STUDY AND IMPLEMENTATION OF WIRELESS SENSOR NETWORKS TO MANAGE ENERGY IN A SMART HOME

A Thesis Presented to the Department of

Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of

MASTER of Computer Science

By

Adebayo Gbenga Waliyi

Abuja, Nigeria

December 2017.

CERTIFICATION

This is to certify that the thesis title **"Study and Implementation of Wireless Sensor Network to Manage Energy in a Smart Home"** submitted to the school of postgraduate studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of the Master's degree is a record of original research carried out by Adebayo Gbenga Waliyi in the Department of Computer Science.

STUDY AND IMPLEMENTATION OF WIRELESS SENSOR NETWORKS TO MANAGE ENERGY IN A SMART HOME

By

Adebayo Gbenga Waliyi

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED: ______________________________________

Supervisor, Prof. Ousmane Thiare

____

Co-supervisor

____

Head, Department of Computer Science

APPROVED: ______________________________

Chief Academic Officer

____

December 9th, 2017

ABSTRACT

Adebayo Gbenga Waliyi: **Study and implementation of Wireless Sensor Network to Manage Energy in a Smart Home.**

(Under the direction of Ousmane Thiare)

Many studies have shown that smart homes can use energy more efficiently than traditional buildings. Thus, several researchers have advocated building a smart home in order to reduce energy consumption. In the literature, Wireless Sensor Network is adopted as the dominant technology for every proposed smart home . In this thesis I explain the challenges of the state of the art technology in WSN for energy management in a smart home using one of the most prominent LPWAN technologies: LoRa^{TM}. The system is not only low-cost but is also flexible enough to accept multiple sensor nodes and collect the data, irrespective of the distance from the gateway to the various appliances in the smart home. Leveraging on LPWAN technologies, I establish a series of models that cover various aspects of a LoRa network.

Then, a new Network Simulator 3 (NS3) module is introduced to simulate a LoRa-based IoT network in a typical urban scenario. Finally, the performance of the LoRa system is evaluated and analysed.

I emphasize the importance of having a generic system in which its LoRaWAN network configuration function set (router and coordinator) mode of data transmission will be implemented in API mode to accommodate any sensor node for better packet reception (RX) and transmission (TX).

DEDICATION

This research work is dedicated to Allah and my late father (Mr. Isiaka Adebayo).

ACKNOWLEDGEMENTS

It has been a great privilege to work under Prof. Ousmane Thiare as my supervisor. I consider myself very lucky to have my work thoroughly scrutinized by such an erudite scholar. I express my unreserved gratitude to him for being patient enough to read my work, guide and encourage me, and for his willingness to discuss my problems.

I wish to thank Prof. Amos David, HoD Computer Science, whose continuous project presentations have exposed me to the real world of research. Kudos to him.

I express my sincere gratitude to my uncle, Mr. Olajide Salami, and a friend like no other, Mr. Shittu Ismail. I really appreciate your support both in cash and in kind.

I appreciate sincerely my dear mother, Iya Tailor, and my fiancée, Adunni, for their prayers and moral support.

I appreciate the support of my siblings, cousins, and family.

Last, but not the least, I appreciate my league of friends: Kaneh, Hisman, Habib, Yusuff, Nas (room-mate), Moukhtar and others who are too numerous to mention.

Table of Contents

CERTIFICATION ii

ABSTRACT v

DEDICATION vi

ACKNOWLEDGEMENTS vii

LIST OF FIGURES x

LIST OF TABLES x

LIST OF PROCEDURES x

LIST OF ABBREVIATIONS xi

CHAPTER ONE : INTRODUCTION 1

- 1.1 Background of the Research 1
- 1.2 Brief Overview of Wireless Sensor Network 2
  - 1.2.1 Characteristics of Wireless Sensor Networks 3
  - 1.2.2 Requirements for Wireless Sensor Networks 4
- 1.3 Statement of Purpose 5
- 1.4 Motivation 5
- 1.5 Objectives of the Research 6
- 1.6 Research Methodology 6
- 1.7 Organization of the Thesis 6

CHAPTER TWO : LITERATURE REVIEW 8

- 2.1 State of The Art 8
- 2.2 The Advent of Smart Homes 9
- 2.3 Present Smart Home 10
- 2.4 Benefits of the Smart Home 12
- 2.5 Overview of Wireless Communication Technology 13
  - 2.5.1 Solutions for IoT Connectivity 13
- 2.6 The LoRa Modulation 16
  - 2.6.1 LoRa's Chirp Spread Spectrum Implementation 16
  - 2.6.2 LoRa Physical Layer Packets 18
  - 2.6.3 Spreading Factor Orthogonality 21

2.6.4 Main Semtech Chips and Independent Implementations 22
2.7 The LoRaWAN Standard 23
2.7.1 Topology and Device Classes 23
2.7.2 Packet Structure and MAC Commands 26
2.7.3 Encryption and Device Activation 27
2.7.4 Frequency Bands 28
2.7.5 Notable LoRaWAN Implementations 29
2.8 European Regulations 29
2.8.1 Effective Radiated Power (ERP) Limitations 30
2.8.2 Duty Cycle Limitations 32
2.8.3 Channel Line-up 32
CHAPTER THREE METHODOLOGY 34
3.1 Implementation 34
3.2 Network Simulator 3 34
3.3 The LoRa Module 37
3.3.1 PeriodicSender 37
3.3.2 End Device LoRaMac 38
3.3.3 EndDeviceLoraPhy 40
3.3.4 LoraChannel 43
3.3.5 LoraNetDevice 44
3.4 Helpers and Tests 45
CHAPTER FOUR RESULTS AND DISCUSSION 48
4.1 Performance Evaluation 48
4.2 Throughput Performance 48
4.3 Comments 51
CHAPTER FIVE CONCLUSIONS AND FUTURE WORK 52
5.1 Conclusions 52
5.2 Future Work 53
REFERENCES: 54

LIST OF FIGURES

Figure 2.1: Smart Home Devices 11
Figure 2.2: The Smart Home Model Architecture 12
Figure 2.3: Spectrogram representation of a LoRa signal 19
Figure 2.4: De-chirped version of a LoRa signal 19
Figure 2.5: Logic scheme for the SX1301 chip 22
Figure 2.6: Sample topology of a LoRa network 23
Figure 2.7: Protocol stacks of the various devices in a LoRaWAN 24
Figure 2.8: Packet structure of a LoRaWAN message 25
Figure 3.1: The LoRaWAN stack as it was represented in the LoRa module 38

LIST OF TABLES

Table 2.1 SNR values for different spreading factors. 17
Table 2.2 Bit rate [bits/s] for a range of spreading factors and bandwidths 18
Table 2.3 Frequency bands for various regions 28
Table 2.4 Maximum on and minimum off times based on duty cycle definition 51
Table 2.5 Channel line-up for LoRa according to ETSI regulations. 55

LIST OF PROCEDURES

Procedure 3.1 Sending procedure in the End Device LoRaMac class 39

Procedure 3.2 Beginning of reception of a packet 41

Procedure 3.3 End of reception of a packet 43 43

Procedure 3.4 Determine whether a packet was destroyed by interference or not 46

LIST OF ABBREVIATIONS

5G	Fifth Generation
6LowPAN	IPv6 Low Power Wide Area Network
ABP	Activation by Personalization
ACK	Acknowledgement
ADR	Adaptive Data Rate
AMI	Advanced Metering Infrastructure
API	Application Programming Interface
BLE	Bluetooth Low Energy
BW	Bandwidth
CAD	Carrier Activity Detection
CIoT	Cellular Internet of Things
CMU	Central Management Unity
CPU	Central Processing Unit
CR	Carrier Parity
CRC	Cyclic Redundancy Check
CSS	Chirp Spread Spectrum
dB	decibel
DDR	Dynamic Data Rate
DES	Descript Event Simulation
DL	Downlink
EC	European Commission
ECC	Electronic Communications Committee
EC-GSM	Extended Coverage GSM
ECI	External Communication Interface
EDs	End Devices
ERP	Effective Radiated Power
ETS	European Telecommunication Standard Institute
FEC	Forward Error Correction
FFT	Fast Fourier Transform
GPL	General Public License
GWs	Gateways
HEAI:	Home Equipment and Appliances Interface
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical
ISO/OSI	Open Standard Interconnection

