and in the downscaled testbed are compatible. In this paper the procedure is applied to testbeds (real and controllable) using exactly the same hardware and radio. The proposed methodology is shown in Figure 3.4, where in white the steps that require programming of devices and running of experiments are underlined, and in grey the processing and design tasks. The methodology consists of two parts: i) the identification of the downscaled testbed and ii) its utilization. The first part is composed of the following steps:

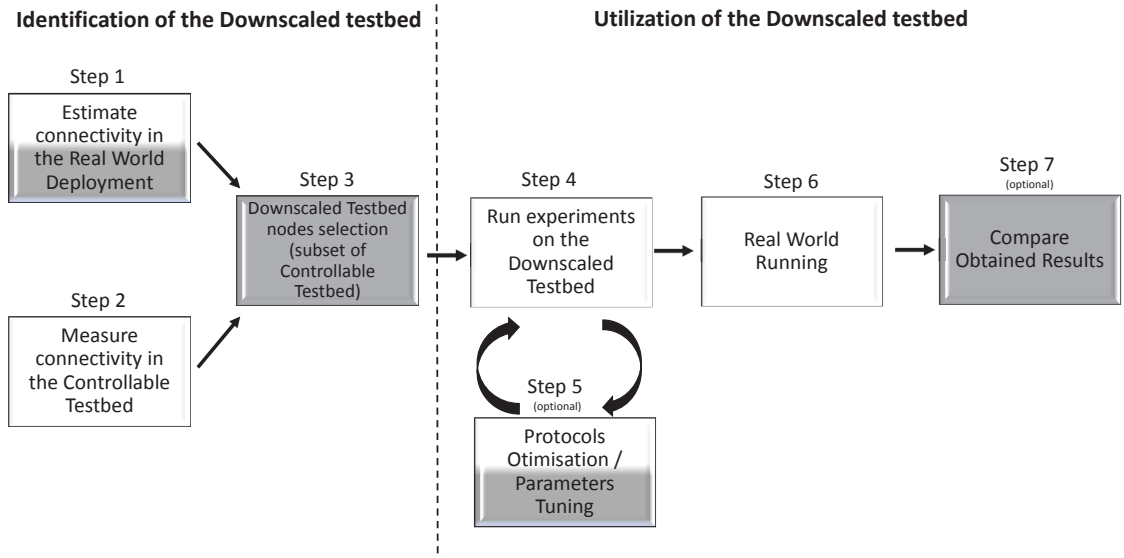

Figure 3.4: Downscaling methodology.

1. Estimation of the level of connectivity among nodes in the real world deployment;

2. Measurement of the level of connectivity among nodes in the controllable testbed;

3. Selection of the downscaled testbed nodes, which is the subset of nodes of the controllable testbed properly reproducing the real world network.

The second part is composed of the following steps:

4. Run experiments on the identified downscaled testbed;

5. (Optional) Optimise the protocols and/or parameters tuning;

6. Real world running, which refers to running the application on the real world deployment;

7. (Optional) Compare results achieved on the two testbeds (real world and downscaled).

In the rest of this section the above mentioned steps are specified, followed by an example of implementation of the procedure is presented in details.

3.2.1 State of the art

The MiNT [55] and Orbit [56] testbeds attempt to shrink a wireless network into a smaller space while maintaining link characteristics through power control. Authors reduce transmission power via software and/or radio frequency components, and received power via augmented environmental noise and/or controllable attenuators. The work presented in [57] focuses on the reliability of spatial scaling of wireless networks as well. [58] presents a work dealing with the emulation of the performance of real world networks on an indoor wireless testbed. In particular, they tend to replicate each link from the real network on the indoor testbed, focusing on the downlink signal-to-noise-ratio mapping.

In contrast to the above works, the work presented here deals with 802.15.4-based networks and proposes a different algorithm to select nodes to create the downscaled testbed. It should be underlined that the proposed methodology, however, is not specific only for the described testbeds, but it is general and independent of hardware and environment.

While to the best of Author's knowledge there are no other similar works dealing with downscaling of real world deployments to an indoor testbed, there are plenty of works dealing with experiments exploiting indoor or outdoor testbeds, which deserve citation here as they inspired the motivation for this work. [59] described a testbed

having a dual purpose: on one hand it allows real world experimentation of IoT related technologies and, on the other hand, it supports the provision of smart city services aimed at enhancing the quality of life in the city of Santander. [60] provides a comprehensive survey of the enabling technologies, protocols, and architecture for an urban IoT referring to a practical implementation of this concept, named Padova Smart City. The target application consists of a system for collecting environmental data and monitoring the public street lighting by means of 300 wireless nodes deployed in the city of Padova. The work [61] presents the VESNA wireless sensor network platform and its role in experimentally-driven research and development. In [62], Authors propose an indoor testbed which is built on the ceiling board of the demonstration room. The purpose of this testbed is to evaluate various types of algorithms and protocols before using them in real world applications. The work [63] gives an overview of all aspects concerning the feasibility of large scale wireless sensor network deployments. Authors refer to a testbed deployed in the buildings of the Department of Information Engineering at the University of Padova. The paper describes different hardware and software architectures for large scale wireless sensor networks, but also efforts that have to be made in terms of communication protocols and strategies for the organization of large testbeds. Twist [64] is a scalable and flexible indoor testbed supporting experiments with heterogeneous node platforms. The testbed consists of 100 TelosB motes spread over a three floor office building. The SensLAB [65] is a testbed consisting of 1000 sensor nodes available for distributed embedding sensor network applications and distributed systems research.

3.2.2 Identification of the downscaled testbed

- **Estimation of the connectivity in the real world deployment**

  The first step is the characterisation of the real world deployment, that is the estimation of the level of connectivity among its nodes. This can be performed in several ways. Among the different approaches there are: i) direct measurement of the path loss between each pair of devices; ii) direct measurement of the path loss between some pairs of devices and estimation of the path loss among all the remaining pairs of devices; iii) use of adequate statistical models to describe the propagation environment; iv) use of ray tracing tools. Therefore this step could be implemented through experiments on the field and/or mathematical evaluation; for this reason the block in Fig. 3.4 is white and grey.

  Let I be the set of nodes in the real world deployment, whose size, $|I|$, is equal to $N^{(r)}$; $P_{ik}^{(r)}$ denotes the average power received by node $i \in I$ when node $k \in I$ is transmitting, provided that the number of received packets is larger than a given threshold. If the number of packets received over the link is lower than the threshold, the two nodes are considered as not connected and a null element will be present in the matrix.

  Therefore, once a suitable or feasible method to compute connectivity among nodes is found, the connectivity matrix, $\mathbf{P}^{(r)}$, of size $N^{(r)}\text{x}N^{(r)}$ can be derived. The matrix $\mathbf{P}^{(r)}$ will represent the outcome of this first step.

  It is important to underline that in this work the average received power was used as metric, instead of its full statistics, to keep the complexity of the methodology under control. In fact, for the procedure to be really useful to IoT developers, the amount of processing on the real deployment tests should be as small

as possible.

- **Measurement of the connectivity in the controllable testbed**
  The second step requires the measurement of the connectivity matrix of the controllable testbed. Note that different matrices could be derived by setting different parameters affecting the connectivity among nodes, as the transmit power, or the receiver sensitivity. Similarly to the case of the real word deployment, J denotes the set of nodes in the controllable testbed of size $|J| = N^{(c)}$ and $P_{jl}^{(c)}$ denotes the generic element of the generic matrix $\mathbf{P}^{(c)}$, representing the average power received by node $j \in J$ when node $l \in J$ is transmitting. Therefore, the outcome of this step will be the set of all possible connectivity matrices of the controllable testbed, each of size $N^{(c)}\text{x}N^{(c)}$. As the controllable testbed is well accessible and deployed in relatively small environments, this step can be executed through direct measurements on the field (white block in Figure 2).

- **Downscaled testbed nodes selection** For each node of the real world deployment a counterpart in the controllable testbed has to be selected with the aim of reproducing the same connectivity properties. More precisely, let us denote with $\phi(i) : I \to J$ a function mapping each node of the real world deployment to a node of the controllable testbed, i.e., for each $i \in I$, $\phi(i)$ is its counterpart in the controllable testbed and $\phi(i) \neq \phi(k) \forall i, k \in I$. In the ideal case, we would like to have $P_{ik}^{(r)} = P_{\phi(i),\phi(k)}^{(c)}$ for all pairs of nodes $i, k \in I$. Usually this is not achievable, but we want to get as close as possible to the ideal case by finding

the mapping function ϕ that minimizes the sum of the distances

$$D = \sum_{i \in I} \sum_{k \in I} \left| P^{(r)}_{ik} - P^{(c)}_{\phi(i),\phi(k)} \right|.$$

The problem can be modeled as a Rectangular Quadratic Assignment Problem (RQAP) [66], which calls for the determination of an assignment of each element of I to a distinct element of J (with $|I| < |J|$), so that the objective function, D, representing the distance between the two connectivity matrices, is minimised. More precisely, if node $i \in I$ is assigned to node $j \in J$ and node $k \in I$ is assigned to node $l \in J$, their distance, d_{ijkl}, is defined as

$$d_{ijkl} = |P^{(r)}_{ik} - P^{(c)}_{jl}| + |P^{(r)}_{ki} - P^{(c)}_{lj}|.$$

Quadratic assignment problems are among the most difficult and studied Combinatorial Optimization problems found in the Operations Research literature. RQAP is strongly NP-hard, i.e., it does not have an approximation algorithm running in polynomial time for any factor, unless $P = NP$ [67]. We introduce a set of $|I|$x$|J|$ binary decision variables

$$x_{ij} = \begin{cases} 1 & \text{if node } i \in I \text{ is assigned to node } j \in J \\ 0 & \text{otherwise.} \end{cases}$$

RQAP can be formulated as the following Integer Program with a quadratic objective function.

$$\min D = \sum_{i \in I} \sum_{j \in J} \sum_{k \in I : i < k} \sum_{l \in J} d_{ijkl} x_{ij} x_{kl} \tag{3.2.1a}$$

$$\text{s.t.} \sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \tag{3.2.1b}$$

$$\sum_{i \in I} x_{ij} \leq 1 \qquad \forall j \in J \tag{3.2.1c}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in I,\ j \in J \tag{3.2.1d}$$

Constraints (3.2.1b) impose that each node in I is assigned to a node of J and constraints (3.2.1c) impose that each node of J is assigned to at most one node of I. For the sake of completeness, it should be noted that more sophisticated models and ad-hoc algorithms for quadratic assignment problems can be found in the literature, but their investigation is out of the scope of this work and the method described here was adequate to tackle the instance of the problem of interest. According to preliminary computational experience, model (3.2.1) turned out to be not solvable in reasonable computing time by state of the art quadratic programming solvers for the size of the instances arising in practical cases. Therefore a standard linearized integer programming model for RQAP was adopted. Consequently, a second set of decision variables is introduced:

$$y_{ijkl} = x_{ij}x_{kl} \quad (i < k \in I,\ j, l \in J).$$

and then RQAP can be written as the following 0-1 linear program:

$$\min D = \sum_{i\in I}\sum_{j\in J}\sum_{k\in I:i<k}\sum_{l\in J} d_{ijkl}y_{ijkl} \tag{3.2.2a}$$

$$\text{s.t.} \sum_{j\in J} x_{ij} = 1 \qquad \forall i \in I \tag{3.2.2b}$$

$$\sum_{i\in I} x_{ij} \leq 1 \qquad \forall j \in J \tag{3.2.2c}$$

$$x_{ij} + x_{kl} - y_{ijkl} \leq 1 \qquad \forall i < k \in I, j, l \in J \tag{3.2.2d}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in I,\ j \in J \tag{3.2.2e}$$

$$y_{ijkl} \in \{0, 1\} \qquad \forall i < k \in I,\ j, l \in J \tag{3.2.2f}$$

The new constraints (3.2.2d) link the two sets of variables, forcing $y_{ijkl} = 1$ if node $i \in I$ has been assigned to node $j \in J$ and node $k \in I$ has been assigned

to $l \in J$. Such formulation is valid since in this case all the coefficients d_{ijkl} are non negative. Thus, it is never convenient to set $y_{ijkl} = 1$, if not imposed by constraints (3.2.2d). Model (2) brings to the selection of a different subset of nodes, for each of the connectivity matrices $\mathbf{P}^{(c)}$. The final subset, identifying the downscaled testbed, is selected by comparing the values of D obtained for the different matrices $\mathbf{P}^{(c)}$, taking the case with minimum D.

- **Utilization of the downscaled testbed (Steps 4-7)** Once the subset of nodes of the controllable testbed has been identified, the resulting downscaled testbed can be used. As shown in Figure 3.4, the first step of this second part of the methodology (step 4) is to run the intended application/protocol on the downscaled testbed, in order to derive the performance metrics of interest. In case the achieved performance does not satisfy the application requirements, the designer can work on protocols optimisation and/or parameters tuning (step 5), before repeating experiments (i.e., moving back to step 4). The designer will remain in this loop (steps 4 and 5) until the desired performance is reached, and then he/she will move to step 5, running the application on the real world deployment. Results obtained on the real environment could be then compared to those achieved on the downscaled testbed (step 7), in order to check the presence of possible differences. In particular, the reasons causing unexpected network behaviors and performance could be investigated and taken into account for possible future deployments.

3.3 Implementation of the downscaling procedure

In this section it will be shown how the previously described downscaling procedure can be implemented, using Flextop as controllable testbed and two smart city real world deployments. The testbeds are first described and then the different steps identified in Chapter 1 are detailed. When downscaling procedure is performed and described in details, the final step will be comparing results obtained on the real world deployment with those achieved on the downscaled testbed, running different sample applications that will be described in the following.

3.3.1 The controllable testbed and the real world deployments

All testbeds considered in this work, are based on TI CC2530, and therefore, IEEE 802.15.4 compliant (see section 1.5.1 for details). As controllable testbeds, previously described EuWIN facilities available at University of Bologna were used.

For what concerns the real world networks, two different testbeds were deployed, both thought for smart city applications, mounting nodes on lamp posts at 1.8 m from the ground level maintaining the same antenna orientation.

The first testbed was composed of 25 devices deployed in a district of a small town near Bologna. Nodes position and nodes identifiers (IDs) are shown in Fig. 3.5. Nodes from 1 to 24 sent their data to the final destination, that is a 3G gateway, depicted as the red square in the figure (denoted as coordinator).

The second testbed was composed of 11 nodes deployed over lamp posts as well, in a parking near the Engineering Faculty at the University of Bologna. Nodes were

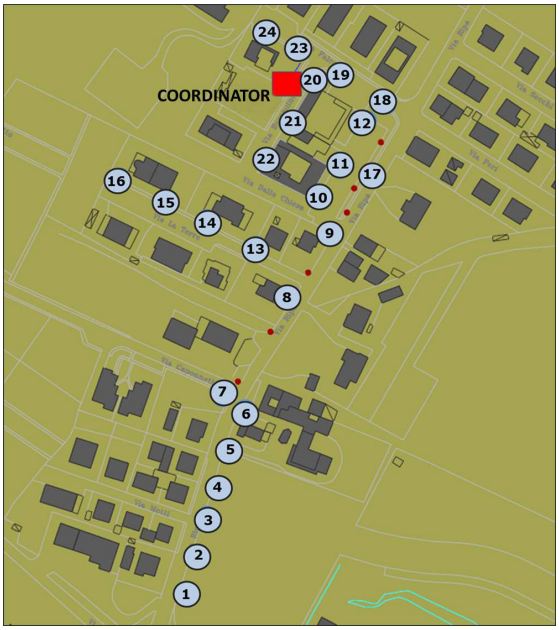

Figure 3.5: The real world deployment: nodes distribution in the district.

Figure 3.6: The real world deployment: nodes distribution in the parking.

Figure 3.7: The real world deployments.

distributed in two parallel lines as shown in Fig. 3.6.

In both the above cited testbeds some preliminary point-to-point measurements were performed, in order to find out a proper value of transmit power to be set, that should be sufficiently high to avoid the presence of isolated nodes (nodes without neighbours). On the other hand, the aim was also to avoid to have a simple star topology, whose performance can be reproduced more easily. As a result of these measurements and considerations, 5 dBm was set as the transmit power in the parking and 20 dBm in the district.

3.3.2 Implemented steps

3.3.2.1 Step 1: Estimation of the connectivity in the real world deployment

The matrix $\mathbf{P}^{(r)}$ of size $N^{(r)}$x$N^{(r)}$, where $N^{(r)}$ is the number of nodes in the real world deployment, has been obtained through direct measurements on the field, as described in the following. Nodes, one at the time, transmit a burst of 10 000 packets while all other nodes are in the reception mode, measuring the received power and averaging over the total number of packets received from the specific transmitter. When all nodes finish transmitting their burst of packets, they report an array, containing the average received power by each transmitter, to the coordinator. In this way, the matrix $\mathbf{P}^{(r)}$ is created and extracted from the coordinator.

Two nodes are considered to be connected if the percentage of packets received over the link is larger than 90%. Therefore if more than 90% of packets are correctly received over the link, we compute the average received power, which is included in the matrix; otherwise, the two nodes are considered as not connected.

Table 3.1: Average received powers [dBm] matrix for the real world deployment in the parking.