ITU	International Telecommunication Union
LBT	Listen Before Talks
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LR-WPAN	Low Rate Wide Personal Area Network
LTE	Long Term Evolution
MAC	Media Access Control
MFSK	Multiple Frequency Shift Keying
MIC	Message Integrity Code
NB-IoT	Nano Band Internet of Things
NF	Noise Figure
NS	Network Server
NS3	Network Simulator 3
OCS	Optimal Consumption Schedule
OTAA	Over the Air Activation
PHY	Physical
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
Rb	Bit Rate
RLC	Residential Load Control
Rx	Reception
SF	Spreading Factor
SHE	Smart Home Environment
SHEMS	Smart Home Energy Management System
SINR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio
ToA	Time on Air
Ts	Symbol Duration
Tx	Transmission
UI	User Interface
UL	Uplink
UNB	Ultra Nano Band
Wi-Fi	Wireless
WSN	Wireless Sensor Network

CHAPTER ONE :
INTRODUCTION

1.1 Background of the Research

Many surveys have shown that smart homes can utilize energy more efficiently than traditional buildings (Alhaj et al., 2015). Thus, several researchers have advocated building a smart home in order to reduce energy consumption. In the literature Wireless Sensor Network (WSN) is adopted as the dominant technology for every proposed smart home .The WSN, rather than Wi-Fi, has been popularly employed for remote control and monitoring applications because it is low cost and consumes little power. For a smart home or home automation various sensors for reducing energy consumption are applied to acquire data from objects and their surrounding environments. Sensors are the devices that can replace or extend the human being's physical senses of sight, hearing, taste, smell and touch.

The application of WSN has been proven to be more flexible and advantageous in such areas as smart homes, telemedicine, industry, environmental monitoring, agriculture, warehouse tracking, transport logistics and surveillance. A WSN consists of spatially distributed autonomous sensors that monitor and control variables that include temperature, voltage, and current. Smart homes can also be called "automated homes" as they can, inter alia, control devices and be used for surveillance purposes.

One of the main purposes of smart homes is to reduce energy consumption which remains the focus of this research work (Nagar et al., 2016). To achieve this goal, smart controls must be implemented in a smart home. Additionally, sensors and smart controllers, by monitoring the exterior and interior lighting levels, enable daylight to be used to reduce the use of electrical lighting while sufficiently illuminating the home. Although many ideas about smart lighting control for energy saving in smart homes have been proposed, a smart lighting control system with high reliability and control accuracy remains to be found.

The main purpose of this research is to propose and implement a network capable of collecting relevant information in a smart home. Wireless sensor networks are composed of a large number of miniature self-organizing wireless sensor nodes. WSN can detect, collect and deal with the information in its surrounding area and then send data to the controller/user (Alhaj et al., 2015). In WSN for smart homes, each node in the network is independent of the other nodes; they are battery powered, small with attached sensors. In this research work, smart homes for energy management using wireless sensor network have been designed for user convenience via the open-source LoRa server project. Hence, it can be implemented without any user intervention.

1.2 Brief Overview of Wireless Sensor Network

Sensing is simply used for obtaining information about a physical object or a process such as change in temperature or pressure. Any object that is able to do this is called a Sensor. When many sensors cooperatively monitor large physical environments, they form a Wireless Sensor Network. The sensor nodes communicate with centralized controls called base stations, also known as the sink nodes. A base station normally allows the dissemination of information to another network, a powerful data processing or storage centre or an access point for human interface.

Communication with the base station could either be single-hop, where the nodes transmit data directly to the base station, or multi-hop, where some clients serve as relays for other sensor nodes, that is, they collaborate to propagate sensed data towards the base station. There could be variation in the processing and communication capabilities of the sensor nodes in WSN. Some could be simple nodes while others could be categorized as complex nodes depending on their configurations.

The two most important operations of a WSN are data dissemination (sending data/queries from the sinks to the sensor nodes) and data gathering (sends sensed data from sensor nodes to the sinks).

The architecture of the network could either be “flat”, where each node plays the same sensing tasks and there is no global identifier in a sensor network or “hierarchical”, where sensor nodes are divided into clusters, where the cluster members send their data to the cluster head which sends the data to the sink node.

The IEEE 802.11 family of standards, which was introduced in 1997, is the most common wireless networking technology for mobile systems. However, the high energy overheads of IEEE 802.11-based networks makes this standard unsuitable for low-power sensor networks (Singh & Singh, 2010). This has led to the development of a variety of protocols that better satisfy the networks’ need for low power consumption and low data rates. These sensor nodes, however, possess some major characteristics which are described below.

1.2.1 Characteristics of Wireless Sensor Networks

- **Limited Resource:** Power consumption is highly constrained as nodes depend on batteries or energy captured from the environment. Memory and processing capacity of the nodes is also limited due to the small size of the nodes. Energy is a crucial resource for sensor networks. Therefore, developing energy-saving techniques has a significant impact on the network architecture.

- **Large Scale of Development:** A sensor network may consist of thousands of heterogeneous nodes with one or more centralized control called Base Stations. The network structure and resources used are often ad hoc.

- **Specific Application:** A sensor node is usually designed to serve a specific application. The nature of the sensor’s application may affect the cost as well as the physical size of the sensor nodes.

- **Harsh Environmental Conditions:** Sensor networks often operate in environments with harsh conditions and should possess the ability to withstand these conditions.

- **Node Failure Recovery:** Because the nodes are usually deployed in a remote and hostile environment, there is usually little or no human intervention. The network topology should therefore have the ability to tolerate the failure of nodes and activate self-configuring schemes to avoid network partition.

- **Self-Management:** When deployed in remote/harsh environments, the nodes should be able to configure themselves, adapting to failures without human intervention. In these energy-constrained devices, the self-management features must be designed and implemented so that few overheads are incurred.

1.2.2 Requirements for Wireless Sensor Networks

1. **Fault tolerance:** Despite the fact that the sensor nodes are prone to errors because of node failure due to the harsh environment, there should be consistency in the network functionality.
2. **Lifetime:** The nodes are dependent on either batteries or energy scavenged from the environment for power supply. The nodes should therefore be able to function to full capacity before completely exhausting the batteries. Thus, energy saving and load balancing must be taken into account in the design and implementation of the WSN platform, protocols and application.
3. **Scalability:** The protocol defined in the network should be able to adapt to high densities and numerous numbers of nodes.
4. **Real-Time:** Strict timing constraint for sensing, processing and communication are necessary since the network is tightly related to the real world.
5. **Production Cost:** Since a large number of nodes are being deployed, the cost of production should be low.
6. **Security:** The need for security in WSNs is evident due to the nature of the nodes. The remote and unattended operation of sensor nodes increases their exposure to malicious intrusions and attacks. Attacks are mainly targeted at the power of the nodes to prevent successful sensor communications.

1.3 Statement of Purpose

Future homes will be able to offer a range of different smart services, e.g. energy, utility, entertainment, medical, and security, all of which require a reliable way to transfer information and a way to detect abnormalities such as the disconnection of nodes or interference (Amato, et al., 2016). All wireless networks are affected by interference, which makes timed deliveries hard to achieve. As for wireless sensor networks, not only is it prone to interference, but the network could also experience changes in its topology (Homes & Tedblad, 2015). Nodes might crash or be physically moved which will result in changes in the network's topology.

Therefore, it is important to consider these asynchronous and dynamic factors in the network. The principle of smart homes is based the collection of a set of information to provide services such as those related to security, helping people, management of energy, etc. In the case of energy management, the aim, among other things is to reduce the electricity bill but more specifically to reduce global energy consumption and carbon footprint as well as to prevent blackouts. This research proposes a technology to implement a wireless sensor network capable of collecting relevant information for the management of energy.

1.4 Motivation

Routing in wireless sensor networks is still an open research field. Many protocols have been designed and evaluated using simulation tools. Only a few have been tested in real world scenarios. This turns out to be a problem in the evaluation of these protocols. In the literature, it is shown that the results of simulation studies do not always reflect the results measured in real world scenarios. For example, in contrast to simulations, routing in a real world WSN application is not the main task of a WSN.

Only a fraction of the already limited resources (like RAM and CPU cycles) of a sensor node can be used by the routing protocol. Consequently, the implementation and evaluation of a real-world sensor application using different routing approaches can produce new insights for routing in WSNs.

1.5 Objectives of the Research

This project aims at achieving the following:

1. To study the state-of-the-art technology in wireless sensor networks for smart homes.
2. To propose a network technology capable of collecting relevant information in a smart home.
3. To have a demonstrator for eventually testing the power management strategies of the proposed technology.

1.6 Research Methodology

In order to achieve the aforementioned objectives, the following approaches were adopted:

1. I surveyed the different types of Wireless Sensor Network technology in order to collect information regarding the provision of services.
2. Since the data collection occurs at the sensor nodes, I reviewed the various protocols for wireless sensor network.
3. Finally, a contribution was proposed and implemented using simulation frameworks such as NS3, C++/python to provide an experimental analysis of the behaviour of the algorithm.

1.7 Organization of the Thesis

This work is organized as follows:

Chapter 2 gives an overview of the state-of-the-art technology in wireless sensor network and introduces the various LPWAN technologies for Wireless Sensor Network.

Chapter 3 introduces Network Simulator 3 and then focuses on the structure of the proposed LoRa models. Chapter 4 discusses the implementation of the proposed models and the evaluation of the results. Chapter 5 provides the conclusion and the future work.

CHAPTER TWO :
LITERATURE REVIEW

2.1 State of the Art

In the literature, energy management has been studied in several works. Bədicə et al. (2013) propose energy saving that aims at using sensors and actuators from the Smart Home Environment (SHE) for controlling energy savings by automatically either switching off the appliances not in use or switching them to low-power mode. A user can also set their own preferences. A Residential Load Control (RLC) scheme that is suitable for grids with real-time pricing is proposed in Hussain et al. (2015). The authors focus on an automatic controller that can predict the price of electricity during the scheduling horizon and schedules appliances to provide an optimum cost and waiting time within that horizon. In Mendes et al. (2015) energy efficiency driven Smart Home systems, are homes where remote control and monitoring facilities are possible in smart devices and appliances. They use battery supplied nodes, which work with a limited amount of energy and decision support tools designed to assist users in making smarter decisions by using energy saving services. In Bədicə et al. (2013), Smart Grid integration that addresses the integration of energy-aware SHE into the Smart Grid was considered. In Hu and Li (2013), Smart Home Energy Management System (SHEMS) reduces the consumers electricity bill and flattens demand peaks. In Ali et al. (2012), the authors focus on reducing the peak-to-average electricity usage ratio by finding an Optimal Consumption Schedule (OCS) for the subscribers in a neighbourhood. The authors employ a game theoretic approach. In Torunski et al. (2012), the authors propose an energy management protocol which allows consumers to set a maximum consumption value and the residential gateway is able to turn off the appliances that are in standby mode, or overwriting the user-defined programs with less energy consuming ones. However, defining a maximum value for consumption is not practical and overwriting consumer settings may result in discomfort of the inhabitants. In Mendes et al. (2015), the authors propose a Renewable Energy Management Driven Smart Home.

This includes the use of renewable resources such as solar and wind power with an intelligent power consumption mechanism to ensure the communication between smart home appliances and the smart grid. In Ullo et al. (2010), the authors investigate the use of IEEE 802.15.4 based WSNs in the substations where the time-critical applications have been shown to suffer from delay. On the other hand, WSNs are considered convenient for residential non-time-critical applications. Moreover, the experimental results of Erol-Kantarci et al. (2011) imply that using Zigbee in the AMI and in the smart home increases interoperability

2.2 The Advent of Smart Homes

The interaction between humans and their surroundings can take place in different ways. People usually do everyday activities at home and numerous advantages would be gained if the environment could react to human behaviour and gestures. The smart home is an intelligent space that is able to respond to the behaviour of residents (De Silva et al., 2012). The concept of smart homes has been developed since the 1990s. According to one of the most recent definitions, “a home which is smart enough to assist the inhabitants to live independently and comfortably with the help of technology is termed as smart home” (Satpathy, 2006). In a smart home, all the mechanical and digital devices are interconnected to form a network which can communicate with each other and with the user to create an interactive space”. Alam et al. (2012) define the smart home as an application that is able to automatize or assist the users through different forms such as ambient intelligence, remote home control, or home automation systems. These descriptions confirm that the primary objective of a smart home is to increase occupants’ comfort and make daily life easier (Cook & Das, 2007). This goal might be achieved in two ways: (i) by identifying the relevant human activities and increasing their automation in home environments, or (ii) by using remote home control in order to provide high comfort levels, improve security, facilitate energy management, reduce environmental emissions and save energy (Ding et al., 2011; Saad al-sumaiti et al., 2014).

Smart homes aim to establish a better quality of living by deploying fully-automated control of appliances and providing assistive support (Alam et al., 2012). They allow energy efficiency to be enhanced by adapting the operation of devices to occupancy. In a smart home, users and appliances are connected by an enhanced communication network comprising twisted-pair power lines or fibre optics, which transfer digital signals according to a given communication protocol. Most smart homes have a central communication device, which enables occupants to control home appliances remotely (Madakam et al., 2015). According to Lê et al.(2012), smart homes have the following five fundamental characteristics:

1. Automation: the ability to accommodate automatic devices or perform automatic functions;
2. Multi-functionality: the ability to perform various duties or generate different outcomes;
3. Adaptability: the ability to learn, predict and meet the needs of users;
4. Interactivity: the ability to allow interaction among users; and
5. Efficiency: the ability to perform functions in a convenient manner that saves time and costs.

2.3 Present Smart Home

A smart home has highly advanced automatic systems for controlling lighting, temperature, remote switches, programmable multimedia equipment, monitoring and activating security apparatus and arming and disarming remote facilities and many more. These qualities provide ‘intelligence’ to the environment that makes the home ‘smart’. For example, the washing machine can be set on and off remotely or even scheduled, and its energy consumption measured in order to reduce costs. This way, every standalone machine can be connected in the smart home to operate seamlessly without user intervention. Anything that uses electricity can be connected in a smart home system. In a smart home, all heterogeneous connected devices and appliances such as lighting, heating, air conditioning, TVs, computers, entertainment systems, security and camera systems should be able to communicate with each other, are proactive and controllable remotely, and schedulable by home owners.

The command can be by voice, remote controller, or computer. Remote control and scheduling of the system is possible from anywhere even if no-one is at home.