Node ID	1	2	3	4	5	6	7	8	9	10	11
1	-	-61	-71	-78	-80	-84	-85	-75	-65	-62	-63
2	-61	-	-68	-66	-75	-73	-67	-62	-56	-67	-63
3	-70	-67	-	-59	-79	-78	-60	-65	-61	-69	-76
4	-76	-66	-60	-	-70	-74	-60	-60	-67	-71	-72
5	-76	-76	-79	-68	-	-80	-65	-74	-83	-84	-
6	-86	-74	-82	-75	-77	-	-69	-85	-74	-83	-83
7	-	-65	-59	-57	-65	-68	-	-63	-65	-83	-86
8	-73	-62	-69	-61	-73	-85	-66	-	-59	-66	-75
9	-65	-56	-63	-66	-81	-75	-66	-58	-	-69	-69
10	-60	-66	-70	-70	-85	-85	-82	-65	-68	-	-67
11	-65	-64	-78	-74	-	-82	-	-75	-70	-69	-

The measured matrix, $\mathbf{P}^{(r)}$, for the case of the testbed composed of 11 devices and deployed on the parking is reported in Table 3.1, where the different values are expressed in dBm and where the symbol "-" indicates the elements of the diagonal and also the not connected links. The matrix $\mathbf{P}^{(r)}$ for the case of the 25 devices testbed deployed in the district is not provided for the sake of conciseness.

3.3.2.2 Step 2: Measurement of the connectivity in the controllable testbed

Four different connectivity matrices $\mathbf{P}^{(c)}$ have been measured by setting different levels of transmit power at nodes: $\mathbf{P}_1^{(c)}, .., \mathbf{P}_4^{(c)}$. The matrices have been obtained in the same way as described in the previous step. The matrices related to the following set of transmit powers were measured: [0,5,10,15] dBm and each matrix had a size $N^{(c)} = 53$, that is the total number of available nodes in the controllable testbed, Flextop. The latter matrices are not included in the thesis for the sake of conciseness.

3.3.2.3 Step 3: downscaled testbed nodes selection

At this step the procedure described in Section 3.2.2 was applied. Since reproducing the pairs of nodes that cannot directly communicate is considered as very important for practical application, elements of $\mathbf{P}^{(r)}$ and $\mathbf{P}^{(c)}$ corresponding to "no connectivity" were set to a large negative value (i.e., -1000 dBm) and the elements of the diagonal equal to "0", that is $P_{ii}^{(r)} = 0$ for all $i \in I$ and $P_{jj}^{(c)} = 0$ for all $j \in J$. The commercial Mixed Integer Programming solver IBM Cplex 12.5 [68] has been used to solve model (3.2.2) on a workstation equipped with an Intel Xeon E3-1220 3.1GHz CPU. However, this is not the only way to approach the problem: heuristic methods (e.g., Simulated Annealing, Tabu Search) can be used to obtain good quality solutions in a reasonable computing time for networks with hundreds of nodes (see [69, 70]).

As described in the previous section, four instances of the problem, corresponding to different transmit power of the Flextop controllable testbed, were considered.

Table 3.2 reports the set of identifiers (IDs) of the selected nodes for the downscaled testbed in the case of the parking, for the different levels of transmit power and obtained by running the algorithm described in Section 1.4.2. For each set of identified nodes the corresponding value of the objective function D is reported. As can be seen, the value of D is very large for the case of 10 and 15 dBm. Using higher transmit power reduces the number of pairs of nodes that cannot communicate and, as a consequence, the flexibility allowed in the selection of the nodes. Recall that if there is no connectivity between two nodes i and k of the real world deployment (i.e., $P_{ik}^{(r)} = -1000$ dBm), they can only be mapped to a pair of nodes j and l of the controllable testbed with no connectivity (i.e., $P_{jl}^{(c)} = -1000$ dBm), otherwise a large distance value is charged in the objective function. For the latter reason the cases of 10 and 15 dBm are not considered in the experiments but only the cases of 0 dBm and 5 dBm. Numerical results demonstrate that 0 dBm is the best solution, bringing to the lowest reproducibility error, allowing to conclude that the set characterised by the smallest objective function obtained is the best one. Table 3.3 reports the

Table 3.2: Mapping of devices: nodes deployed in the parking.

Real Deployment IDs	Controllable Testbed IDs: 0dBm	Controllable Testbed IDs: 5dBm	Controllable Testbed IDs: 10dBm	Controllable Testbed IDs: 15dBm
1	14	5	17	47
2	18	23	31	33
3	30	28	47	36
4	28	36	53	30
5	40	53	39	7
6	43	44	14	23
7	36	38	50	15
8	27	18	36	42
9	23	31	28	40
10	2	15	12	39
11	10	9	5	35
D	550	617	792	1276

mapping of the IDs of the real deployment in the case of the district and the corresponding selected nodes in the controllable testbed. The lowest distance, D, between connectivity matrices has been obtained by setting a transmit power of 0 dBm in the controllable testbed. The coordinator in the downscaled testbed was located in the red square shown in Fig. 3.

Table 3.3: Mapping of devices: nodes deployed in the district.

Real Deployment IDs	Controllable Testbed IDs	Real Deployment IDs	Controllable Testbed IDs
1	48	13	18
2	45	14	16
3	46	15	15
4	38	16	19
5	40	17	14
6	35	18	7
7	34	19	17
8	33	20	3
9	6	21	21
10	23	22	22
11	2	23	5
12	26	24	13

3.3.2.4 Performance Comparison: Numerical Results

In this section the implemented protocols and application are first briefly described, and then the comparison of results obtained on the real world deployments and on the downscaled testbed is presented. In the experiments that were performed, nodes of the controllable testbed selected through the previously described procedure were used, while remaining nodes were turned off. In the case of deployment on the parking, the set of nodes identified in Table 3 when setting the transmit power equal to 0 dBm and 5 dBm was considered.

3.3.2.5 Implemented protocols and applications

For what concerns the routing protocols, the following were considered i) a simple multi-hop protocol thought for linear wireless networks, ii) Zigbee [71] using the default routing protocol, (AODV) [27] and iii) Zigbee using Many-To-One (MTO) [71], both described in details in Section 1.5.2.1.

With reference to point i) nodes were deployed over a line, transmitting data to a final destination located at the end of the line. This is a typical scenario in smart city applications where nodes are mounted on lamp posts and transmit data to a final sink located at the end of the street. The protocol imposes that each node sends its packet to the next hop in the line in order to reach the final destination. Both LWN, and LWSN applications were investigated. AODV routing was deployed for the case of LWN application, while nodes were running MTO routing in the case of LWSN application.

3.3.2.6 Performance comparison: real world deployment in the parking

In this section results obtained in real deployment in the parking are compared with those obtained in the downscaled testbed.

The first set of results is related to measurements obtained when setting 0 dBm as transmit power in the downscaled testbed, bringing to the best solution in terms of fitting with results in the real world deployment. While in the second part results related to the sub-optimal selection of nodes for the downscaled testbed, obtained when setting 5 dBm as transmit power on the controllable testbed are reported. Results demonstrate that the case of 0 dBm is the best, as expected since it is characterised by a lower distance D between the connectivity matrices (see Table 3.2).

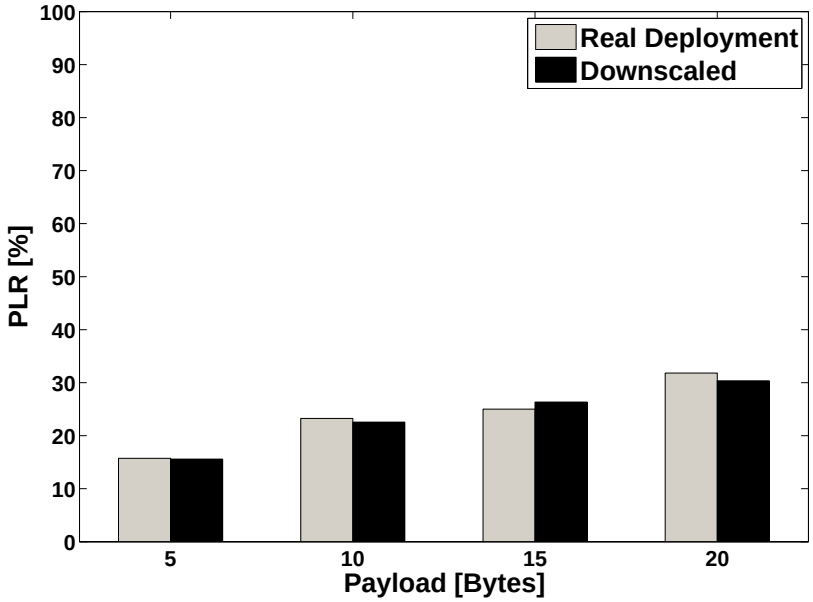

Figure 3.8: PLR as a function of the payload size in the case of 4 hop LWN (parking real world deployment).

Performance have been evaluated in terms of PLR and throughput; in all cases source node(s) generated 10 000 packets, considering different values of payload size.

First, the performance of protocols for the case of linear topology, that is considering a single line, is evaluated. In the case of the real world deployment in the parking nodes from 6 to 11 were setup, node 11 acting as the final destination (see Fig. 3.6). The cases of 4 and 5 hops were considered, where in the case of 4 hops the last node in the line (i.e., node 6) was eliminated. For what concerns the downscaled testbed, the corresponding nodes (see Table 3.2) were setup, that is nodes 43, 36, 27, 23, 2 and 10, being 10 the final destination. Nodes position of selected nodes in the controllable testbed for the case of 0 dBm transmit power are shown in Fig. 3.2.

Both, LWN and LWSN application were considered. In both scenarios the first node in the line (node 6 in the case of 5 hops and node 7 in the case of 4 hops)

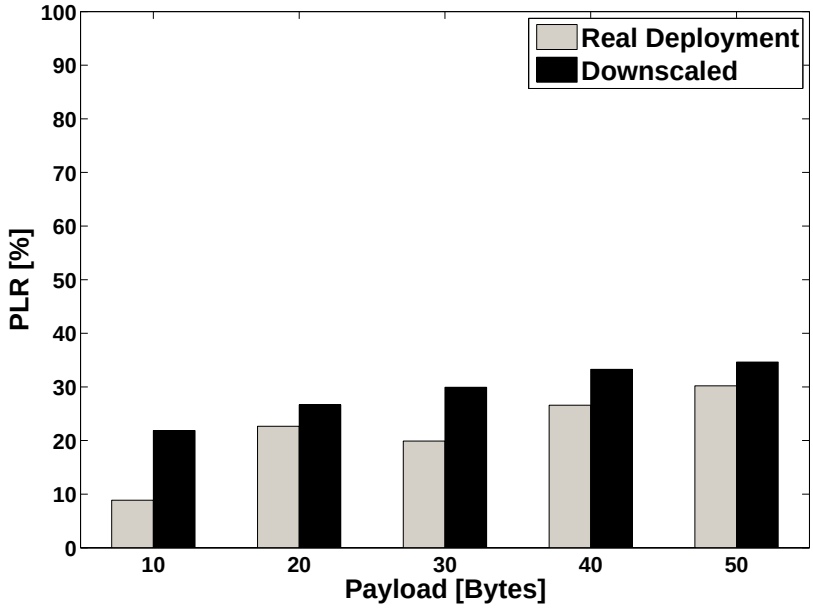

Figure 3.9: PLR as a function of the payload size in the case of 5 hop LWN (parking real world deployment).

generated one packet of given payload whenever it received the acknowledgement from the subsequent node in the line.

Fig.s 3.8 and 3.9 show the PLR in case of LWN application for 4 and 5 hops, respectively, considering different payload sizes. It can be seen that the PLR increases with the payload and number of hops, since the channel is occupied for a longer time, but also due to larger collision probability as the number of nodes increases. For the latter reason the throughput (shown in Fig. 3.10) decreases with the number of hops considered. On the other hand, the throughput increases with the payload size, since more useful information for the same header size is transferred. A good agreement between results obtained on the two networks is obtained.

Fig. 3.11 shows the PLR for the case of LWSN application for 4 and 5 hops, while

Table 3.4: Relative error [%].

Application	Relative error for PLR	Relative error for Throughput
LWN	16	3.7
LWSN	29	3.9

Fig. 3.12 shows the throughput for the same scenario. It can be seen that the trend remains the same as in the case of LWN application, as well as that results obtained in both networks (real deployment and the downscaled network) match very well.

In order to quantify the reproducibility error obtained in different cases, the relative error, defined as the ratio between the absolute error and the mean error, was measured. The absolute error is defined as the magnitude of the difference between the exact value and the approximated value. In this case the exact value, x_r, corresponds to the value obtained on the real deployment, while the approximated value, x_d, corresponds to the value obtained on the downscaled testbed. Therefore, the absolute error is given by $\Delta x = x_r - x_d$, while the mean error is defined as $\bar{x} = \frac{x_r + x_d}{2}$. Finally, the relative measurement error, indicating the offset between measurements performed in two networks, can be written as: $\delta_x[\%] = \frac{|\Delta x|}{\bar{x}} \cdot 100$ [72].
The relative error, $\delta_x[\%]$, measured for the PLR and throughput for the cases of LWN and LWSN is shown in Table 3.4. In all cases the relative error has been obtained by averaging results related to the different number of hops and payloads. As can be seen the error increases when passing from LWN to LWSN, as expected due to the complexity of the traffic generated. In the case of Zigbee network, all the 11 nodes deployed in the parking were setup. The coordinator was located in position 1 in the

Table 3.5: PLR for the case of Zigbee network (parking real world deployment).

Payload size [Bytes]	Downscaled Testbed	Real Deployment
10	1.5	2
20	1.5	2
30	1.5	2.5

real world deployment (see Fig. 3.6) and in position 14 in the downscaled testbed (see Fig. 3). The traffic toward the coordinator was generated periodically (every 10 ms), by just one node, that is node 5 in the real deployment, and the corresponding one in the downscaled testbed. Paths were selected according to AODV and according to Zigbee [71] refreshed in case of link failures. Figure 3.13 and Table 3.8 show the throughput and the PLR obtained. It can be seen that the PLR was close to zero in both networks, while the throughput increases with the payload size, for the previously described reason. As in the case of the linear topology, results obtained are very similar, even though a more complex mesh topology is considered in this case.

Table 3.6 reports the same set of results when the transmit power in the downscaled testbed is set to 5 dBm. Although the trend is maintained and well reproduced as in the previous set of measurements, the offset between results achieved in two testbeds is larger in this case.

To better demonstrate the latter, and to show that the proposed methodology brings to a satisfying level of reproducibility, we show in Table 3.7 the relative error obtained when considering the two sets of nodes identified for the two different transmit powers. The relative error is derived for the PLR and the throughput, by averaging among results related to the different payloads and number of hops. It can be seen

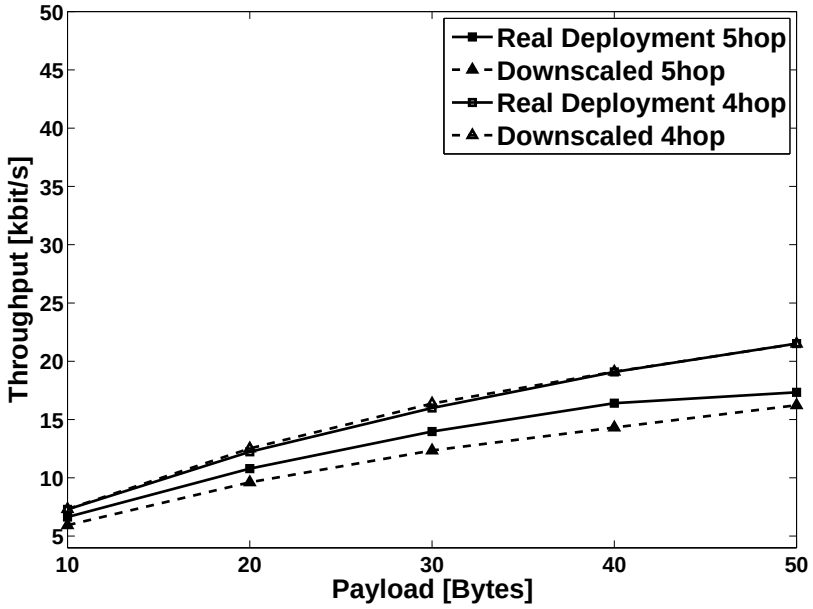

Figure 3.10: Throughput as a function of the payload size for the LWN case (parking real world deployment).

that the relative error is much smaller when we use the subset of nodes for which the minimum distance was found (corresponding to 0 dBm transmit power). The latter proves that the distance among the average received power matrices represents a valid and trustworthy metric to be used for reproducibility purposes.

3.3.2.7 Performance comparison: real world deployment in the district

In this section results obtained in the district deployment near Bologna are compared with those achieved on the downscaled testbed identified in the previous section. The description of the downscaled testbed identification for this case is omitted, given the fact that was described in details for the real world deployment in the parking case.

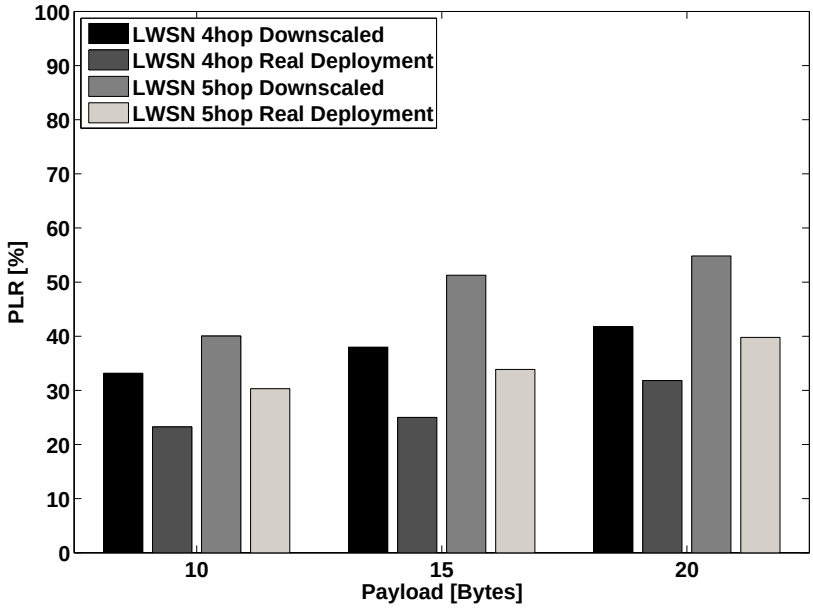

Figure 3.11: PLR as a function of the payload size in the case of 4 and 5 hops LWSN (parking real world deployment).