Figure 2.1: Smart Home Devices

Source: (Smart Home Energy, 2016)

Smart homes consist of various types of devices and applications that should be multifunctional with the intelligent control system, easily synchronizable, energy efficient and secured. In Bregman & Korman (2009), a universal implementation model for a smart home is proposed. The smart home model architecture comprises four modules:

1. Central management unit (CMU)
2. User interface (UI)
3. Home equipment and appliances interface (HEAI)
4. External communication interface (ECI)

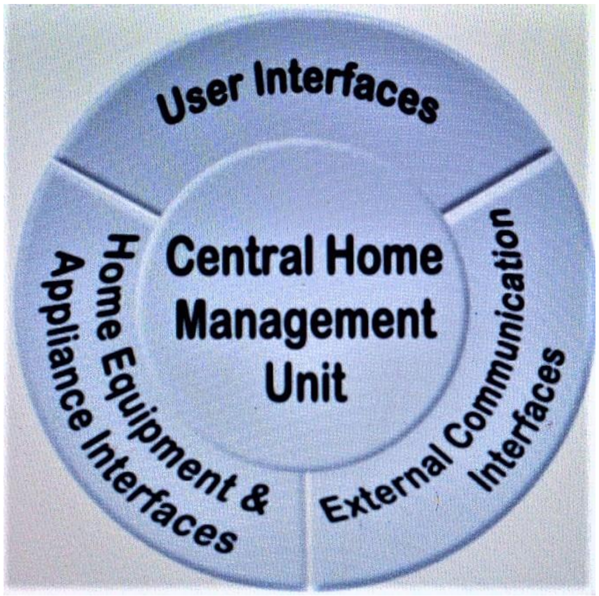

Figure 2.2: The Smart Home Model Architecture

Source: (Bregman & Korman, 2009)

2.4 Benefits of the Smart Home

Every advancement in technology enhances the impressive capabilities of the smart home and makes our lives easier. The smart home is now one of the most significant new trends in digital innovation after the smartphone. Transition to a smart home has considerable benefits. Some of those are listed below:

1. Safety
2. Accessibility
3. Convenience
4. Energy efficiency
5. Cost effectiveness
6. Resale value

2.5 Overview of Wireless Communication Technology

Depending on the requirements of different applications, every technology has its own features and strengths. The following sections show a brief comparison of wireless technologies based on the requirements of robustness, low consumption of power, and throughput data rate.

The IoT paradigm comprises three layers that deal with sensing, network, and data management. Since this work will analyse the proposed solution for the Network portion of the stack, the following sections provide an overview of the different technologies that have been designed so far to connect IoT devices. After this introduction and comparison of rival technologies, the LoRa modulation and the LoRaWAN standard will be focused on in order to describe the mechanisms that will be modelled and simulated in the following chapters.

2.5.1 Solutions for IoT Connectivity

One can identify three main categories of competitors who vary in range, throughput, and cost. Not necessarily happen that only one of these approaches will survive, since each one has its own strengths and weaknesses. Nevertheless, it is clear that there is currently competition between multiple architectures, and that one will prevail and end up providing the bulk of the connectivity to IoT devices.

2.5.1.1 Low-Rate Wireless Personal Area Networks

Low-Rate Wireless Personal Area Network (LR-WPAN) technologies create small networks, typically covering and interconnecting the devices owned by an individual or operating in a house. These standards provide low data rates and short-range communication, in order to focus on efficient battery use.

The IEEE 802.15.4 specification for the PHY and MAC layers provides a starting point for many different solutions that aim at completing the standard with the upper layers, like IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN), Z-Wave and Thread.

At the network level these solutions assume a mesh topology, thus featuring multi-hop connections while increasing the robustness of the connection, which is an advantage. However, using nodes as relays limits their ability to go into sleep mode to save battery power, and routing requires additional computational work. The range of a single hop for these devices is around 10 m, with a raw data rate between 20 and 250 kbits/s, depending on the band that is used. Another notable LR-WPAN standard is Bluetooth. With the recent introduction of Bluetooth Low Energy (BLE), this technology is now able to provide a range of 100 m, and application-level data rates of 270 kbits/s and is thus suitable for applications similar to those of standards based on IEEE 802.15.4.

2.5.1.2 Cellular IoT

Cellular IoT (C-IoT) standards will operate in licensed bands and leverage the existing cellular network coverage to provide internet access to IoT devices: the fact that the infrastructures for the network are already installed is a great benefit and will make deployment time very short.

Currently, three different standards have been proposed (Ericsson, 2016): EC-GSM, LTE-M and NB-IoT. EC-GSM is designed to leverage and improve on legacy EDGE and GPRS systems to provide better coverage and range, with limited power requirements. LTE-M will integrate with LTE to make use of its capacity and performance and bring new power-saving options to increase device battery life. Finally, the new NB-IoT standard will focus on ultra-low-end IoT applications, once again leveraging the existing LTE infrastructures (Ericsson AB, 2016). Future 5G networks are also expected to provide connectivity to IoT devices by design.

2.5.1.3 Low Power Wide Area Networks (LPWAN)

LPWANs have recently been emerging as an alternative to LR-WPANs and CIoT, mainly thanks to the range limitations of LR-WPANs and to the fact that CIoT is still in a very early stage of deployment.

These networks provide wireless connectivity using a star topology and long-range transmission in the unlicensed sub-GHz frequency bands (Centenaro, et al., 2016). The other great benefit brought by LPWANs is increased power efficiency; many of these technologies so far have made the claim of being able to sustain a device for 10 years on a couple AA batteries.

One of the main competitors among standards for this architecture is Sigfox, which employs an Ultra Narrow Band (UNB) modulation capable of sending a 12 bytes' payload in 6 seconds using a 100 Hz band. Due to band regulations, Sigfox currently allows users to send up to 140 messages per day. Messages sent by devices are then picked up by a gateway, which is connected to a centralized Sigfox server. From a device's perspective, access to the wireless medium is not regimented, making Sigfox an ALOHA-like random access scheme. Sigfox networks are expected to provide massive access to devices based on the fact that signals are very narrowband and that transmission frequencies are randomly chosen. Finally, three copies of every packet are sent, at random frequencies, in order to reduce the probability of losing a packet because of collisions or frequency selective fading. Sigfox also limits the number of packets that can be sent downlink from the network to a device to 4 messages per day.

In contrast to Sigfox, LoRa is a technology that exploits a new spread spectrum Physical Layer (PHY) design that enables a higher receiver sensitivity in order to trade data rate for coverage, decreasing the former to increase the latter. LoRa and LoRaWAN are, respectively, a proprietary modulation developed and owned by Semtech Corporation (Semtech, 2015) and a network standard, focused on leveraging useful properties of the LoRa modulation, proposed by the LoRa Alliance. The LoRa modulation allows for very good receiver sensitivities at a contained cheap cost, thus achieving long range transmissions (up to 13 miles in a rural environment) at the price of a reduced data rate, in the $0.3 - 50$ kbps range. At the same time, the LoRaWAN standard that allows multiple LoRa devices to communicate together aims at shifting the burden of administering the network towards a central control point.

This allows devices to be as simple as possible and gives a central coordinator the power of easily tuning each device's parameters in order to accommodate new nodes in the network. Since the work of this thesis is focused on this technology, the following sections will explore how the LoRa modulation and LoRaWAN were designed to meet the IoT requirements listed earlier.

2.6 The LoRa Modulation

LoRa is a proprietary PHY layer technology based on the Chirp Spread Spectrum (CSS) modulation technique. Because of the fact that this technology is patented, no clear description of the modulation is available. Some pieces of information can be found in semi-official documents from Semtech and the LoRa Alliance, such as Semtech (2015) and Engineering (n.d.). This void in the documentation has been filled by a few researchers and hobbyists that analysed and successfully reverse engineered the modulation. The most comprehensive example is Matthew Knight's work (2016).