Table 3.6: Comparison between real world deployment in the parking and downscaled testbed results for the case of 5 dBm transmit power.

Payload [bytes]	**Throughput [kbit/s]**				**PLR [%]**			
LWN	Real Deployment 5 hop	Downscaled 5 hop	Real Deployment 4 hop	Downscaled 4 hop	Real Deployment 5 hop	Downscaled 5 hop	Real Deployment 4 hop	Downscaled 4 hop
10	6.63	5.67	7.29	8.96	8.87	34.44	4.75	4.89
20	10.79	9.35	12.23	14.59	22.65	39.06	7.19	14.29
30	13.96	12.07	15.994	19.28	19.9	41.9	10.97	18.47
40	16.4	14.13	19.08	24.66	26.29	45.61	12.72	14.73
50	17.33	15.48	21.51	27.96	30.2	49.21	15.77	15.45
LWSN	Real Deployment 5 hop	Downscaled 5 hop	Real Deployment 4 hop	Downscaled 4 hop	Real Deployment 5 hop	Downscaled 5 hop	Real Deployment 4 hop	Downscaled 4 hop
5	13.55	13.3	13.9	12.88	19.19	39.2	15.73	16.02
10	21.7	21.15	23.13	20.27	30.31	45.71	23.25	27.43
15	28.28	25.86	30.64	25.62	33.87	52.61	25.02	35.63
20	31.1	28.25	35	29.93	39.78	59	31.81	37.89

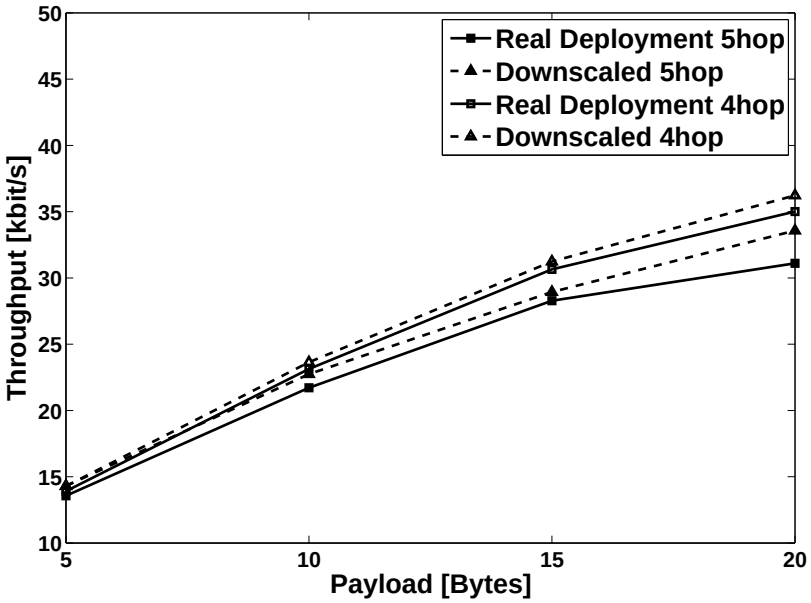

Figure 3.12: Throughput as a function of the payload size for the LWSN case (parking real world deployment).

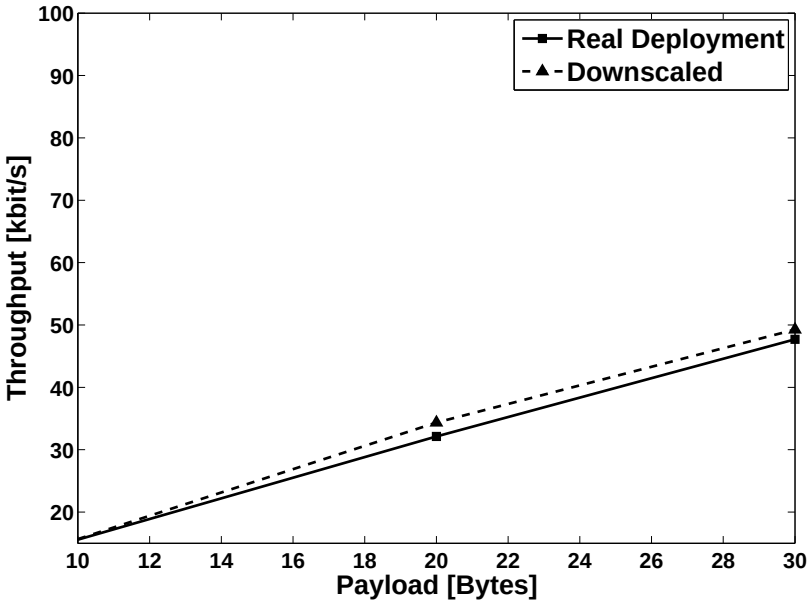

Figure 3.13: Throughput as a function of the payload size for the case of Zigbee AODV routing (parking real world deployment).

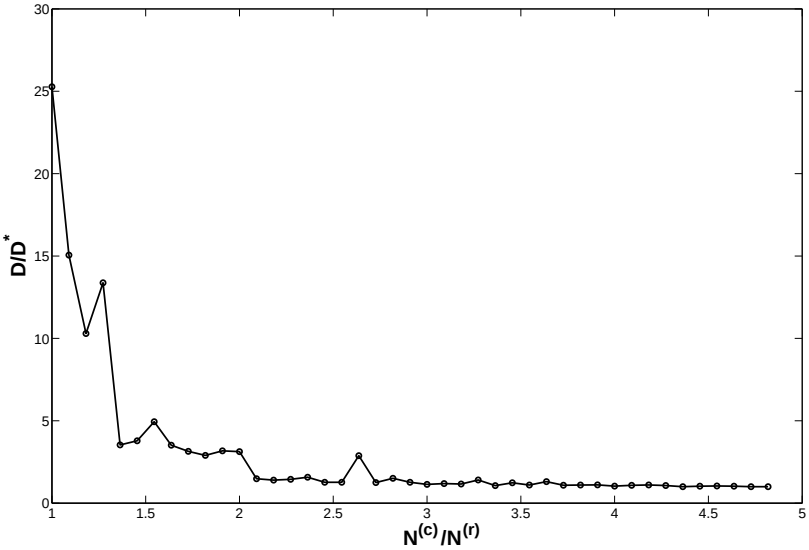

Figure 3.14: Evaluation of the minimum ratio $N^{(c)}/N^{(r)}$.

Table 3.7: Relative error [%].

Transmit Power	Relative error for PLR	Relative error for Throughput
0 dBm	30	5.2
5 dBm	45	14.2

In the reference application, the coordinator sends every minute a service request (i.e, query) to nodes in the network, which then perform a measurement of the level of luminosity generated in the environment (payload of 10 bytes), and they send back a reply to the coordinator. Before each query transmission, a MTO-RR is sent by the coordinator, in order to establish new routes, used for data transmission that follows. Results are compared in terms of i) Packet Loss Rate (PLR), that is the percentage of packets generated by the sources and not correctly received by the final destination (i.e., the network coordinator); ii) number of hops needed to reach the coordinator, averaged among the different nodes and iii) average probability that a

node is used as relay by other nodes, to reach the coordinator. During measurements every minute the different topologies formed and the number of data packets received by the coordinator were stored. 10 000 topologies and queries/replies have been generated and results have been averaged among these transmissions. As can be seen from the Table 3.8, exactly the same performance has been obtained in the two testbeds. Network throughput, that is the number of information bits (i.e., bits of the payload) per second correctly received by the coordinator, was not computed in this case, since having the PLR close to zero and data generated periodically, brings to the constant network throughput and the same for the case of the two testbeds.

Table 3.8: Dowscaling of the real world deployment in the district.

Metric	Real world deployment	Downscaled testbed
PLR	$< 10^{-2}$	$< 10^{-2}$
Average number of hops	1.41	1.42
Average probability of being a relay	0.041	0.041

3.3.3 Further results and discussion

In the following, the ratio among number of nodes available in two (controllable in the real) testbeds is discussed.

Results regarding the analysis of the ratio among number of nodes available in the controllable testbed $N^{(c)}$ and the number of nodes in the real deployment network, $N^{(r)}$, needed to properly reproduce the behavior of the real network is reported. The intuition suggests that the larger the number of nodes in the controllable testbed, the higher the probability of correctly reproducing the property of the real deployment network, i.e., a small value of D. In order to gain more insight on this, experiments

when considering different number of nodes in the controllable testbed and the real deployment in the parking composed of 11 devices, were performed. More precisely, subsets of k nodes of the controllable testbed ($k = 11, .., N^{(c)}$) were randomly selected, and then the corresponding optimization problems for the selection of the downscaled testbed nodes was solved. The plot in Figure 3.14 shows the behaviour of D/D^* (where D^* is the distance measured when the maximum number of nodes in the controllable testbed $N^{(c)}$ is used) as a function of the ratio $N^{(c)}/N^{(r)}$ when the devices are transmitting at 0 dBm. The results confirm the intuition and suggest to use a controllable testbed with at least two or three times more nodes then in the real deployment network ($\frac{N^{(c)}}{N^{(r)}} \geq 2$).

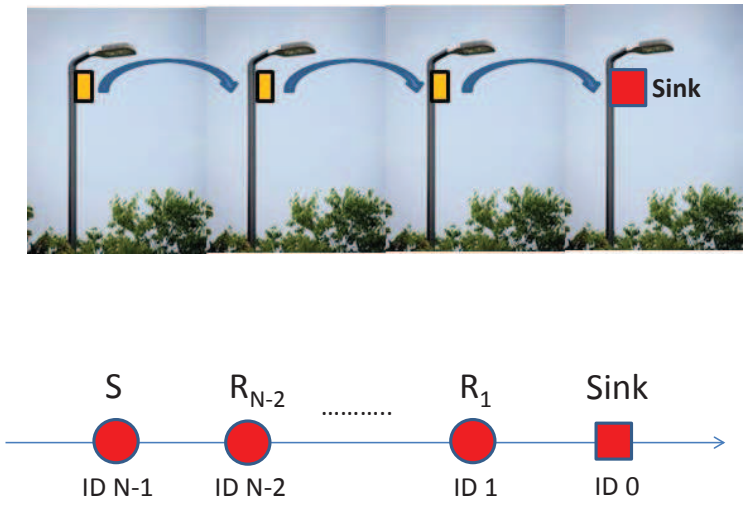

Figure 3.15: Reference scenario: Smart Lighting System.

3.4 Protocol Optimisation: Smart Lighting System

Once the downscaling procedure is performed, and suitable nodes for a given application are selected, various experiments dealing with protocol optimisation can be carried out. In this section, a smart lighting application is addressed. Nodes for experiments were selected according to previously described and validated downscaling procedure.

A network of N statically deployed (see Fig. 3.15) nodes is considered: the last node in the line acts as network coordinator/sink. $N-2$ relays (R_1,..,R_{N-2}) and one source S are deployed. Both, LWN and LWSN applications are investigated. A novel, broadcast-based routing protocol, minimising the overhead and maximising the network throughput, is proposed. According to this protocol, each node identifies,

during a neighbour discovery phase, the best next hop to be used in order to reach the sink. Each node, therefore creates an optimal routing table and, as the network is static, an excellent performance is guaranteed. The proposed solution has been tested in EuWIn laboratory, using Flextop and mobile DataSens nodes. The proposed solution is compared to the following benchmark solutions: i) Hop-by-Hop: a simple solution where nodes transmit to the next hop in the line (i.e., direct neighbour) toward the sink; ii) Single-direction Broadcast: each node, receiving a packet from a relay behind it in the line, rebroadcasts it; iii) Irresponsible Forwarding (IF) and iv) Zigbee: using AODV [27, 71]. The observed results show that the proposed protocol outperforms other considered solutions.

3.4.1 The Proposed Protocol

The proposed protocol requires that during the association phase the sink, acting as coordinator, assigns to nodes addresses (i.e., identifiers (IDs)), corresponding to their position in the line. This can be simply obtained by switching on nodes one by one, starting from the node nearest to the sink to which the sink assigns ID 1, ending with the farthest (to which the sink assigns ID $N-1$). In Figure 3.15 an example of IDs assignment is shown.

The proposed protocol fundamentally involves two phases (when the association phase is completed): i) a neighbour discovery (ND) phase, whose flow chart is shown in Figure 3.16, and ii) a data transmission (DT) phase, whose steps are shown in Figure 3.17.

At the end of the association phase, nodes perform the ND phase. The goal of this phase is to allow each node to find its own best neighbour, which is the closest

to the sink. The ND phase for each node starts by setting a timer to a value T_{ND} from the time instant in which the node receives the first ND packet. This does not hold for the source node, which starts transmitting ND packets immediately. When the timer expires, the node transmits five ND packets in broadcast. ND packets are transmitted every 10 ms, they have 5 bytes payload and they contain the ID of the transmitting node. This allows reducing collisions among ND broadcast packets of the different nodes. During the ND phase, each node receiving the ND packet performs the following steps: i) check the ID of the transmitting node, ii) compare the ID contained in the ND packet with its own ID and with the ID of the current best neighbour. If the ID contained in the ND packet is lower than its own and of the current best neighbour, the node updates the address of its best neighbour to the ID contained in the ND packet. Data transmission takes place after the ND phase according to the following procedure. When a node generates a data packet, it transmits it in broadcast, embedding its previously selected best neighbour ID into the packet. When a relay receives a packet it checks the ID of the best neighbour contained in the packet: if the ID corresponds to its own address, the node retransmits the packet immediately, including its own best neighbour ID; otherwise, it waits for a period T_{listen} in order to check if the selected best neighbour forwards the packet or not. During T_{listen} the node stays in listening mode: if the node having the ID included in the received packet forwards it, the node discards the packet; otherwise, once T_{listen} expires the node forwards the packet, including its own best neighbour in the payload.

In the case of LWSNs, a simple form of data aggregation (DA) or data concatenation [73] was implemented as follows. Each relay, before sending its own data packet,

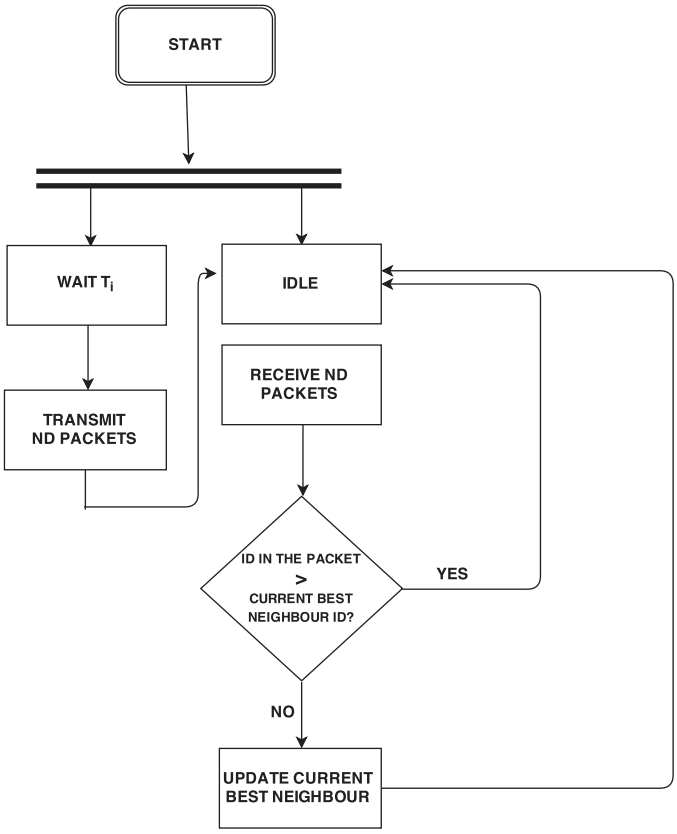

Figure 3.16: Neighbor discovery phase flow chart.

waits for a period T_{data}: if the relay during T_{data} receives a data packet to be forwarded toward the sink it appends its own data to the payload of the received packet before forwarding it. If no packets are received from preceding nodes in the line, the relay generates a new packet including its data.

3.4.2 Benchmark Protocols

As previously stated, the considered benchmark solutions are: i) Hop-by-Hop (HBH); ii) Single-direction Broadcast (SDB); iii) Zigbee protocol using AODV and iv) Irresponsible Forwarding.

As in the case of the proposed protocol, in all the considered benchmark solutions, but Zigbee, it is requested that during the association phase the sink assigns to nodes addresses, corresponding to their position in the line. This can be obtained by implementing the association procedure as described in Section II. Only the case of Hop-by-Hop is applied to the LWSN application, since the other protocols demonstrate strong performance degradation. The number of retransmissions in all protocols considered was set to zero.

In the Hop-By-Hop protocol, each node transmits each data packet (generated by the node behind it in the line) to the node in front of it. In the case of LWSN, we also apply data aggregation, by imposing to all relays to append their generated data to the payload of the packet they have just received from the preceding node.

In the case of Single-direction Broadcast, all nodes simply rebroadcast the received packet, considering only packets arriving from nodes behind in the line, like in the case of IF protocol. For what concerns Zigbee, the detailed description of AODV routing protocol that is used is given in Section 1.5.2.1.

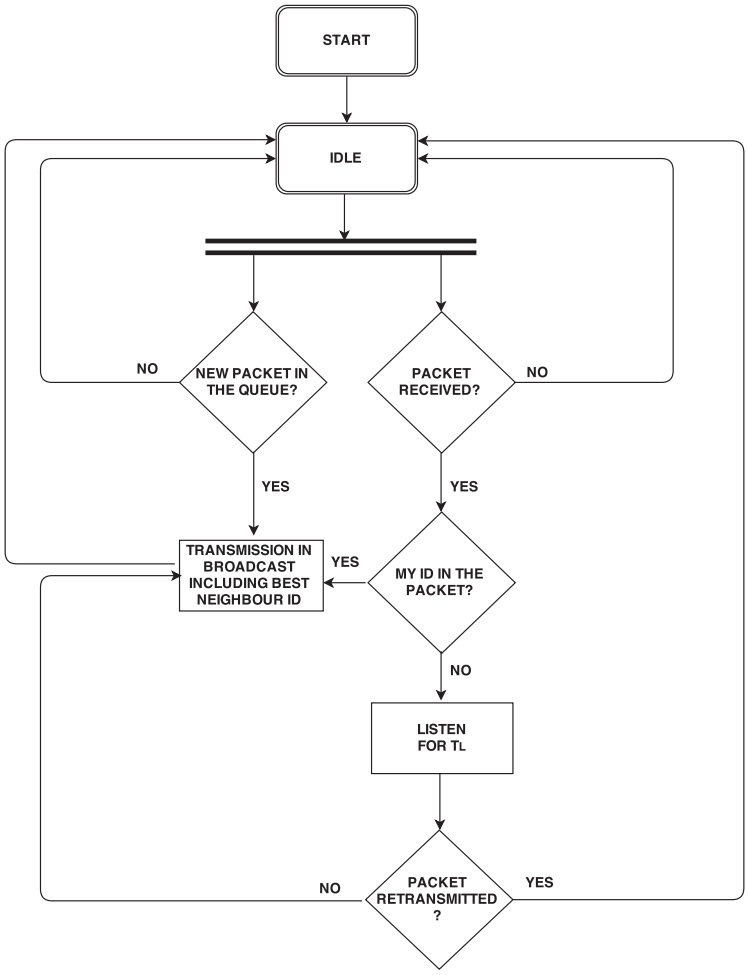

Figure 3.17: Data Transmission phase flow chart.

IF is a probabilistic forwarding protocol such that every node, upon reception of a packet to forward, computes its own retransmission probability in a per-packet manner. The broadcast forwarding process of IF can be summarized as follows. A source node transmits a packet: this first transmission is denoted as the 0-th hop transmission. The packet is then received by all nodes which are in the transmission range of the source, i.e., the source neighbours. All these neighbours are potential rebroadcasting nodes for the 1-st hop and each of them rebroadcasts the packet independently with a probability computed according to a proper Probability Assignment Function (PAF). Once the packet is forwarded by some nodes in the first hop, other nodes rebroadcast according to the same PAF and the process repeats recursively.

The choice of the PAF of IF is based on the intuitive observation that the farther the potential rebroadcaster is from the transmitter, the higher its associated rebroadcast probability should be, as this would yield the highest forward progress. Based on this idea, in [74] the PAF of IF is introduced for a monodimensional scenario (e.g., a narrow street):

$$p = \exp\left\{-\frac{\rho(z-d)}{c}\right\} \tag{3.4.1}$$

where: d (dimension: [m]) is the distance between a transmitting node and a potential rebroadcaster; z (dimension: [m]) is the transmission range; c is a shaping coefficient (adimensional), which can be used in order to tune the retransmission probability [75]; ρ (dimension: [nodes/km]) is the node linear density. According to the PAF in (3.4.1), if the network is sparse, the overall retransmission probability is high in order to ensure complete connectivity. On the other hand, if the network is dense the overall retransmission probability is low in order to reduce useless redundant transmissions and, thus, collisions.

The idea behind the IF rebroadcast paradigm is that once a node receives a packet, it evaluates, in an average statistical sense, the presence of other nodes in its proximity. If the probability that another node can rebroadcast the packet is sufficiently high, then the node of interest "irresponsibly" chooses not to rebroadcast. In order to avoid harmful flooding of the network, packets coming from nodes that are in front of the given node in the line, are not considered for rebroadcasting. IF is based on the assumption of the knowledge of some topological network parameters, such as internode distance and node linear density—this is realistic in a LWSN scenario where nodes are static.

3.4.3 Experimental Setup

The Flextop platform (denoted as *short corridor* in the following), shown in Figure 3.2 was used. The following nodes were selected for the experiments: 19, 22, 28, 35, 39, and 45, where node 45 is the sink, having ID 0; node 39 is R_1, with ID 1; node 35 is R_2 with ID 2; node 28 is R_3 with ID 3; node 22 is R_4 with ID 4; node 19 is S, with ID 5.

DataSens platform (denoted as *long corridor* in the following), was also used in this work, as battery equipped nodes and longer corridor provided further degree of freedom for the setup. 6 nodes (IDs from 0 to 5), were deployed on the ground into a longer corridor at the University. In this case, the distance between two consecutive nodes is 20 meters (see Figure 3.3), node 0 is the final sink and node 5 is the source, S. The matrix $\mathbf{P}$, characterising the level of connectivity among the nodes in the long corridor environment is reported in Table 3.10.

Table 3.9: Average Received Powers [dBm] Matrix: Dense Matrix case.

IDs	0	1	2	3	4	5
0	-	-60	-	-69	-81	-81
1	-63	-	-80	-71	-82	-85
2	-	-70	-	-80	-88	-
3	-78	-75	-56	-	-58	-73
4	-85	-84	-89	-55	-	-55
5	-82	-85	-	-68	-51	-

Table 3.10: Average Received Powers [dBm] Matrix: Sparse Matrix case.

IDs	0	1	2	3	4	5
0	-	-77	-85	-	-	-
1	-78	-	-76	-86	-	-
2	-85	-75	-	-80	-84	-87
3	-	-86	-79	-	-78	-81
4	-	-	-83	-81	-	-79
5	-	-	-	-83	-79	-

3.4.4 Numerical Results

Both reference applications, LWN and LWSN were considered. In all experiments, source node(s) periodically generates one data of a given payload size with a period of 10 ms and results are averaged over 10,000 packets generated. The IF protocol parameters are set as follows: $d = 20m, c = 1, z = 2.5 \cdot d, \rho = 4/5$ [nodes/m], while for the proposed protocol the parameter values were set to $T_{ND} = 2$ s and $T_{listen} = T_{data} = 10$ ms. The transmission range z was empirically determined from connectivity matrices.

The following performance metrics was considered: i) packet loss rate (PLR); ii) throughput; iii) iii) average number of hops. In order to compute the PLR, in each experiment the number N_{RX} of packets received at the sink was counted. Then, the PLR (dimensions: [%]) can be written as $PLR[\%] = (10,000N - N_{RX})100/(10,000N)$, where N in the number of sources—$N = 1$ in the case of LWN and $N = 5$ in the case of LWSN. The network throughput is evaluated by counting the average payload rate (dimensions: [bit/s]) of correctly received data by the sink. In the case of LWN, the source throughput, being data generated only by S, is evaluated. Finally, the average number of hops needed to reach the sink is measured. The latter metric is important to have a precise indication about delays, as the two metrics are strongly correlated.

3.4.5 Impact of Network Size

First, the impact of the different network sizes (i.e., number of hops) on the network throughput is shown, and the impact of having a LWN, where only the last node in

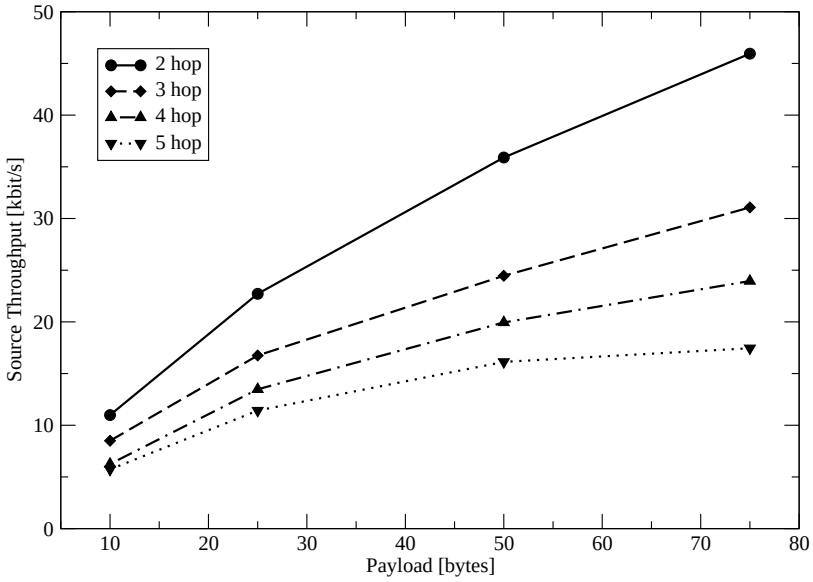

Figure 3.18: Hop-by-Hop: Throughput as a function of the payload size, for different number ‹

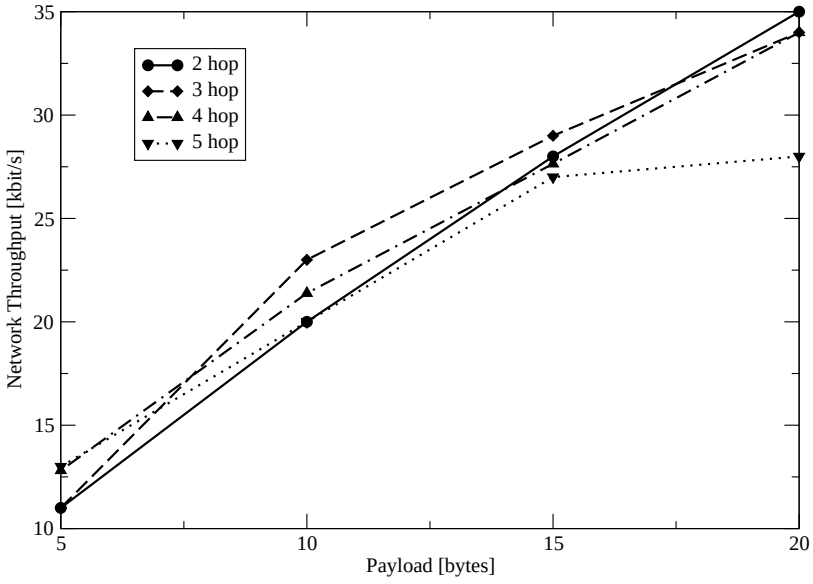

Figure 3.19: Hop-by-Hop: Throughput as a function of the payload size, for different number of hops, in the case of LWSN application.

the line generates data and the case of LWSN, where all nodes in the network are generating data. The case of Hop-by-Hop protocol is considered and data aggregation is applied in the case of LWSN and tests have been performed on the short corridor (i.e., the environment characterised by the dense matrix). Note that the same behaviours are expected also for the proposed protocol. In the case of 5 hops, all selected nodes are used (19, 22, 28, 35, 39, 45), in the case of 4 hops, node 19 is eliminated (the source will become node 22 in the case of LWN), in the case of 3 hops node 22 is eliminated (the source will become node 28 in the case of LWN), etc..

Results related to the LWN case are shown in Figure 3.18, while Figure 3.19 reports the network throughput for the case of LWSN. As can be seen in the case of LWN, the source throughput always decreases by increasing the number of hops, since the probability of loosing the packet in one of the links separating the source to the sink is larger. In the case of LWSN, instead, there is an optimum number of hops maximising the network throughput, which changes with the payload. On one hand, by increasing the number of hops, the number of generated bytes per second incrases, but, on the other hand, the packet loss rate increases as well (as shown in Figure 3.18): the trade-off between the two effects allows the throughput maximisation. In particular, it can be noticed that the optimum number of hops decreases by increasing the payload size: 4 hops for payload within (5-7.5) bytes, 3 hops for payload within (7.5-17.5) and 2 hops for payload within (17.5-20). Note that the maximum payload size is set equal to 20 bytes, due to the fact that DA is used and the maximum packet size is 100 bytes (5 nodes with payload of 20 bytes each), which almost reaches the maximum allowed in IEEE 802.15.4 [9].

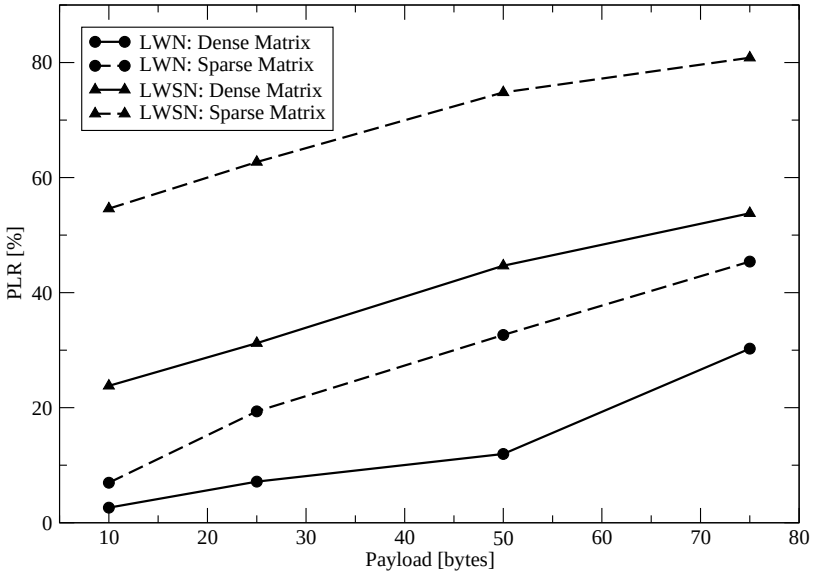

Figure 3.20: Comparing PLR for the sparse and dense matrix environments.

3.4.6 Impact of the Environment

In this section the impact of the different environments on the performance is shown. Figures 3.20 and 3.21 report results in terms of PLR and source (for the case of LWN) and network (for the case of LWSN) throughput, respectively, obtained in sparse and dense environments. As expected, in all cases results related to the dense matrix are better, since nodes perform on average less hops to reach the final sink. This is shown in Table 3.12 commented in the following. Reasonably, in all cases the throughput for the case LWSN is larger than that of the case LWN.

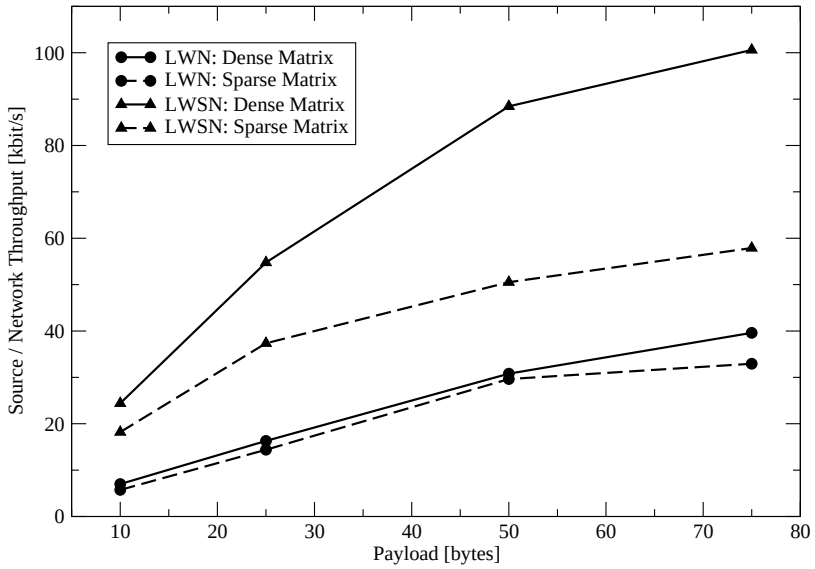

Figure 3.21: Comparing source (LWN) and network (LWSN) throughput for the sparse and dense matrix environments.

3.4.7 Comparing the Proposed Protocol with Benchmark Solution

Figure 3.22 reports results in terms of source throughput for the case of LWN.

Results of Figure 3.22 have been obtained in the short corridor, that is considering the environment characterised by the dense matrix. The proposed protocol is compared with Hop-by-Hop, Single-direction Broadcast, IF and Zigbee. Again, the proposed solution outperforms the others, but IF protocol demonstrates to perform better than standardised solutions. Therefore, both both introduced protocols seem to be promising solutions for the reference application. Figures 3.23 and 3.24 report results in terms of PLR and network throughput, respectively, for the case of LWSN.

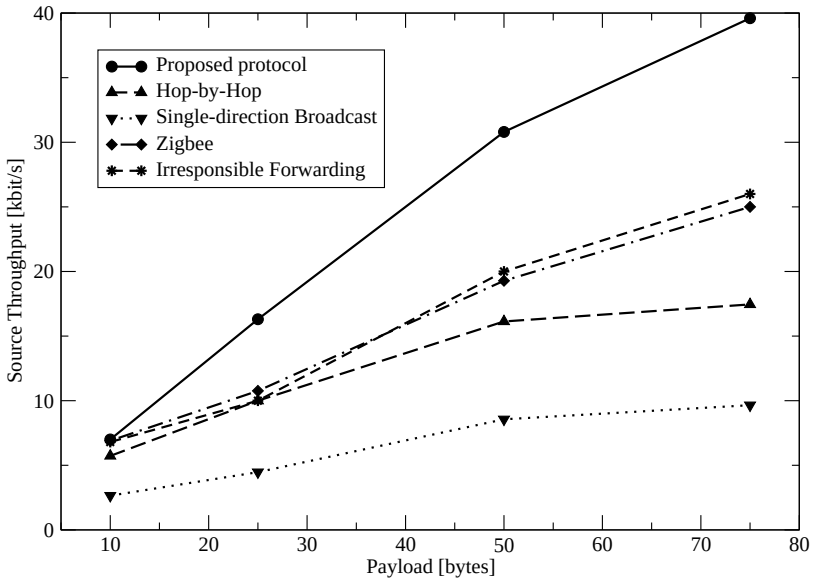

Figure 3.22: LWN: Comparing source throughput for the different solutions.

In this case results have been achieved using devices located in the short corridor. Only the case of Hop-By-Hop and the proposed solution have been considered, since the other protocols cannot be applied to the LWSN case for previously described reasons. The figure shows, on one hand the impact of using data aggregation on the proposed protocol and the corresponding improvement achieved, and on the other hand, compares Hop-By-Hop and the proposed solution when using data aggregation. Results show that the proposed solution outperforms Hop-By-Hop.