2.6.1 LoRa's Chirp Spread Spectrum Implementation

The idea behind CSS is that a sinusoidal signal of linearly varying frequency and fixed duration, called chirp, can be employed to "spread" information over a wider spectrum than it would normally need to occupy. This uniform distribution of a symbol over a larger bandwidth provides resistance to frequency-selective noise and interferers at the price of a lower spectral efficiency. Using some additional precautions, CSS can also be more resilient to multi-path interference and the Doppler effect than other more conventional modulations. Assuming that the available frequency band for transmission is B = [f0, f1]. A chirp can be constructed so that it increases linearly in frequency from a starting frequency fs ∈ B to that same frequency, wrapping around from f1 to f0 when hitting the end of the available band. In LoRa, a chirp's starting frequency inside the available bandwidth seems to be used to represent a symbol (Knight, 2016). The number of bits that LoRa encodes in a symbol is a tuneable parameter, called SF.

This means that a chirp using spreading factor SF represents 2^{SF} bits using a symbol, and that there are $M = 2^{SF}$ possible starting frequencies for a chirp. A transmission's spreading factor is also used to determine the duration of a symbol in accordance with the following expression:

Table 2.1 SNR values for different spreading factors

SF	SNR
7	-7.5dB
8	-10dB
9	-12.5dB
10	-15dB
11	-17.5dB
12	-20dB

$$T_s = \frac{2^{SF}}{B} \qquad (2.1)$$

This means that, assuming the modulation is using a fixed bandwidth, an increase of the spreading factor of 1 will yield symbols that last twice the duration. Analogously, a bigger bandwidth increases the rate at which chirps are transmitted, and consequently the bitrate of the modulation. An increase in the transmission time for a chirp (i.e. a symbol) gives the message a higher robustness to interference or noise. On the other hand, this effect may be partially balanced by the fact that for higher spreading factors the number of possible symbols increases, making the occurrence of symbol errors more likely. The reason for this is that achieving synchronicity between the receiver and the signal is especially critical when low data rates are employed. Another disadvantage of transmitting longer messages is the increased probability of collisions.

Because of the reasons above, the choice of SF affects receiver sensitivity, which is defined as:

$$S = -174 + 10\log_{10}(B) + \mathrm{NF} + \mathrm{SNR}\ \ \mathrm{dB}, \quad (2.2)$$

where the first term is due to thermal noise at the receiver in 1 Hz of bandwidth, NF is the noise figure at the receiver (which is fixed for a given hardware setup), and SNR is the signal to noise ratio required by the underlying modulation scheme. SNR values for different spreading factors are represented in Table 2.1, where it can be seen that increasing the spreading factor allows for a better sensitivity.

Given Eqt. (2.1), we can now get the bit rate for a certain pair of SF and *B using a simple computation:*

Table 2.2 Bit rate [bits/s] for a range of spreading factors and bandwidths

SF	125 kHz	250 kHz	500 kHz
7	6835	13671	27343
8	3906	7812	15625
9	2197	4396	8793
10	1220	2441	4882
11	671	1342	2685
12	366	732	1464

$$R_b = \frac{SF}{T_s}. \quad (2.3)$$

The bit rates for a range of spreading factors and bandwidths can be found in Table 2.2.

2.6.2 LoRa Physical Layer Packets

An example of a LoRa packet can be seen in Figure 2.3, which shows a spectrogram representation with time on the horizontal axis and frequency on the vertical axis.

It can be seen that a PHY layer LoRa message consists in the chirp signal sweeping the frequency band. After some repetitions of this frequency sweep, that constitutes a preamble (whose minimum length is 4.25 chirps), data is encoded in the signal as instantaneous changes in the frequency of the chirp, or lack thereof. A decoding process is proposed in Knight (2016), and it consists in first “de-chirping” the signal, and then taking a Fast Fourier Transform (FFT) of the signal, with a number of bins equal to the number of symbols *M* that corresponds to the used spreading factor. Figure 2.4 shows the de-chirped version of a LoRa transmission, again with time on the horizontal axis and frequency on the vertical axis. The signal can now be interpreted as if it were modulated using Multiple Frequency Shift Keying (MFSK). By taking multiple overlapping FFTs and looking at the bin with the highest power content we can now detect the symbol in each time frame.

Figure 2.3: Spectrogram representation of a LoRa signal

Source: (Knight, 2016)

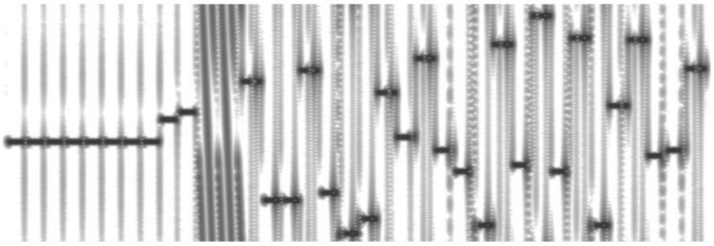

Figure 2.4: De-chirped version of a LoRa signal

Source: (Knight, 2016)

Aside from the modulation itself, LoRa also specifies a set of encoding operations that are applied before modulation and transmission:

1. **Data whitening:** Is used in order to shrink the probability of long equal bit runs occurring in the data. Whitening also helps distribute information across the whole bandwidth of the radio channel. It should be noted that Knight (2016) found the whitening sequences specified in Whitening (2013) to be different from the version that is actually implemented in the chips, but was still able to find out the right whitening sequence.
2. **Forward Error Correction (FEC):** Is implemented as a Hamming Code. The code's information word length is fixed at 4 bits, and the length of the code-word is a tuneable parameter in the [5,8] bits range. The code rate for a LoRa packet, then, is $C \in \{4/5, 4/6, 4/7, 4/8\}$.
3. **Interleaving:** Scrambles the output of FEC to make the code more resistant against bursts of error. Once again, Knight (2016) found that the documents that are available online differ from the chip implementation. Through reverse engineering, it was found that the modulation uses diagonal interleaving, with the two most significant bits reversed.
4. **Grey mapping:** Is finally used to map a block of *SF* bits into one of the *M* symbols in the constellation, while making sure that two adjacent symbols differ by at most 1 bit, in order to increase the channel code's chances to correct possible errors.

The on-air time of a packet can be computed with the following formula (Semtech, 2015):

$$\mathbf{t_{packet} = t_{preamble} + t_{payload}} \qquad \textbf{(2.4)}$$

where t_{preamble} is the time it takes to transmit the preamble, and t_{payload} is the time it takes to transmit the actual data. These two entities have the following expressions:

$$\mathbf{t_{preamble} = (n_{preamble} + 4.25) \bullet ts} \qquad \textbf{(2.5)}$$

$$\mathbf{t_{payload} = n_{payload} \bullet ts} \qquad \textbf{(2.6)}$$

where n_{preamble} is a configurable parameter that affects the number of symbols in the preamble (and thus the probability that a receiver will detect an incoming packet, at the cost of a longer time on air).

The computation of n_{payload} is more complicated, since it depends on many different parameters:

5. PL is the number of payload bytes;
6. H can be either 0 when the PHY header is disabled or 1 when it is set to enabled. The PHY header is used to carry information about the packet length, enabling variable payload sizes. This information can be omitted to save on-air time if the transmitter and receiver both know the duration of a packet.
7. DE can be either 0 when the low data rate optimization is disabled or 1 when it is enabled. Low data rate optimization is a measure used to counter clock drift when sending very long symbols, and to achieve correct time synchronization between transmitter and receiver.
8. CR is the number of added parity bits.

Given the above parameters, the number of payload symbols becomes:

$$= 8 + \max\left(\left\lceil \frac{8\mathrm{PL} - 4\mathrm{SF} + 44 - 20\mathrm{H}}{4\left(\mathrm{SF} - 2\mathrm{DE}\right)} \right\rceil \left(\mathrm{CR} + 4\right), 0\right)$$

n_{payload} (2.7)

2.6.3 Spreading Factor Orthogonality

One very powerful feature of the LoRa modulation is that different spreading factors are pseudo-orthogonal, even when the same centre frequency and bandwidth settings are used. This allows a receiver to correctly detect a packet using spreading factor i even if it is overlapping in time with another transmission employing spreading factor j, as long as $i \neq j$ and the received packet's Signal to Interference plus Noise Ratio (SINR) is above a certain threshold (also called isolation) that depends on both i and j. This pseudo-orthogonality between different packets allows a network employing LoRa devices to exploit different spreading factors to achieve a higher throughput with respect to more traditional modulation schemes, in which a collision can cause the incorrect reception of both the intended packet and the interferer. While the exact figure for the isolation margin is never explicitly stated in Semtech documents, in Gorce et al. (2015) this was investigated and some estimates were made based on a model of the LoRa simulation.

2.6.4 Main Semtech Chips and Independent Implementations

Since the technology is proprietary, commercially available chips implementing the LoRa modulation chain are currently only available from Semtech Corporation. There are two main kinds of LoRa radio chips: the SX1272 and SX1273 are the most basic chips, intended for simple LoRa devices, while the SX1301 is able to decode multiple packets on different frequencies simultaneously. This chip is intended to be used in aggregators that can pick up transmissions from a whole network made of simpler LoRa devices. The scheme for an SX1301 receiver can be seen in Figure 2.5: the chip has 8 built-in configurable dynamic data rate (DDR) *receive paths*. These blocks are able to work concurrently in order to decode different overlapping transmissions using any spreading factor and centre frequency (Fsk & Lora, 2017), leveraging the pseudo-orthogonality between different spreading factors to distinguish between packets arriving at the antenna simultaneously and decoding each one correctly.

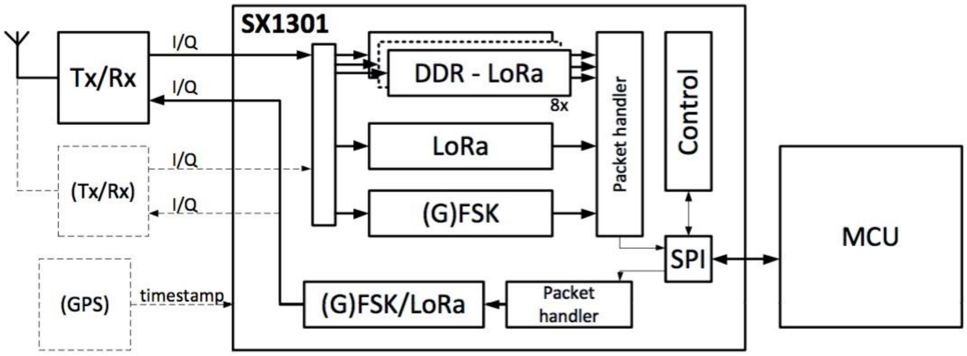

Figure 2.5: Logic scheme for the SX1301 chip

Source: (Semtech Corporation)

Aside from the official ven22223dor’s chips, some software-defined radio (SDR) projects devoted to implementing an open LoRa PHY layer recently came to life: the most notable examples are by Knight (2016) and Blum (2016). Both implementations leverage the software development toolkit GNU Radio. Researchers and hobbyists often create these open implementations in order to explore the modulation: these tools can be used to identify ways to improve the modulation and to expose some of its defects or security weaknesses.

2.7 The LoRaWAN Standard

While the LoRa PHY layer is proprietary, the rest of the protocol, known as LoRaWAN, is open and described in (Semtech, 2015) by the LoRa Alliance, a group of vendors and research institutions that are interested in spreading and leveraging LoRa technology. What follows is a brief overview of the main components of a LoRaWAN and of the frequencies. It was decided the networks would operate in different parts of the world.

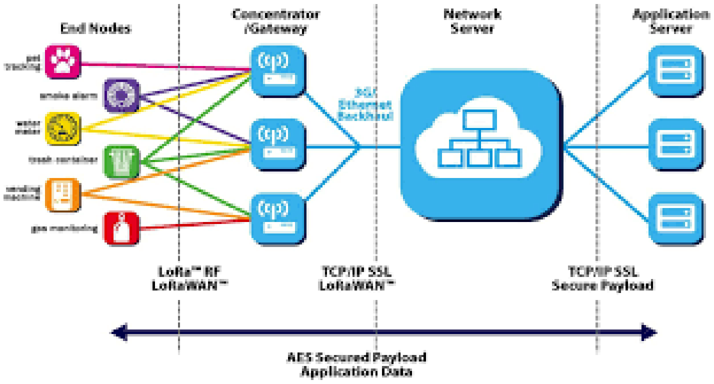

Figure 2.6: Sample topology of a LoRa network

Source: (LoRa Alliance)

2.7.1 Topology and Device Classes

LoRaWAN networks are laid out in a star-of-stars topology, in which end devices (EDs) send/receive messages wirelessly to/from one or more gateways (GWs), the gateways in turn relay them, to a centralized network server (NS) via a high throughput and reliable link.

This topology, as represented in Figure 2.6, actually allows one ED to deliver messages to more than one gateway. In fact, EDs are not explicitly paired with a single gateway: messages are simply sent by the devices on the wireless channel, assuming that at least one gateway will receive them and forward them to the NS. Then the centralized system has the responsibility of filtering duplicates and picking the most appropriate gateway through which to send downlink messages to that device. In order to increase the network's robustness to interference, multiple logical channels are defined for the whole network and devices needing to transmit a packet are required to pick a channel in a pseudo-random fashion.

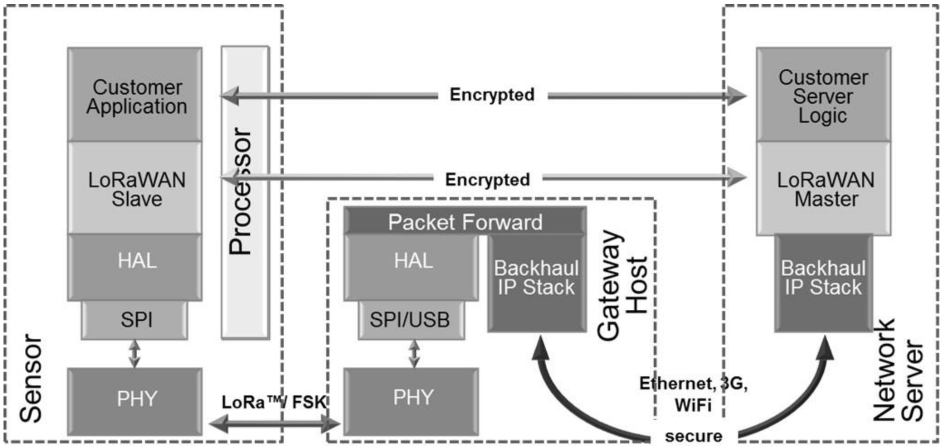

Figure 2.7: Protocol stacks of the various devices in a LoRaWAN
Source: (LoRa Alliance)

The protocol stacks of EDs, GWs and of the NS is represented in Figure 2.7. While the ED and NS stacks have an application layer, gateways are only tasked with forwarding messages between the sensor (i.e., the EDs) and the NS, and are thus very transparent to the end devices' application, which is connected directly to the one on the NS.

According to Semtech (2015), a LoRa ED can behave according to one of the following models:

1. **Class A** is the default operation mode of LoRa end devices. Each end device transmits packets coming from upper layers on the wireless channel in a very asynchronous fashion, thus implementing an Aloha Medium Access Control (MAC).

After each uplink transmission, the node, waiting for any command or data packet returned by the NS, opens, at most, two reception windows. The first window is opened on the same channel as the node's uplink communication, while the second window is opened on a different sub-band (previously agreed upon with the NS, and modifiable through MAC commands) in order to increase the resilience against channel fluctuations. This class is expected to be implemented by devices whose energy budget is limited and therefore should keep the radio transceiver off as long as possible.