In order to justify results of the previous figure and also to provide an estimation of the average delays, the average number of hops needed by nodes to reach the sink is evaluated. As benchmark an *Ideal* solution is considered, where it is assumed that routes in the network are established by a central controller, which receives as input the connectivity matrices for the two reference cases (sparse and dense environments characterised by connectivity matrices reported in Tables 3.9 and 3.10) and computes

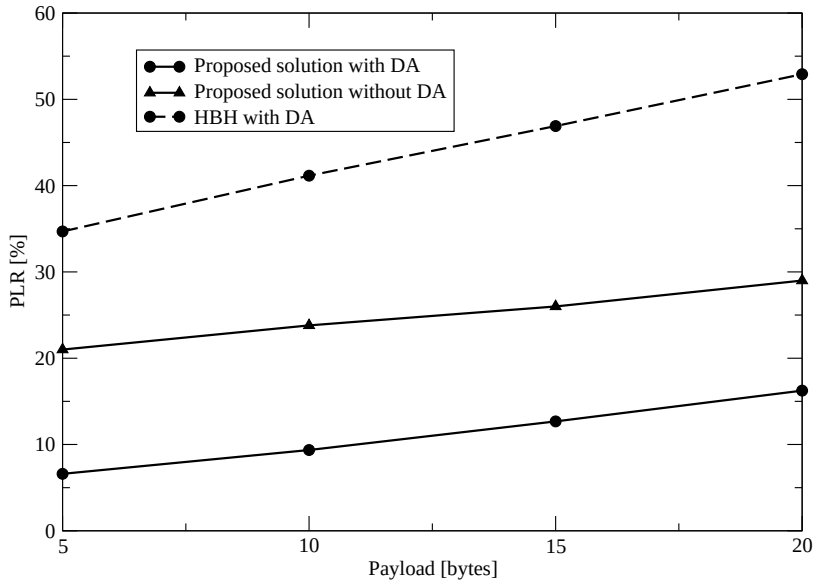

Figure 3.23: LWSN: Comparing PLR for the different solutions.

the optimum paths, by applying Dijkstra. In particular, the algorithm is applied by considering the two matrices as the adjacency matrices of the weight graph, where the vertex are nodes and edges are the links, with weights equal to the absolute value of the average power received on the link. As for the case of the proposed protocol, the topologies were derived by considering that each node select as next hop the neighbour characterised by the smallest ID. As for the case of HBH topologies are straightforward. The resulting topologies are shown in the Table 3.11 and the correspondent number of hops for the different cases in Table 3.12. The number of hops provides also an indication about delays, as it is a function directly proportional to the number of hops.

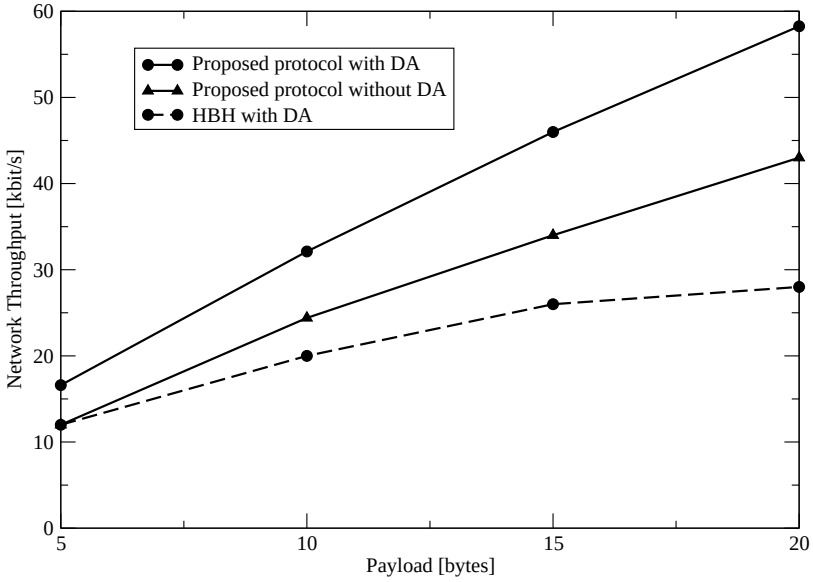

Figure 3.24: LWSN: Comparing network throughput for the different solutions.

Table 3.11: Topologies.

Source ID	Ideal, sparse matrix : Next Hop ID	Ideal, dense matrix: Next hop ID	Proposed Protocol, sparse matrix: Next Hop ID	Proposed Protocol, dense matrix: Next hop ID
1	0	0	0	0
2	0	0	0	0
3	1	3	1	1
4	2	0	2	0
5	2	0	3	0

Table 3.12: Average number of hops for the different protocols.

Protocol	**Dense Matrix LWN**	**Sparse Matrix LWN**	**Dense Matrix LWSN**	**Sparse Matrix LWSN**
Ideal	1	2	1.2	1.6
Proposed Protocol	1	3	1.2	1.8
HBH	5	5	3	3

3.5 Protocol optimisation: Smart building application

This section presents results obtained by testing and comparing three different solutions that are considered as possible enablers for the IoT and Smart City implementation. A smart building application was considered, shown in Figure 3.25. Nodes equipped with various sensors (as temperature, humidity, luminosity, motion, denoted by different colors in Figure 3.25)can be deployed inside flats for smart metering purposes. Sensors' data are collected by the flat concentrator and forwarded to a building concentrator for further processing. The work presented in this section focuses on a single flat unit. Nodes for experiments were selected according to the previously described downscaling procedure in order to emulate an environment that corresponds to a single flat unit with sensors and flat concentrator deployed. The aim of this work was to determine a type of routing protocol and network architecture that fits best the application of interest. A centralised solution based on Software Defined Network (SDN), called Software Defined Wireless Networking (SDWN), was compared with two standard and distributed solutions, that are ZigBee and 6LoWPAN. SDWN uses a centralized network layer protocol, where routing policies are defined by an external controller that can be positioned anywhere in the network. The other two solutions are actually the most common protocol stacks for wireless sensor networks, and they both use a distributed routing protocol. The comparison is achieved through experimentations performed on Flextop platform. Nodes for experiments that were carried out were selected according to previously described and validated downscaling procedure, in order to emulate a smart building application where nodes

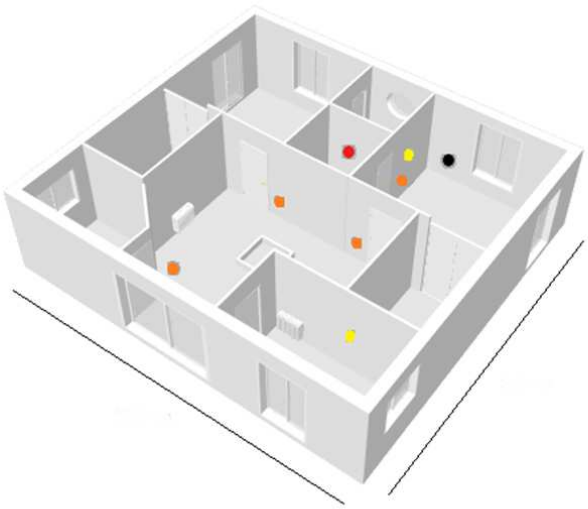

Figure 3.25: Smart building scenario.

are distributed inside the building and used for smart metering purposes, collecting data and transmitting them to the flat concentrator for the further processing. As previously mentioned and described in section 1.5, the common standards for IoT applications are ZigBee [25] and 6LoWPAN [29]. Both standards are implemented on top of the IEEE 802.15.4 standard [76]; however ZigBee uses 802.15.4 medium access control (MAC) addresses, while 6LoWPAN uses IPv6 addresses. Recently, a third approach based on the Software Defined Network (SDN) paradigm has been proposed [77]. It is called the Software Defined Wireless Networking (SDWN) and uses a centralised routing protocol. The coordinator/gateway gathers information on the status of the network of things, and brings this knowledge to a controller that can decide on the exploitation of resources within the wireless network. The controller has a centralised vision of the network of things, and can even control things that lie behind several coordinators/gateways. This approach brings the potential advantage

of optimal resource exploitation, provided that the overhead is controlled and the environment does not change too frequently.

The aim of this section is not to optimise Zigbee or 6LoWPAN, since they are considered in their standard version normally used by IoT application developers, but rather to fairly compare one protocol stack to the other and both to a novel solution - SDWN, through experiments. Results of an extensive measurement campaign evaluating different performance metrics, such as packet loss rate, round-trip-time and overhead generated in the network are presented, considering different network topologies and sizes, payload sizes and environmental conditions, from static to dynamic. Results demonstrate that SDWN achieves better performance in terms of all considered metrics in static and quasi-static scenarios. However, a severe performance degradation has been observed when the changes in network topology are frequent and significant.

3.5.1 Related Work

Few works exploit the potential of a SDN approach in wireless networks, especially in sensor networks. [78] presents the idea of exploiting the OpenFlow technology to address the reliability in WSNs. The Authors claim that OpenFlow-based sensors are more reliable than typical sensors, and simulation results show that the proposed approach achieves better performance for large networks. The use of OpenFlow in a wireless mesh network allows a rapid change of forwarding and routing algorithms [79]. A survey on challenges and opportunities in using wireless SDNs is presented in [80]. The paper claims that the SDN technology will have to face problems regarding slicing, isolation, status reporting, and handoffs, whereas it will improve, QoS, planning,

security, and localization. [81] proposes a SDN system, where experimentations show that the proposed solution reduces the energy consumption and provides a higher level of flexibility in network management.

Many research papers deal with the implementation of ZigBee networks. For example, [82–84] refer to the implementation of a ZigBee network for smart home applications. [85] measures the impact of Wi-Fi interference over ZigBee networks, while [86] evaluates the performance of a small ZigBee network (composed of less than five nodes) in terms of throughput and latency. An experimental analysis of star and tree ZigBee networks based on commercially available hardware and software is provided in [87], in order to determine the limitations of technology. Finally, [88] provides a comparison between ZigBee Pro and ZigBee IP, in terms of latency, where a network is composed of five nodes.

Refering to 6LoWPAN, [89] presents an implementation over Texas Instruments (TI) MSP430 devices. A star topology with an edge router and three nodes was deployed, and IP addressability features were tested. In [90], a novel architecture for supporting applications in the field of Intelligent Transportation Systems is presented. The implementation and evaluation of different neighbour management policies applied to RPL (Routing Protocol for Low-Power and Lossy Networks) are given in [91]; experiments were conducted on the TU-Berlin TWIST testbed with 100 TelosB motes spread over a three floor office building. In [92], the performance of the RPL protocol is evaluated by using the Lille SensLAB testbed composed of 100 TI CC2420 devices, randomly deployed in an indoor environment.

Several papers are also comparing ZigBee and 6LoWPAN: [93] provides a qualitative comparison, without addressing any quantitative evaluation of protocols' performance. In [94], the Authors present a comparative performance assessment of ZigBee and 6LoWPAN protocols for industrial applications. The testbed is composed of four TelosB nodes deployed in a linear topology.

To the best of Author's knowledge, there are no works in the literature dealing with the comparison of the SDN approach and the distributed approach represented by ZigBee and 6LoWPAN. The most important aspect, that differentiates this work from the previous ones, is that experimental results have been achieved in a controlled environment, where tests can be conducted and replicated under predictable conditions, thus making the comparison fair.

3.5.2 Considered solutions

3.5.2.1 Zigbee and 6LoWPAN

For this work, the ZigBee-Pro 2007 release specified in [25] was considered, whose protocol stack is shown in Fig. 3.26. The Home Automation profile is considered, and Many-to-One (MTO) routing, described in section 1.5.2.1, is implemented. AODV protocol (see section 1.5.2.1) was used to establish the route between the coordinator and the multicast group; in this case the RREQ packet, sent in broadcast, includes the address of the multicast group to be discovered. Nodes in the network that are linked to the target multicast group, send a RREP back to the coordinator through the selected path. The latter path is used for the transmission of query packets. In the uplink direction, that is from the queried nodes to the coordinator, nodes use the

same protocol as for the unicast transmissions, therefore MTO.

In the case of 6LoWPAN solution (see section 1.5.3 for details), RPL routing, described in section 1.5.3.1 was used. Finally, the third, centralised solution, based on SDN, will be described in the following.

3.5.2.2 The SDWN Solution

The first implementation of SDWN was developed in October 2012 [77]. The main idea behind the protocol is to adapt a centralized approach, such as the one proposed in SDN networks, to a wireless environment, thus giving the opportunity to support the flexible definition of rules and topology changes.

The SDWN protocol stack is shown in Fig. 3.26: physical (PHY) and MAC layers are those of the 802.15.4 (see Sec. VI), while upper layers are inspired by the SDN paradigm.

A typical SDWN network is composed of a *controller* device, a *sink* node, as well as several other nodes. The controller gathers the information from nodes, maintains a representation of the network, and establishes routing paths for each data flow. The sink is the only node that is directly connected to the controller, and it acts as a gateway for nodes. In our implementation, the sink coincides with the network coordinator and its protocol stack is equivalent to that of a generic node.

The stack of a generic node is divided into three parts: the Forwarding layer (FWD), the Aggregation layer (AGGR), and the Network Operating System (NOS). The MAC layer provides incoming packets to the FWD layer that identifies the type of the packet. Six different types of packets are defined:

- *Data*: generated (delivered) by (to) the application layer;
- *Beacon*: periodically sent in broadcast by all nodes in the network;
- *Report*: containing the list of neighbors of a node;
- *Rule Request*: generated when it receives a packet for handling which it has no information (i.e., the path);
- *Rule Response*: generated by the controller as a reply to the Rule Request;
- *Open Path*: used to setup a single rule across different nodes.

When a non-beacon packet is received by the FWD layer, it is sent to the NOS that searches for the corresponding rule in an appropriate data structure called *Flow Table*. The Flow Table stores all the rules coming from the controller. For each rule, there are three types of action that could be executed: forward to a node, modify the packet, or drop it. If a packet does not match any of the rules in the table, a Rule Request is sent to the controller. The path between the sink and the node for sending/receiving Rule Request/Rule Response packets must be chosen effectively, considering both reliability of the path and its length. Each node constantly stores its distance (in number of hops) from the sink, and the received signal strength indicator (RSSI) that is the level of power it receives from the next hop toward the sink. During the network initialization, each node is in a quiescent state waiting for messages. When the sink turns on, it sends a Beacon, containing the number of hops from the sink (zero in this case). When a node A receives the Beacon, it performs the following four operations:

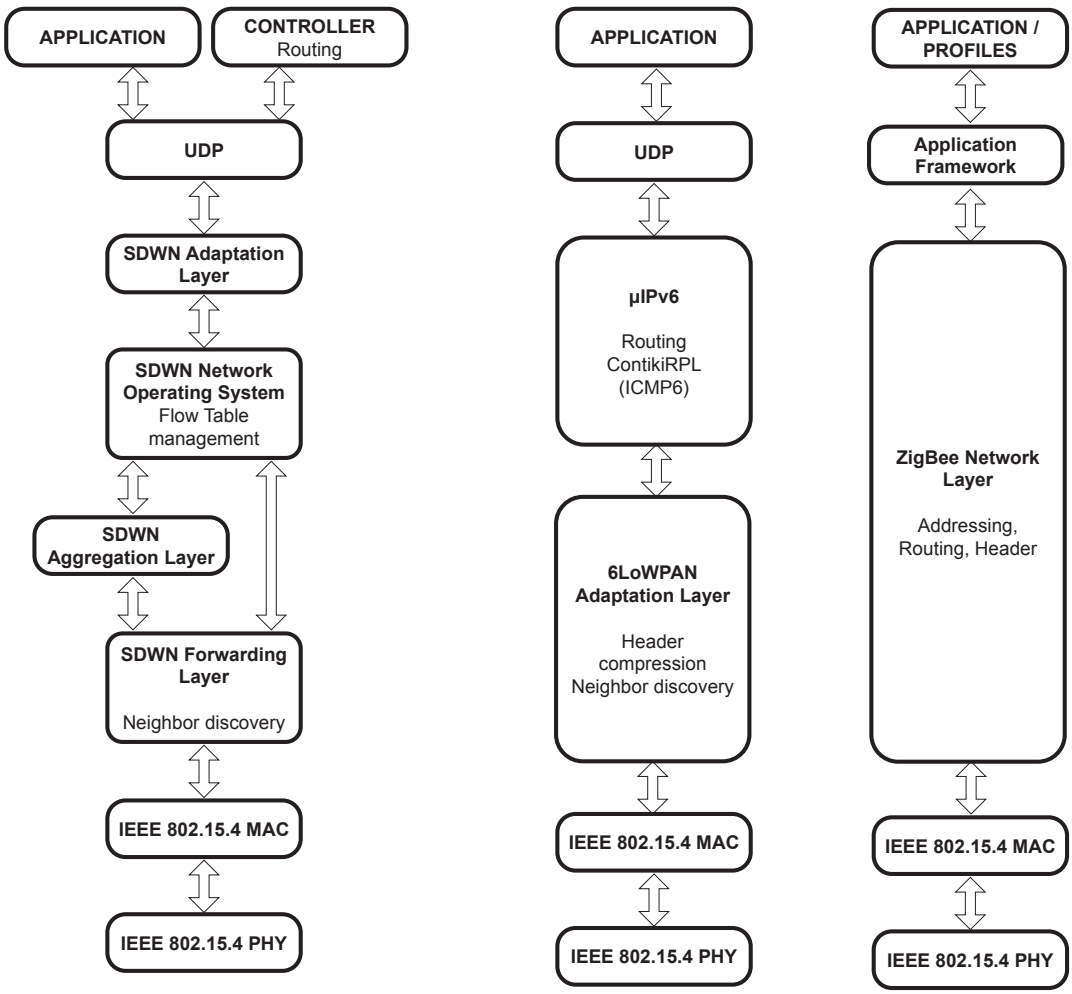

Figure 3.26: Protocol architectures: SDWN (on the left), 6LoWPAN (in the center), and ZigBee (on the right).