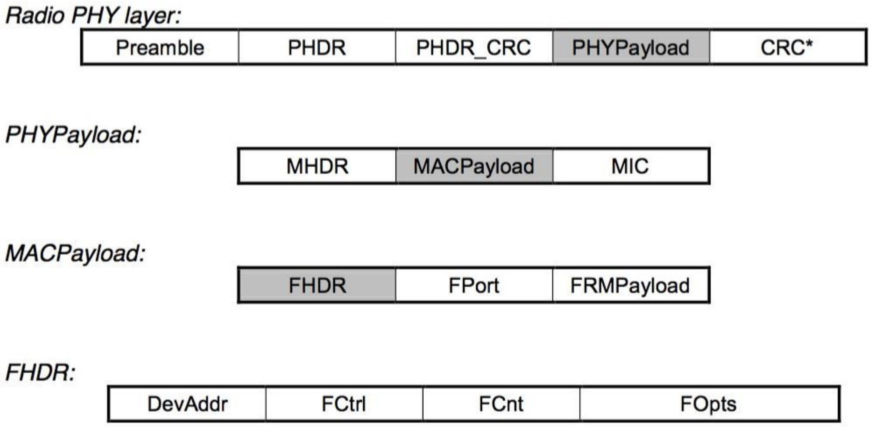

Figure 2.8: Packet structure of a LoRaWAN message
(Source: Google)

2. **Class B** end devices are synchronized with the NS by means of beacon packets that are broadcast by Class B gateways. Thanks to the beacon mechanism, they can receive downlink data or command packets in specific time windows, irrespective of the uplink traffic. Class B is intended for end devices that need to receive commands from a remote controller, e.g., switches or actuators.
3. **Class C** is defined for end devices without strict energy constraints (e.g., devices that are connected to the power grid), which can hence keep the transceiver constantly on, waiting for downlink messages. This class is expected to be useful to devices operating an actuator and having stricter delay requirements.

2.7.2 Packet Structure and MAC Commands

In addition to the topology of the network, Semtech (2015) also describes the communication protocol. This includes the format of PHY and MAC layer packets, a set of network parameters, like the SF and channel frequencies used by an end device, and the MAC commands which must be used to tune the aforementioned parameters. The packet structure of a LoRaWAN message can be seen in Figure 2.8: at the PHY layer, the packet is made of a preamble, a header, a payload and two CRC codes to protect header and payload. Inside the PHY payload, the MAC header contains information about the version of the standard the device is using and about the type of message:

- A join packet is the first packet that is sent by a device attempting to enter a network.
- Data messages can be either uplink or downlink. Additionally, it is stated whether the message requires an acknowledgement or not.
- Proprietary messages can be used to implement non-standard message formats that can be used between devices having a common understanding of some proprietary extensions.

The MAC payload, instead, contains a frame payload, a frame port, and a frame header. The Frame payload typically contains data coming from the application layer, and the frame port field is used to identify which application the message is intended for. Some port numbers are reserved for standardized application extensions that will be created in the future. The frame header, instead, contains various fields that are tied to different aspects of a LoRa network. First, a short 4-byte device address is used to identify a device in a network. One byte, called the frame control field, is intended to house the acknowledgement (ACK) and frame pending bits. In addition to these, 2 more bits are used to implement the adaptive data rate (ADR) feature: the NS has the ability to ask an ED to modify the spreading factor it employs for transmission. The ADR algorithm may choose to increase the spreading factor of a device whose SNR margin is too low or decrease it in case the device’s transmissions arrive consistently well above sensitivity.

In other words, the mechanism needs to ensure the reliability of a device's link (typically by using high SF values) while minimizing the on-air time of the device's packets (by using lower SFs) to avoid collisions. It should be noted that the ADR algorithm is not standardized. This means that implementations that weight efficiency and complexity in different ways are possible and open to being compared. If the ADR bit is set, the device knows that the NS is controlling its data rate, and that it will find further instructions among the MAC commands contained in the FOpts field.

Various MAC commands that can be exchanged between the EDs and the NS enable the following features:

- Link checking: a device can ask its link margin to the NS, which replies with the power that was received at the gateway.
- ADR requests and replies.
- Duty Cycle: the NS can set limitations to a device's aggregated transmission time.
- Setting of parameters regarding the received windows.
- Device status: the NS can ask a device for its battery level and demodulation margin.
- Creation of new radio channels.

Additionally, to the above features, 128 more command identifiers are reserved for proprietary network command extensions.

2.7.3 Encryption and Device Activation

According to Semtech (2015), if a data frame carries a payload, the frame payload must be encrypted. The encryption scheme employed by LoRaWAN is based on the algorithm that is also used by the IEEE 802.15.4 standard (Davide, 2016), using an AES cipher, and a message integrity code (MIC) to prevent attackers from tampering with the encrypted message without the receiver noticing.

The key used in the AES procedure can be either a network-wide or an application-specific key, depending on the frame port to which the message refers. An ED first obtains both these keys during its activation, a procedure by which a device first accesses the network, also obtaining a 32-bit address, that uniquely identifies it inside the network, and an application identifier. The LoRaWAN standard provides two ways in which a device may perform activation in a LoRaWAN:

1. Over-the-air activation (OTAA) allows a device to be activated by exchanging a series of messages with the NS in order to obtain a network key over the air in a secure way;
2. Activation by personalization (ABP) implies that the device comes preconfigured with an address and both the network and application keys, thus being able to access a predefined network without exchanging data with the NS in join procedures.

2.7.4 Frequency Bands

Three regions: Europe, China and the United States, have allocated fixed frequencies in which LoRaWANs are expected to operate. These frequencies are shown in Table 2.3. For each one of these regions, the standard mandates customized parameters that define the preamble, channel frequencies, allowed spreading factors, maximum payload size, receive windows and join procedures to make sure that LoRaWAN complies with local legislation.

Table 2.3 Frequency bands for various regions (Source: (ETSI, 2002))

Region	Frequency band [MHz]
Europe	868–870
US	902–928
China	779–787

Note that all chosen frequency bands are in the 780–930 MHz range. For a LPWAN needing to focus on achieving the widest range possible, this set of frequencies is preferable to the 2.4 GHz and 5 GHz ISM bands that are leveraged by IEEE 802.11 standards, because attenuation is lower.

2.7.5 Notable LoRaWAN Implementations

Some fully operational LoRa networks have already been successfully deployed. Senet is an American company that offers a subscription-based service, allowing 50 million clients to connect to a wide LoRa network covering more than 225 cities in the United States, including Los Angeles, New York City, Washington D.C., Chicago, Philadelphia, Dallas, Seattle, San Diego, Atlanta, Denver, San Francisco, Boston and San Jose, . Typical applications include energy management, irrigation, parking, and water monitoring. Another notable implementation of LoRaWAN is The Things Network, a crowdsourced solution for the IoT leveraging LoRa technology, which allows anyone to install their own gateway and connect it to the organization’s central network server to increase the network’s coverage. The City of Amsterdam, where the movement started, was covered in 4 weeks and is currently served by more than 50 gateways. Local communities around the world are encouraged to grow and deploy LoRa gateways, which can be purchased for a relatively small price from the organization’s website.

2.8 European Regulations

In order to understand some of the limits that LoRa networks have to respect, it is necessary to look at how the “free” bands that these networks use is regulated by various entities. The fact that a band is unlicensed does not mean that it is also not regulated. There are essentially three levels at which different organizations handle spectrum allocation and limit its use:

1. At the *national level*, the frequency spectrum is managed by national administrations, which:
    i. Compile a table of spectrum allocations;
    ii. Define a framework for use of these bands; and
    iii. Assign each band to different users, possibly via licences.