- Add the source of the Beacon and the RSSI received in the list of nodes (neighbours table) that are one hop distant from A.

- Analyse the distance contained in the Beacon and the RSSI of the received message, then compare these values to the corresponding stored values: if the

number of hops is lower and the RSSI is higher, the source of the Beacon is elected as the best next hop toward the sink, and the values stored in A are updated.

- The Beacon timer is activated and node A will periodically send its own Beacon in broadcast.

- The Report message timer is activated: the neighbours table of A is sent periodically to the sink node using the best next hop toward the sink. After each transmission, the list of neighbors is deleted in order to have an updated view of the network. The Report period must be greater than the period used to broadcast Beacon messages (Beacon period).

The information included in the Report messages are used by the controller to create a map of the network. Based on this representation, the controller is able to respond to Rule Requests and to decide the routing paths for data packets, while Rule Request will keep following the previously discovered path.

The actual implementation of the controller uses Dijkstra's routing algorithm to solve Rule Requests. The weight of the edges in the topology representation is a function of the received RSSI.

A possible change in the network is notified to the controller using Report messages. As specified above, the controller obtains periodically all the lists of neighbours, according to the Report period that is bounded by the Beacon period. By decreasing the latter period, a faster responsiveness to environmental changes could be obtained to the detriment of having larger overhead. In the actual implementation of SDWN, the controller sends a Rule Response only after receiving a Rule Request from a node

and rules contained in the nodes expire after a configurable period of time. Therefore, at the end of this period, the controller receives a new Rule Requests for the unmanageable packets.

As previously mentioned, more than one action can be executed for an incoming packet, thus achieving the multicast communication. By performing multiple actions, the controller is able to clone an incoming message into multiple outgoing messages. Unfortunately, a drawback of this approach is that the multicast is locally executed by transmitting a series of unicast messages. In other terms, the broadcast nature of the wireless communication is not exploited.

3.5.3 Experimental Setup

Two network setups were considered: i) a network consisting of 10 nodes (nodes 4, 6, 13, 17, 22, 25, 38, 43, 45, 51); ii) a network of 20 nodes, where the following nodes are added: 1, 8, 10, 11, 15, 20, 23, 30, 31, 33. In all cases, the node 53 at the end of the corridor, is used as the network coordinator.

3.5.3.1 Data Traffics Generated and Environmental Conditions

A query-based application was considered, where the coordinator periodically sends a query packet to one or several *target* nodes, and waits for the reply from it/them. This is a typical scenario for smart building applications, where nodes deployed for metering purposes report measurements to the building concentrator upon request. Both queries and replies are data packets with a given payload that is the same in both cases; different payload sizes were considered.

Two different communication configurations are evaluated: i) *unicast*, where the coordinator sends the query to one specific node that could be one, two or three hops far from the coordinator; and ii) *multicast*, where the coordinator queries contemporaneously a subset of nodes, and waits for replies from all of them.

As for the environmental conditions, all experiments were performed during the night, when no people were moving around, to avoid uncontrollable environmental changes and to ensure a fair comparison. However, in order to measure the level of reactivity of protocols to possible changes such as in real environments, due to for example mobility of people and crowded conditions, the performance in *quasi-static* and *dynamic* conditions was also evaluated. In particular, experiments were still performed during the night, but the "disturbs" specified below were introduced. In the case of *quasi-static* environment, a day-like situation was emulated, where people move around, by letting two people walk along the corridor at a constant speed, following a pre-defined path. The comparison among protocols is still fair, since exactly the same situation was reproduced(same people, path and speed) during all experiments. This case is denoted as *quasi-static*, since only two people were moving without creating huge obstacles and fast fading. In the case of *dynamic* environment, the movement of nodes leaving the network and possibly coming back, was emulated by switching off and on nodes at random instances. In particular, the following procedure was implemented: i) once a node switches on, it remains in this state for at least 5 seconds, after which it ii) generates a random and uniformly distributed delay between 0 and 10 seconds at the end of which iii) the node switches off for 1 second, and then it switches on again (back to step i)). The comparison among protocols is still fair, since the above described duty cycling is implemented in all the

tests identically. Moreover, the channel conditions could be considered as extremely dynamic, since nodes switch off frequently and at random time instances.

3.5.3.2 Parameters settings

All the parameter settings related to PHY and MAC are the equivalent for the three protocols, and they are provided in Table 3.13. Table also includes the network layer parameters, different for the three protocol stacks, but set to the same values, when possible. Therefore, tor the sake of fair compariso the SDWN Beacon packets period was set equal to the ZigBee Link status period, as well as the SDWN Flow tables refreshing time equal to the ZigBee MTO-RR period. Therefore, when the environment is static, routing tables are refreshed and new paths are discovered with the same frequency (i.e., every 150 s). Broadcast packets used to compute link costs/RSSI values are sent with the same frequency (i.e., every 10 s). Obviously, in the presence of changes in the environment, the two protocols behave differently. In case of 6LoWPAN, as stated above, the frequency of generation of DIO packets is managed by the Trickle algorithm: the RPL router will schedule the emission of a DIO at some time in the future, depending on the events in the network and real-time environment condition. In this case, the default period between two consecutive DIO messages equal to 12 s was set.

All the remaining parameters were set to default values, as specified by the corresponding standard. In relation to the packet sizes, all protocols use a MAC acknowledgement of 11 bytes and a PHY header of 6 bytes. The MAC header is 18 bytes in the case of ZigBee and SDWN, since short addresses are used, while it is 22 bytes for

Table 3.13: Parameter Settings

PHY Layer	
Bit Rate	250 kbit/s
Frequency Band	channel 11, at 2.405 GHz
Transmit Power	-5 dBm
Receiver Sensitivity	-92 dBm
PHY layer header	6 bytes
MAC Layer	
BE_{min}	3
BE_{max}	5
NB_{max}	5
Max number of retransmissions at MAC level	3
MAC header for ZigBee and SDWN	18 bytes
MAC header for 6LoWPAN	14 - 22 bytes
NET Layer	
SDWN	
Beacon packet period	10 s
Report packet period	20 s
Flow tables refreshing time	150 s
Maximum number of children per parent	6
ZigBee	
Link status period	10 s
MTO-RR period	150 s
MTO-RR number of retransmissions	3
Maximum number of children per parent	6
Random jitter for broadcast packets	(0, 127) ms
6LowPAN	
Minimum DIO period	12 s
DIO period doublings	8 s
Maximum number of children per parent	6
Random jitter for DAO packets forwarding	(0, 4) s
Random jitter for DIS packets generation	(30 - 60) s

6LoWPAN in the case of unicast packets (data packets and DAO), and 14 bytes in the case of broadcast packets (DIO and DIS), due to the use of long addresses. The MAC Service Data Unit lengths for the different packets and the different protocols are presented in Table 3.14.

3.5.3.3 Performance Metrics

The following performance metrics was considered: i) PLR; ii) round-trip-time (RTT); iii) overhead; and iv) throughput. In all experiments, the coordinator is sending one query every 300 ms toward the target node(s), and a total number of 5,000 queries are generated at the application layer. To compute the PLR, in each experiment the number of replies received at the coordinator, n_{RX}, from each target node, is counted. Therefore, a loss is counted if the query or the reply is lost, independently from the link in which the packet is lost. In the case of unicast transmission $PLR[\%] = (5,000 - n_{RX}) * 100/5,000$, while in the case of multicast an average PLR, averaged among the target nodes is computed. The resolution of the PLR is approximately 0.5%, since 5,000 packets were transmitted.

The RTT is defined as the interval of time between the transmission of the query at the application layer of the coordinator, and the instant in which the reply is received at the application layer of the coordinator as well. In order to compute the RTT of each packet, a software-defined timer implemented at the application layer of the coordinator is used, having a resolution of 1 ms. Results are then averaged over all packets received in each experiment, and among the target nodes for the multicast case.

Table 3.14: MAC Service Data Unit lengths

SDWN Packet Type	MAC Service Data Unit length (bytes)
Data	10 + Payload
Beacon	10 + 2
Report	10 + 3 + (3 * No. of neighbors)
Rule Request	10 + Payload
Rule Response	10 + (16 * No. of rules sent)
Open Path	10 + (2 * No. of nodes in the path)

ZigBee Packet Type	MAC Service Data Unit length (bytes)
Data	15 + Payload
MTO-RR	15
RREC	13 + (2 * No. of nodes in the path)
Link Status	13 + (2 * No. of neighbours)

6LoWPAN Packet Type	MAC Service Data Unit length (bytes)
Data	15 + Payload
DIO	85
DAO	48
DIS	6

Two definitions are used for the overhead: i) the ratio between the total number of packets transmitted in the network (data packets transmitted for the first time or retransmitted, acknowledgement, or control packets), and the number of queries generated at the application layer of the coordinator; ii) the ratio between the total number of bytes transmitted in the network, and the number of bytes of information included in the generated replies. The latter is computed by processing the data gathered by two sniffers located at fixed positions at the end (near the coordinator) and in the middle of the corridor.

The network throughput was measured by counting the average number of payload bits contained in replies per second, correctly received by the coordinator.

Finally, note that results related to energy consumption are not provided in this thesis. However, as this metric is strictly related to both, delays and reliability, the best solution in terms of RTT and PLR is expected to be the best also from the consumption viewpoint.

3.5.4 Numerical Results

In this section, numerical results obtained in the experimental campaign are presented. First, the results for the static and quasi-static cases (Section 3.5.4.1) are provided, then the dynamic case is addressed (Section 3.5.4.2).

3.5.4.1 Static and Quasi-Static Environments

Results among all considered protocols, for the case of static and quasi-static environments are first discussed in the following. In Fig. 3.27, the RTT as a function of

the number of hops for the case of 20 bytes of payload, considering unicast transmission and static environment is shown. The set of target node(s) is different for the different protocols, since different topologies are generated. In particular, the set of target nodes is reported in Table 3.15, with the corresponding number of hops and path connecting the node to the coordinator. It can be observed that the node 51 is always directly connected to the coordinator. For example, the node 4 is connected by three hops in the case of SDWN and 6LoWPAN, while for ZigBee only two hops are needed.

Table 3.15: Target node(s) with the number of hops and paths.

Protocols	1 hop target node	2 hops: target node	2 hops: path	3 hops: target node	3 hops: path
SDWN	51	22	22 - 45 - 53	4	4 - 22 - 45 - 53
ZigBee	51	4	4 - 22 - 53	6	6 - 22 - 46 -53
6LoWPAN	51	13	13 - 43 - 53	4	4 - 25 - 38 - 53

As expected, the RTT increases with the number of hops, since the packet has to pass through more routers. Fig. 3.28 depicts the RTT as a function of the payload size in the case of one hop, considering unicast and static environment. It can be observed that RTT slightly increases with increasing the payload size.

In both figures it can be noticed that SDWN achieves better performance than other solutions, resulting in the lowest RTT in all cases. This is due to the fact that in SDWN, once the path between source and destination is established, forwarding at intermediate routers is very quick, since intermediate nodes just have to check the action corresponding to the received packet. In ZigBee and 6LoWPAN, instead, routing must be performed at each intermediate node, resulting in increased delay.

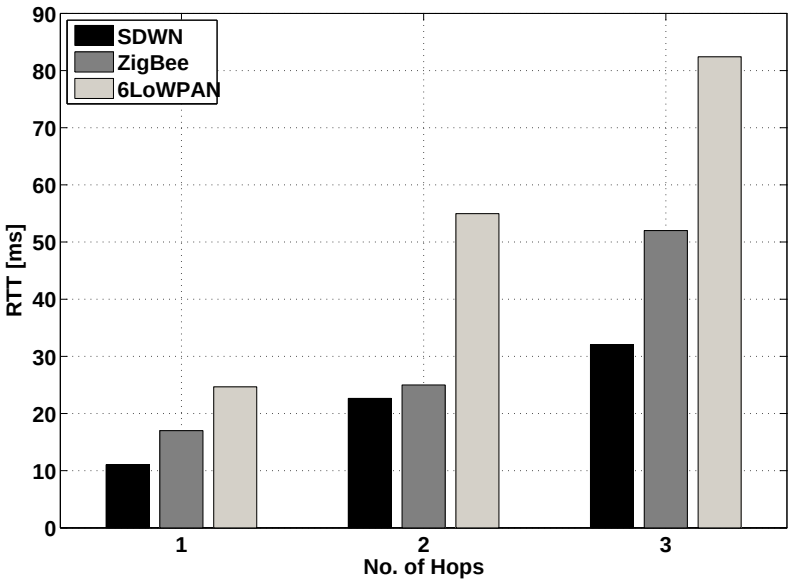

Figure 3.27: Unicast traffic: RTT as a function of the number of hops when transmitting 20 bytes of payload in static conditions.

Moreover, it can be observed that ZigBee notably outperforms 6LoWPAN. The reason is that the protocol stack implemented by 6LoWPAN is more complex, resulting in longer processing time, especially at the adaptation layer (implementing addressing and fragmentation). Finally, the packet size in the case of 6LoWPAN is larger due to the use of IP addresses.

In Table 3.16, the overhead generated by the different protocols when considering a payload of 20 bytes, static environment, unicast traffic and different number of hops is compared. As expected, the overhead is almost doubled by passing from 1 to 2 hops. Moreover, it is increasing by passing from SDWN to ZigBee and to 6LoWPAN solution. This is due to the fact that, in static conditions, SDWN keeps under control the number of packets transmitted during the path formation phase,

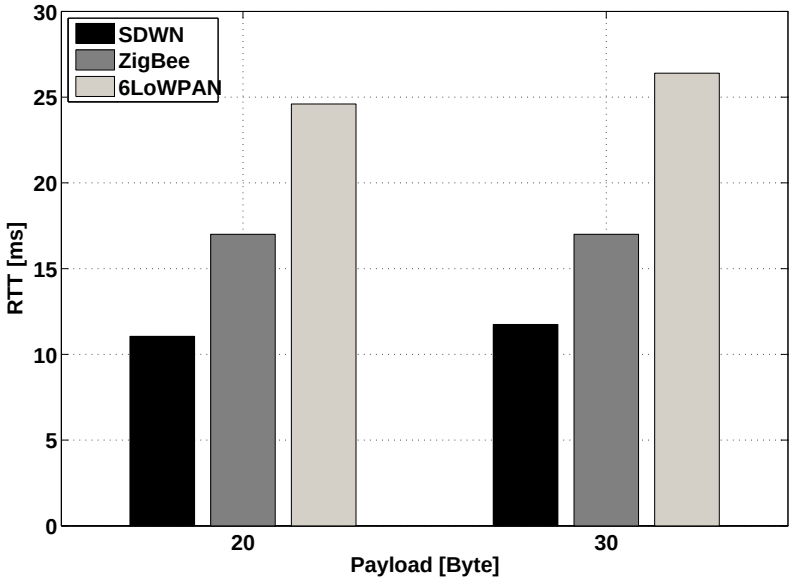

Figure 3.28: Unicast traffic: RTT as a function of the payload size in the case of one hop and static conditions.

while optimising paths reduces the number of data retransmissions. Referring to the overhead in number of bytes, the difference is also more notable, since headers in SDWN are shorter than in ZigBee and 6LowPAN (see Tables II and III).

It should be emphasized that, for all protocols and for each case considered, the PLR was below 0.5%. In Fig. 3.29, RTT achieved in case of static and quasi-static

Table 3.16: Overhead: comparison among protocols.

Protocol	Packets:1 hop	Packets:2 hops	Bytes:1 hop	Bytes:2 hop
SDWN	2.6	5.6	2.5	5.6
ZigBee	4.7	8.7	6.5	11.4
6LoWPAN	6.2	9.5	10.9	16.8

environments is compared, when considering unicast traffic, 20 bytes of payload and 2 hops. As can be seen, in all cases, the RTT increases when passing from static to quasi-static conditions, due to: i) the need for searching for new paths when links become unreliable and/or ii) links being unreliable inducing more retransmissions, thus increasing the latency. However, in the considered environment, SDWN still remains the best solution, since the channel fading is still quite low and changes in the environment are slow, such that SDWN could properly react and work. Finally, note that 6LoWPAN shows the lowest performance degradation when passing from static to quasi-static, since the implemented Trickle algorithm allows for better adaptation of routing to environmental changes. Moreover, in the case of quasi-static environment, the PLR remains below 0.5% for all the cases, demonstrating the good reactivity of protocols when the environment changes slowly. Results related to the multicast traffic, when triggering a multicast group that consists of nodes 4 and 6 are shown in the sequel. Fig. 3.30 shows the average RTT, averaged between the two trigged nodes, while Fig. 3.31 compares the average PLR. As can be seen, RTT is much higher than in the unicast case, especially for 6LoWPAN. The latter is due to an increase of the PLR that was below 0.5% in the case of unicast; losses due to collisions between data packets originating from the nodes 4 and 6 that cause retransmissions, and consequently, the increase of delays. However, the multicast traffic increases the network throughput, as shown in Table 3.17. The throughput was computed by considering an offered traffic of one query every 300 ms. Results demonstrate the improvement of the throughput when passing from unicast to multicast, since more than one node is queried at the same time. Note that, in the case of unicast, the throughput is the same for all the three protocols, since in all cases the PLR is lower than 0.5%.

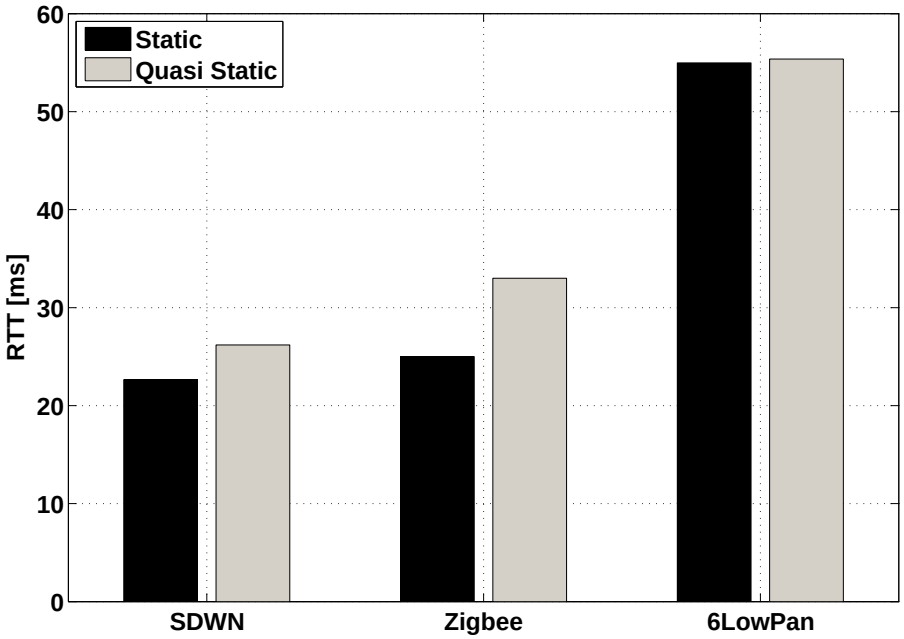

Figure 3.29: Unicast Traffic: RTT for the different protocols in the case of *static* and *quasi-static* conditions, setting 20 bytes of payload and 2 hops.

Table 3.17: Throughput [kbit/s] comparison: unicast and multicast.

Protocol	Unicast 20 Bytes	Unicast 30 Bytes	Multicast 20 Bytes	Multicast 30 Bytes
SDWN	0.53	0.8	1.06	1.59
ZigBee	0.53	0.8	1.05	1.57
6LoWPAN	0.53	0.8	0.97	1.43

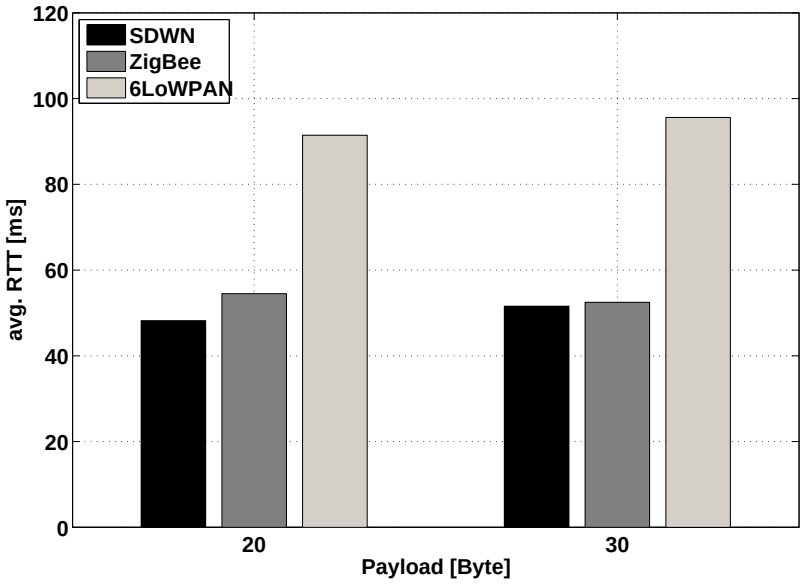

Figure 3.30: Multicast traffic: Average RTT as a function of the payload size.

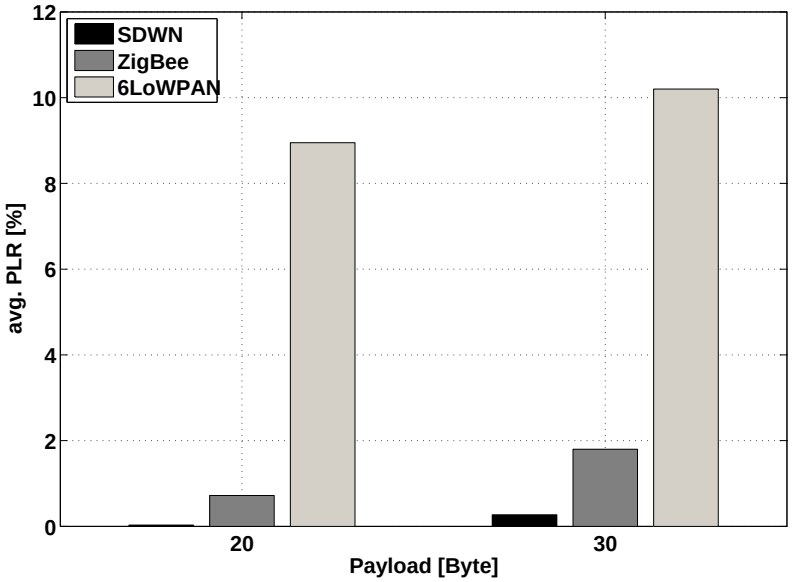

Figure 3.31: Multicast traffic: Average PLR as a function of the payload size.

In the following, a network composed of 20 nodes (selected nodes are reported at the beginning of Sec. 3.5.3), implementing the unicast application with 20 bytes of payload is considered. The coordinator queries node 4, and static and quasi-static environments were considered. Results are reported in Table 3.18, where only the cases of SDWN and ZigBee are considered, having already demonstrated that 6LoWPAN has the worst performance in all cases. As can be seen, SDWN is again performing better than ZigBee, since environmental conditions are still almost static, therefore for larger networks SDWN is still performing well. Obviously, for both protocols, RTT and PLR are larger with respect to the case of 10 nodes network, since more nodes are transmitting packets during the path discovery phase, resulting in more collisions and possibly longer and suboptimal paths.

Table 3.18: 20 nodes network: Comparing RTT and PLR.

Protocol	RTT[ms]: Static	RTT[ms]: Quasi-Static
SDWN	44	49
ZigBee	51	76
Protocol	PLR[%]: Static	PLR[%]: Quasi-Static
SDWN	1.5	2
ZigBee	13	21.5

3.5.4.2 Dynamic Environment

This section concludes by considering the case of dynamic environment, whose performance in terms of RTT and PLR are reported in Table 3.19. Results have been

Table 3.19: Dynamic conditions: Comparing SDWN and ZigBee.

Protocol	RTT [ms]	PLR [%]
SDWN	40	96
ZigBee	61	33.5

achieved by considering the 10 nodes network, unicast application, and 20 bytes of payload, where the coordinator queries node 4. In this case, a highly dynamic environment is emulated by making routers switch on and off at random time instances. This requires nodes to refresh routes very quickly, because a router in a path already established could switch off and the source should search for a new relay for reaching the destination. Both, ZigBee and SDWN experience performance degradation. However, SDWN reaches a very large PLR, since most of the packets cannot find a proper route to reach the coordinator. The average RTT of SDWN still remains lower than in case of ZigBee, since when a packet manages to find a proper route with all routers switched on, forwarding is still very quick. This demonstrates that SDWN presents some issues in the case of highly dynamic environments, as expected.

3.6 Conclusions

In this chapter, a methodology proposed and described in Chapter 1, which allows to reproduce a real world deployment on a downscaled testbed, was implemented. The methodology was validated through experiments and results demonstrate that the behavior of the real world deployment is well reproduced by the downscaled testbed. This methodology was then used to properly select nodes for other experiments aiming

at the optimisation and performance evaluation of routing protocols for smart lighting and smart building systems.

In particular, with reference to smart lighting systems, a novel efficient routing protocol was proposed. The proposed solution is based on an initial discovery phase, to let nodes select the best neighbour to be used as preferred relay to reach the sink. Results of an extensive measurement campaign were shown, to characterise the performance when changing the network size, the environment and the traffic generated. The proposed protocol was compared with several benchmark protocols. Numerical results, derived through experimentation on the field, showed that the proposed solution outperforms the other considered solutions.

Finally, a smart building system was considered. A comparison among different solutions (ZigBee, 6LoWPAN and a software defined-based solution, SDWN) that could be deployed in such system was presented. Results showed that in static and quasi-static conditions SDWN outperforms other solutions, independently on the network size, payload size, traffic generated, and performance metric considered. The reason for this is the fact that SDWN allows to optimise paths selection and minimise forwarding time at routers. However, SDWN presents some limitations when high dynamic environments are considered, because of the time needed to refresh paths.

Conclusions

The general topic investigated in this thesis was performance evaluation of enabling solutions for IoT and Smart City implementation under different perspectives. Both theoretical and experimental aspects were considered and consequently, the work presented in thesis was developed in two phases. In particular, in the first part of the thesis a novel mathematical framework for modelling multi-hop networks was proposed. The main contribution of this thesis stands in the fact that in both phases novel methodologies were proposed and applied to two reference applications of this thesis. Most of works in literature that use Markov chain-based analysis assume that all nodes in the network are considered to operate independently one from each other and work in saturated traffic conditions, or have a packet to be transmitted in the queue with a known probability. The model developed and proposed here, instead, takes into account the dependance among all links in the network. The fact that both single node and network status were considered in the analysis distinguishes this work from those already present in the literature. The node states were modelled through a semi-Markov chain. Then, through a finite state transition diagram describing the network status, parameters to be included in the semi-Markov chain were derived. The analysis was applied to a CSMA-based protocol as well as to slotted

Aloha; however, the extension of the semi-Markov chain analysis to other protocols is straightforward. It was shown that semi-Markov chains can be a useful tool for protocol performance evaluation, as they allow to capture the node and network behaviour while accounting for network evolution in time.

The second part of the thesis is dedicated to the experimental work performed during the PhD, using facilities available at University of Bologna within the EuWIn laboratory. First, as a response to the needs for increased realism identified in Chapter 1, a methodology to reproduce a real world deployment on a downscaled testbed, deployed in an indoor and controlled environment was proposed and described. The proposed procedure allows replication of experiments for optimisation purposes. Networks are often deployed in environments not easily accessible and highly unpredictable, where doing experiments is very expensive and time consuming. Therefore, the proposed procedure significantly simplifies the experimentation and reduces time for running experiments while contemporaneously providing the realistic picture of the protocol performance. The downscaled testbed is represented by a subset of nodes of a controllable platform. The described procedure is based on solving an optimization problem, namely a Rectangular Quadratic Assignment Problem, where the objective function is the minimization of the total connectivity difference between the real world deployment and the downscaled testbed. The methodology can be applied to any real IoT world deployment, provided that the following conditions identified during this work are fulfilled: i) a controllable has a number of nodes at least two or three times larger than the size of the real world deployment, and ii) setting different levels of transmit power is allowed and possible. The second phase of experimental work presented in this thesis was dedicated to the performance evaluation of network

protocols and architectures for smart lighting and smart building systems. Through the downscaling procedure, nodes of testbed were identified and selected in order to emulate the real network deployment.

With regard to smart lighting system, an efficient routing protocol was designed, implemented and tested. The proposed solution is based on an initial discovery phase, to let nodes select the best neighbour to be used as preferred relay to reach the sink. Results of an extensive measurement campaign were shown, to characterise the performance when changing the network size, the environment and the traffic generated. The proposed protocol was compared with several benchmark protocols. Results, derived through experimentation on the field, show that the proposed solution outperforms other considered solutions.

With reference to smart building application, a comparison among different solutions (ZigBee, 6LoWPAN and a software defined-based solution, SDWN, implementing a centralised routing) was presented. Results of an extensive measurement campaign considering various environmental conditions, network sizes and data traffic patterns were provided and discussed. Results show that in static and quasi-static conditions SDWN outperforms the other solutions, independently on the network size, payload size, traffic generated, and performance metric considered. The reason for this is the fact that SDWN allows to optimise paths selection and minimise forwarding time at routers. However, SDWN presents some limitations when high dynamic environments are considered, because of the time needed to refresh paths. It can be concluded that SDWN is the most suitable solution for applications where nodes are in fixed positions and for low mobility scenario, as in this case of smart

buildings applications. However, when the situation is dynamic and an increased mobility of nodes is expected, a distributed solutions like ZigBee and 6LoWPAN could work better. As an example, the case of smart city applications, where nodes could be mounted over lamp posts in streets where object (e.g., cars and people) are moving around, or where nodes could be directly carried by moving objects, requires solutions characterised by high reactivity rather than lower delays.

This work presented in this thesis aimed at providing some general guidelines for the design of systems optimised for specific application-dependent requirements through novel mathematical and empirical methodologies that were proposed.

List of Acronyms

IoT Internet of Things

SC Smart City

RFID Radio-Frequency IDentification

MTO Many-to-one

IP Internet Protocol

IEEE Institute of Electrical and Electronics Engineers

OSI Open Systems Interconnection

SF superframe

NC network coordinator

DAO Destination Advertisement Object

DIS DAG Information Solicitation

DIO DAG Information Object

List of Acronyms

DODAG Destination Oriented Directed Acyclic Graph

RPL Routing Protocol for Low power and lossy networks

RREQ route request

RREP route reply

MTO-RREQ many-to-one route request

MTO-RREP many-to-one route reply

M2M Machine-to-machine

BPSK binary phase shift keying

O-QPSK Offset Quadrature Shift Keying

QPSK Quadrature Shift Keying

SHR Synchronization Header

6LoWPAN Low power Wireless Personal Area Networks 6LoWPAN

IPv6 Internet Protocol version 6

IPv4 Internet Protocol version 4

LLN Low power and Lossy Network LLN

AODV Ad hoc on demand distance vector

WAN Wireless Ambient Network

RSSI Received Signal Strength Indication

ED energy detection

ACK acknowledgement

LQI link quality indicator

MTU maximum transmission unit

OF objective function

RPL Routing Protocol for Lowpower and Lossy Networks (LLNs)

DDSP distributed digital signal processing

DS-SS direct sequence spread spectrum

RFD reduced function device

PHR Physical Header

LLC Logical Link Control

CRC cyclic redundancy check

PAN Personal Area Network

CAP Contention Access Period

CFP Contention Free Period

CSMA carrier-sense multiple access

CSMA/CA carrier-sense multiple access with collision avoidance

CSS chirp spread spectrum

List of Acronyms

SMC semi-Markov chain

FSTD finite state transition diagram

FFD full function device

GTS Guaranteed Time Slot

HHA Hybrid Hierarchical Architecture

i.i.d. independent,identically distributed

ISM industrial scientific medical

MAC medium access control

MFR MAC Footer

MHR MAC Header

MPDU MAC Payload Data Unit

MSDU MAC Service Data Unit

PHY physical

PPDU Physical Protocol Data Unit

PSDU Physical Service Data Unit

RF radio frequency

r.v. random variable

NoE Network of Excellence

EC European Commission

WHN Wireless Hybrid Network

WPAN wireless personal area network

WSN Wireless Sensor Network

SMP semi-Markov Processes

MC Markov chain

EMC Embedded Markov Chain

EuWIn European Laboratory of Wireless Communications for the Future Internet

FSTD finite state transition diagram

Flextop Flexible Topology Testbed

DataSens Data Sensing and Processing Testbed

TI Texas Instruments

ND neighbour discovery

DT data transmission

LWN Linear Wireless Network

LWSN Linear Wireless Sensor Network

IF Irresponsible Forwarding

DA data aggregation

List of Acronyms

PAF Probability Assignment Function

SDN Software Defined Network

SDWN Software Defined Wireless Networking

PLR packet loss rate

BP Beacon Period

CW Contention Window

BE Backoff Exponent

NB Backoff Counter

RTT round-trip-time

List of Figures

List of Figures

List of Tables

Bibliography

[1] J. A. Stankovic. Research directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1):3–9, Feb 2014.

[2] S. L. Keoh and S. S. Kumarand H. Tschofenig. Securing the Internet of Things: A standardization perspective. *IEEE Internet of Things Journal*, 1(3), June 2014.

[3] Cisco. The Internet of Things. *[Online]*, Jan 2014.

[4] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805, October 2010.

[5] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo. A survey on facilities for experimental internet of things research. *Communications Magazine, IEEE*, 49(11):58–67, November 2011.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols and applications. *Communications Surveys Tutorials, IEEE*, PP(99):1–1, 2015.

[7] Lora specification. Lora Alliance, 2015.

[8] Sigfox technology.

[9] IEEE standard 802.15.4 2003: part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs).

[10] R. Khan, S.U. Khan, R. Zaheer, and S. Khan. Future internet: The internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, pages 257–260, Dec 2012.

[11] M.R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L.A. Grieco, G. Boggia, and M. Dohler. Standardized protocol stack for the internet of (important) things. *Communications Surveys Tutorials, IEEE*, 15(3):1389–1406, Third 2013.

[12] Satyanarayana V. Nandury and Beneyaz A. Begum. Smart wsn-based ubiquitous architecture for smart cities. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 2366–2373, Aug 2015.

[13] Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1028–1031, Sept 2011.

[14] S. Suakanto, S.H. Supangkat, Suhardi, and R. Saragih. Smart city dashboard for integrating various data of sensor networks. In *ICT for Smart Society (ICISS), 2013 International Conference on*, pages 1–5, June 2013.

[15] M. Fazio, M. Paone, A. Puliafito, and M. Villari. Heterogeneous sensors become homogeneous things in smart cities. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 775–780, July 2012.

[16] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *Internet of Things Journal, IEEE*, 1(1):22–32, Feb 2014.

[17] Y.M. Yusoff, R. Rosli, M.U. Karnaluddin, and M. Samad. Towards smart street lighting system in malaysia. In *Wireless Technology and Applications (ISWTA), 2013 IEEE Symposium on*, pages 301–305, Sept 2013.

[18] M. Castro, A.J. Jara, and A.F.G. Skarmeta. Smart lighting solutions for smart cities. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1374–1379, March 2013.

[19] R.F. Karlicek. Smart lighting - beyond simple illumination. In *Photonics Society Summer Topical Meeting Series, 2012 IEEE*, pages 147–148, July 2012.

[20] I. Jawhar and N. Mohamed. A hierarchical and topological classification of linear sensor networks. In *Proc. of Wireless Telecommunications Symposium (WTS), 2009*, pages 1–8, April 2009.

[21] Z. Wang, X. Zhao, and Xu Qian. The application and issues of linear wireless sensor networks. In *Proc. of System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2011*, volume 2, pages 9–12, October 2011.

[22] F. Osterlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt. Integrating building automation systems and wireless sensor networks. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pages 1376–1379, Sept 2007.

[23] M.K. Kadiyala, D. Shikha, R. Pendse, and N. Jaggi. Semi-Markov process based model for performance analysis of wireless LANs. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011*, pages 613–618, March 2011.

[24] C. Buratti and R. Verdone. L-CSMA: A MAC protocol for multi-hop linear wireless (sensor) networks. *IEEE Transactions on Vehicular Technology*, PP(99):1–1, 2015.

[25] Zigbee specification. ZigBee Alliance, 2008.

[26] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[27] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of IEEE Workshop on Mobile Computing Systems and Applications, 1999*, February 1999.

[28] Z. Shelby and C. Bormann. 6lowpan: the wireless embedded internet, 2010.

[29] IETF RFC 4919. Rfc 3819, internet engineering task force rfc 3819, advice for internet subnetwork designers, 2004.

[30] J. Hui G. Montenegro, N. Kushalnagar and D. Culler. RFC 4944, internet engineering task force rfc 4944, transmission of IPv6 packets over IEEE 802.15.4 networks, 2007.

[31] G. Fairhurst D. Grossman R. Ludwig J. Mahdavi G. Montenegro J. Touch P. Karn, C. Bormann and L. Wood. RFC 4944, internet engineering task force rfc 4944, transmission of IPv6 packets over IEEE 802.15.4 networks, 2007.

[32] JeongGil Ko, A. Terzis, S. Dawson-Haggerty, D.E. Culler, J.W. Hui, and P. Levis. Connecting low-power and lossy networks to the internet. *Communications Magazine, IEEE*, 49(4):96–101, April 2011.

[33] T. Winter et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. In *RFC 6550 (Proposed Standard), Internet Engineering Task Force*, March 2012.

[34] J. Hui O. Gnawali P. Levis, T. Clausen and J. Ko. Rfc 6206, internet engineering task force rfc 6206, the trickle algorithm, 2011.

[35] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[36] J. Misic, S. Shafi, and V.B. Misic. Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic. *Parallel and Distributed Systems, IEEE Transactions on*, 17(4):361–376, April 2006.

[37] S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor. Performance analysis of slotted Carrier Sense IEEE 802.15.4 Medium Access Layer. *Wireless Communications, IEEE Transactions on*, 7(9):3359–3371, September 2008.

[38] Hao Wen, Chuang Lin, Zhi-Jia Chen, Hao Yin, Tao He, and Eryk Dutkiewicz. An improved Markov model for IEEE 802.15.4 slotted CSMA/CA mechanism. *Journal of Computer Science and Technology*, 24(3):495–504, 2009.

[39] Pangun Park, P. Di Marco, P. Soldati, C. Fischione, and K.H. Johansson. A generalized Markov chain model for effective analysis of slotted IEEE 802.15.4. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 130–139, Oct 2009.

[40] C. Buratti. Performance analysis of IEEE 802.15.4 beacon-enabled mode. *Vehicular Technology, IEEE Transactions on*, 59(4):2031–2045, May 2010.

[41] Hao Wang, Guixia Kang, and Kai Huang. An Advanced Semi-Markov Process Model for Performance Analysis of Wireless LANs. In *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pages 1–5, Sept 2012.

[42] P. Di Marco, Pangun Park, C. Fischione, and K.H. Johansson. Analytical modeling of multi-hop IEEE 802.15.4 networks. *Vehicular Technology, IEEE Transactions on*, 61(7):3191–3208, Sept 2012.

[43] Duong Hoang and R.A. Iltis. Performance evaluation of multi-hop csma/ca networks in fading environments. *Communications, IEEE Transactions on*, 56(1):112–125, January 2008.

[44] G.J. Sutton, R.P. Liu, and I.B. Collings. Modelling ieee 802.11 dcf heterogeneous networks with rayleigh fading and capture. *Communications, IEEE Transactions on*, 61(8):3336–3348, August 2013.

[45] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin. Unsaturated throughput analysis of ieee 802.11 in presence of non ideal transmission channel and capture effects. *Wireless Communications, IEEE Transactions on*, 7(4):1276–1286, April 2008.

[46] P. Di Marco, C. Fischione, F. Santucci, and K.H. Johansson. Modeling IEEE 802.15.4 networks over fading channels. *Wireless Communications, IEEE Transactions on*, 13(10):5366–5381, Oct 2014.

[47] J.R. Gallardo, D. Makrakis, and H.T. Mouftah. On modeling contention-based MAC protocols using Markov chains. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6, May 2010.

[48]

[49] I. Jawhar, N. Mohamed, and K. Shuaib. A framework for pipeline infrastructure monitoring using wireless sensor networks. In *Proc. of Wireless Telecommunications Symposium (WTS), 2007*, pages 1–7, April 2007.

[50] X. Yang, K.G. Ong, W.R. Dreschel, K.Zeng, C.S. Mungle, and C.A. Grimes. Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment. *Sensors*, 2002(2):455–472, 2002.

[51] Y.M. Yusoff, R. Rosli, M.U. Karnaluddin, and M. Samad. Towards smart street lighting system in Malaysia. In *Proc. of IEEE Symposium on Wireless Technology and Applications (ISWTA), 2013*, pages 301–305, September 2013.

[52] R. Nelson. *Probability, Stochastic Processes, and Queueing Theory*. New York, USA, 1995.

[53] Texas Instruments CC2530 Datasheet. Texas Instruments.

[54] Texas Instruments CC2530 datasheet. Texas Instruments.

[55] P. De, A. Raniwala, S. Sharma, and C. Tzicker. MiNT: a miniaturized network testbed for mobile wireless research. In *Proceedings IEEE INFOCOM 2005.*, pages 2731–2742 vol. 4.

[56] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, pages 1664–1669 Vol. 3.

[57] V. Naik, E. Ertin, Hongwei Zhang, and A. Arora. Wireless testbed Bonsai. In *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–9, 2006.

[58] Jing Lei, R. Yates, L. Greenstein, and Hang Liu. Mapping link snrs of real-world wireless networks onto an indoor testbed. *IEEE Transactions on Wireless Communications*, 8(1):157–165, Jan 2009.

[59] L. Sanchez, V. Gutierrez, J.A. Galache, P. Sotres, J.R. Santana, J. Casanueva, and L. Munoz. Smartsantander: Experimentation and service provision in the smart city. In *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–6, June 2013.

[60] Angelo Cenedese, Andrea Zanella, Lorenzo Vangelista, and Michele Zorzi. Padova smart city: An urban internet of things experimentation. In *International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2014.

[61] Mihael Mohorcic, Miha Smolnikar, and Tomaz Javornik. Wireless sensor network based infrastructure for experimentally driven research. In *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5, Aug 2013.

[62] A.G. Dludla, A.M. Abu-Mahfouz, C.P. Kruger, and J.S. Isaac. Wireless sensor networks testbed: ASNTbed. In *IST-Africa Conference and Exhibition (IST-Africa)*, pages 1–10, May 2013.

[63] Paolo Casari, Angelo P. Castellani, Angelo Cenedese, Claudio Lora, Michele Rossi, Luca Schenato, and Michele Zorzi. The wireless sensor networks for city-wide ambient intelligence (WISE-WAI) project. *Sensors*, 9(6):4056–4082, 2009.

[64] Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. TWIST: A scalable and reconfigurable testbed for wireless indoor experiments with sensor network. In *Proc. of the 2nd Intl. Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality, (RealMAN 2006)*, May 2006.

[65] Very large scale open wireless sensor network testbed, sensLAB. https://www.iot-lab.info/.

[66] Rainer E Burkard, Mauro Dell'Amico, and Silvano Martello. Assignment problems, revised reprint, 2009.

[67] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, July 1976.

[68] IBM ILOG CPLEX Optimizer. urlhttp://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, 2010.

[69] GeraldoR. Mateus, MauricioG.C. Resende, and RicardoM.A. Silva. Grasp with path-relinking for the generalized quadratic assignment problem. *Journal of Heuristics*, 17(5):527–565, 2011.

[70] Mohamed Saifullah Hussin and Thomas Stützle. Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances. *Comput. Oper. Res.*, 43:286–291, March 2014.

[71] Zigbee specification. ZigBee Alliance, 2008.

[72] D C Baird. Experimentation: an introduction to measurement theory and experiment design, 1962.

[73] Hongqiang Zhai and Yuguang Fang. A distributed packet concatenation scheme for sensor and ad hoc networks. In *Proc. of Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 1443–1449 Vol. 3, October 2005.

[74] S. Panichpapiboon and G. Ferrari. Irresponsible forwarding. In *8th International Conference on Intelligent Transport System Telecommunication (ITST)*, pages 311–316, Phuket, Thailand, Oct 2008.

[75] S. Busanelli, G. Ferrari, and S. Panichpapiboon. Efficient broadcasting in IEEE 802.11 networks through Irresponsible Forwarding. In *IEEE Global Telecommunications Conference*, pages 1–6, Honolulu, Hawaii, USA, Dec 2009.

[76] IEEE 802.15.4 Standard. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).*

[77] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo. Software defined wireless networks: Unbridling SDNs. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, Oct 2012.

[78] A. Mahmud and R. Rahmani. Exploitation of openflow in wireless sensor networks. In *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, volume 1, Dec 2011.

[79] P. Dely, A. Kassler, and N. Bayer. Openflow for wireless mesh networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, July 2011.

[80] C. Chaudet and Y. Haddad. Wireless software defined networks: Challenges and opportunities. In *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, Oct 2013.

[81] T. Miyazaki, S. Yamaguchi, K. Kobayashi, J. Kitamichi, Song Guo, T. Tsukahara, and T. Hayashi. A software defined wireless sensor network. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*, Feb 2014.

[82] Lili Liang, Lianfen Huang, Xueyuan Jiang, and Yan Yao. Design and implementation of wireless smart-home sensor network based on Zigbee protocol. In *Communications, Circuits and Systems, 2008. ICCCAS 2008. International Conference on*, May 2008.

[83] S.D.T. Kelly, N.K. Suryadevara, and S.C. Mukhopadhyay. Towards the implementation of IoT for environmental condition monitoring in homes. *Sensors Journal, IEEE*, 13(10):3846–3853, Oct 2013.

[84] C. Gezer, C. Buratti, and R. Verdone. Capture effect in ieee 802.15.4 networks: Modelling and experimentation. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, pages 204–209, May 2010.

[85] D. M. Abrignani, C. Buratti, and R. Verdone. Testing the impact of Wi-Fi interference on ZigBee networks. In *Proc. of IEEE Euro Med, 2014*, Nov 2014.

[86] M. Armholt, S. Junnila, and I Defee. A non-beaconing Zigbee network implementation and performance study. In *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007.

[87] E.D. Pinedo-Frausto and J.A Garcia-Macias. An experimental analysis of Zigbee networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, Oct 2008.

[88] M. Franceschinis, C. Pastrone, M.A Spirito, and C. Borean. On the performance of ZigBee Pro and Zigbee IP in IEEE 802.15.4 networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, Oct 2013.

[89] B. Pediredla, Kevin I-Kai Wang, Z. Salcic, and A. Ivoghlian. A 6LoWPAN implementation for memory constrained and power efficient wireless sensor nodes. In *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, Nov 2013.

[90] G. Pellerano, M. Falcitelli, M. Petracca, M. Pagano, and P. Pagano. 6LoWPAN conform ITS-station for non safety-critical services and applications. In *ITS Telecommunications (ITST), 2013 13th International Conference on*, Nov 2013.

[91] S. Dawans, S. Duquennoy, and O. Bonaventure. On link estimation in dense RPL deployments. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, Oct 2012.

[92] K. Heurtefeux and H. Menouar. Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, April 2013.

[93] M. Kovatsch, M. Weiss, and D. Guinard. Embedding internet technology for home automation. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, Sept 2010.

[94] E. Toscano and L. Lo Bello. Comparative assessments of IEEE 802.15.4/Zigbee and 6LoWPAN for low-power industrial WSNs in realistic scenarios. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, May 2012.

Publications

The work described in this thesis has led to the following publications to Journals:

- Buratti, Chiara; Stajkic, Andrea; Gardasevic, Gordana; Milardo, Sebastiano; Abrignani, Melchiorre; Mijovic, Stefan; Morabito, Giacomo; Verdone, Roberto; "Testing Protocols for The Internet of Things on The EuWIn Platform"; "IEEE Internet of Things Journal"; 3(1):124-133, October 2015, DOI: 10.1109/JIOT.2015.2462030.

- Stajkic, Andrea; Abrignani, Melchiorre; Buratti, Chiara; Bettinelli, Andrea; Vigo, Daniele; Verdone, Roberto; "From a Real Deployment to a Downscaled Testbed: A Methodological Approach"; "IEEE Internet of Things Journal"; PP(99):11, February 2016, DOI: 10.1109/JIOT.2016.2521170

- Stajkic, Andrea; Buratti, Chiara; Verdone, Roberto; "Modeling Multi-Hop Networks Using Contention-Based MAC Through Semi-Markov Chains"; submitted to "IEEE Transactions on Networking";.

- Mijovic, Stefan; Stajkic, Andrea; Cavallari, Riccardo; Buratti, Chiara; "Low Power Listening in BAN: Experimental Characterisation"; "Journal International Journal of E-Health and Medical Communications"; October 2014, DOI: 10.4018/ijehmc.2014100104.

and to the following papers presented to International Conferences:

- Mijovic, Stefan; Stajkic, Andrea; Cavallari, Riccardo; Buratti, Chiara; "Experimental characterization of Low Power Listening in BAN"; IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom), Lisbon, Portugal, pages 145-149, October 2013, DOI: 10.1109/HealthCom.2013.6720656

- Abrignani, Melchiorre; Buratti, Chiara; Dardari, Davide; El Rachkidy, Nancy; Guitton, Alex; Martelli, Flavia; Stajkic, Andrea; Verdone, Roberto; "The EuWIn Testbed for 802.15.4/Zigbee Networks: From the Simulation to the Real World"; IEEE International Symposium on Wireless Communication Systems (ISWCS), Ilmenau, Germany, pages 1-5, Aug 2013.;

- Gardasevic, Gordana; Mijovic, Stefan; Stajkic, Andrea; Buratti, Chiara; "On the performance of 6LoWPAN through experimentation"; IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, pages 696-701, Aug 2015, DOI: 10.1109/IWCMC.2015.7289168;

- Stajkic, Andrea; Buratti, Chiara; Verdone, Roberto; "Modeling multi-hop CSMA-based networks through Semi-Markov chains"; IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik,

Croatia, pages 520-525, Aug 2015, DOI: 10.1109/IWCMC.2015.7289138;

- Stajkic, Andrea; Clazzer, Federico; Liva, Gianluigi; "Neighbor Discovery in Wireless Networks: A Graph-based Analysis and Optimization"; accepted for pubblication in "IEEE ICC 2015 Workshop on Massive Uncoordinated Access Protocols";

As for the Newcom# project, contributions to the following deliverables were given:

- D22.1 – "Definition of EuWIn@CNIT/Bologna testbed interfaces and preliminary plan of activities", edited by Davide Dardari, April 2013.
- D22.2 – "Preliminary tests over the lab infrastructures ", edited by Davide Dardari, November 2013.
- D22.3 – "Experimental results over the lab infrastructure", edited by Davide Dardari, November 2014.
- D22.4 – "Final results obtained in the lab infrastructures", edited by Davide Dardari, November 2014.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Roberto Verdone and Chiara Buratti. To Prof. Verdone for giving me this unique opportunity to join Radio Networks group and spend my PhD within it. Thank you for your positivism and for encouraging me and showing me that I can do things I thought I would never be able to. Thank you for your highly professional hand made maps that guided me and Chiara never allowing to get lost. Or was it so?

Thank you Chiara, for being a supervisor, colleague, friend and family when needed during these years. Thank your for recognizing when I needed the help and support even without the need to ask for it.

Special thanks to reviewers, Prof. Gianluigi Ferrari and Prof. Michele Zorzi, for their detailed suggestions and comments.

I wish to thank to DLR Satellite Networks crew for the hospitality and memorable 3 months. Special thanks to Gianluigi and Federico for their time, kindness and patience they had with me even when it seemed that we speak different languages.

I am very thankful to my crew, Danilo, Riccardo and Stefan, for growing up together, for not letting me "become 100% woman" and for sharing this journey with me from the first to the last day, with all the beautiful and bad moments we had. In particular, thanks to Stefan who taught me how to be an engineer (or least tried), how to maintain objectivity by all means and in any situation and how to face problems in any sphere of life.

I owe my sincere thanks to my girls, Milica, Barbara, Kico and Ana. Thank you

for all the words, joy and tears we shared that meant a world to me. The distance lost in this game. Thank you Milan, for the infinite laughter and strength and for not letting the 'seeyli' Andrea fade away even when it wasn't easy, smooth and shiny. Thank you Ranko, for beating any kind of problems by your genuine sarcasm and beautiful voice.

I am thankful to all my "old" and new office mates, Flavia, Ramona, Giorgia, Colian, Charles, Luca, Lucia, who made going to work a pleasure, not a burden. Guys, this journey would have been much harder without you. As someone would say, seriously.

Finally, i owe the biggest gratitude to my family, for the love and support and for the most valuable life lessons.

This thesis is dedicated to all of you who participated in this journey and made it possible to arrive to this page.