2. At the *European level*, there are three organisations that cooperate to regulate spectrum usage:

   i. The European Commission (EC);
   ii. CEPT’s Electronic Communications Committee (ECC); and
   iii. The European Telecommunications Standard Institute (ETSI).

3. At the *worldwide level*, the International Telecommunication Union (ITU) coordinates regional and national organisations.

The two most important documents which cover the ETSI and ECC regulations are ETSI (2002) and Engineering (n.d.).

LoRaWANs operate on industrial, scientific, and medical (ISM) bands: in the $902 - 928$MHz band in the US and the $863 - 870$MHz band in Europe (ETSI, 2002). These licence-exempt bands are subject to regulations of radio emissions (Actility, 2013). Thus radios are required to either adopt a listen-before-talk (LBT) policy or duty cycled transmission, in order to limit the rate at which the end devices can actually generate messages. The latter policy is adopted in the vast majority of cases. Furthermore, each transmission in a sub-band of the $863 - 870$MHz frequency range must respect some limitations on the emitted power. The following sections explore each one of these limits, and finally list a channel line-up for LoRaWANs, specifying the limits that LoRa devices must respect in each channel.

2.8.1 Effective Radiated Power (ERP) Limitations

There are mainly three ways to describe the emissions radiated by a device:

1. The electrical field strength (E) at a specified distance from the transmitting antenna.
2. The Effective Isotropic Radiated Power (EIRP), defined as the power that would have to be used on an isotropic antenna in order to get the same field strength that the tested device produces at the same distance. Conversion between field strength and EIRP is obtained using the following formula:

$$\text{EIRP}_{\text{dBm}} = 10\log\left(\frac{E^2 \cdot r^2}{0.03}\right) \quad (2.8)$$

The ERP is similar to the EIRP but uses a half-wave dipole instead of an isotropic antenna. Since the gain of a half-wave dipole is at most 2.15 dBi (dB relative to an isotropic radiator), the following holds:

$$\text{ERP} = \text{EIRP} - 2.15 \quad (2.9)$$

ETSI uses the ERP metric to evaluate the emissions of a device. The way that power must be measured is described in two different clauses, based on whether the device under test includes an antenna or not. In the case the device is provided without an antenna, the vendor is also required to state the maximum gain of an antenna that can be connected to the device. In this case, the power must be measured at the connector, and the antenna gain needs to be factored in. In the case the device is shipped with an integrated antenna, the effective radiated.

Table 2.4 Maximum on and minimum off times based on duty cycle definition

As specified in (ETSI, 2002)

Duty Cycle	Max “on” time/hour	Max “on” time	Min “off” time
≤ 1.0%	36	3.6	1.8
≤ 10%	360	36	3.6

Power is the power radiated in the direction of the maximum field strength, under specified conditions of measurement. ETSI (2002) also describes how the test to measure the radiated power should be performed: it specifies the setup of the antenna and receiver, polarization, height and other factors intended to make sure the measurement is performed in the direction where the antenna is emitting the most power.

2.8.2 Duty Cycle Limitations

Duty cycle is defined as the ratio, expressed as a percentage, of the maximum transmitter “on” time over one hour, relative to a one-hour period. Duty cycle limitations are given in Goursaud & Gorce (2015), clause 7.10. These limitations apply to every receiver, excluding those with LBT capabilities. Since LoRaWAN has defined no LBT, all LoRa devices must respect the mechanism as of now, according to these limits.

Table 2.4 contains a couple of examples taken from Engineering (n.d.), illustrating how a duty cycle limitation translates to a certain maximum time on air and minimum waiting time between consecutive packet transmissions:

For example, a device with a 1% duty cycle can perform 10 transmissions of 3.6 seconds within one hour, while a device with a duty cycle of 10% can perform 10 transmissions of 36 seconds within one hour.

2.8.3 Channel Line-up

Three different ISM sub-bands can be distinguished in Europe (ETSI, 2002), in the range specified by:

1. The g1 line 3 ($867 - 868$MHz), g1.1 ($868 - 868.6$MHz) and g1.4 ($869.7 - 870$MHz) sub-bands, with a maximum of 36 seconds per hour time on air

Table 2.5 Channel line-up for LoRa according to ETSI regulations

S/N	*F*	*B*	% of time on air	Max ERP	Regime
1	868.1	125 kHz	1%	14 dBm	g1.1
2	868.3	125 kHz	1%	14 dBm	g1.1
3	868.5	125 kHz	1%	14 dBm	g1.1
4	868.85	125 kHz	0.1%	14 dBm	g1.2
5	869.05	125 kHz	0.1%	14 dBm	g1.2
6	869.525	125 kHz	10%	27 dBm	g1.3

(ToA), 1% duty cycle to be shared between its sub-channels and an ERP limit of 14 dBm;

1. The g1.2 (868.7 − 869.2MHz) sub-band, with a maximum of 3.6 seconds per hour ToA and 0.1% duty cycle to be shared between its sub-channels, ERP limit of 14 dBm;
2. The g1.3 sub-band: (869.4 − 869.65 MHz) with maximum 360 seconds per hour ToA and 10% duty cycle, ERP limit of 27 dBm.

Table 2.5 proposes an example line-up of 6 LoRa channels according to ETSI constraints on European ISM bands, taken from Actility (2013). It is important to note that, as long as the regulations in each frequency band are respected, an end device is allowed to transmit on different channels, contained in different sub bands, in order to increase the aggregate, while still respecting the duty cycle limit in each sub-band.

CHAPTER THREE
METHODOLOGY

3.1 Implementation

To perform the network simulations of a LoRa system, the Network Simulator 3 (NS3) software ("ns-3 Tutorial," 2017), an open source discrete event simulation (DES) suite, has been used. The simulator has been expanded with the creation of a LoRa module that implements the various models described in Chapter 2. This chapter first provides a brief introduction to the NS3 software and then describes the structure and implementation of the new LoRa module.

3.2 Network Simulator 3

NS3 is a network simulation software intended for research and educational use, licensed under the GNU General Public License (GPL) and developed by a community of users. By combining several C++ objects, with each class modelling an aspect of a network, NS3 is able to simulate complex networks in a detailed and realistic fashion. Classes that model related concepts or systems are grouped in modules: as an example, the Wi-Fi module contains several classes that model the components of a Wi-Fi system, such as access points, Wi-Fi enabled devices, the Wi-Fi MAC layer and a wireless channel. These classes are interconnected and, when combined with some other modules that model core functionality, device mobility, propagation and so on, they can be used to simulate a whole network implementing the Wi-Fi standard.

The fact that NS3 works according to the DES principle means that a simulation consists of a series of *events*, each one tied to a certain time. The simulator's task is to execute these events (i.e., perform the appropriate function call that is linked to that event), which results in a change of the state of the simulation and possibly in the scheduling of more events.

An example of an event is the transmission of a packet on a wireless channel: this is represented by a function call from the class representing the device's PHY layer to the one representing the channel, which in turn will schedule an event representing reception of the packet by another device's PHY layer, possibly after a channel delay. Once an event is executed, the simulator moves on in the events list to perform the next function call. Since one event may schedule multiple other events, it is not guaranteed that at some point the simulation will end. In such cases, a special stop event can be issued to terminate the simulation. NS3's event-driven approach ensures that, even if a simulation features only two events that are scheduled far apart from one another, the simulator will execute one directly after the other. This approach speeds up simulations while still keeping them realistic, since no change in the state of the system was scheduled to happen between the two events.

NS3 also provides a built-in pseudo random number generator (PRNG), namely the MRG32k3a by Pierre L'Ecuyer (L'Ecuyer et al., 2002). The generator provides $1.8 \cdot 10^{19}$ independent streams of random numbers, each one consisting of $2.3 \cdot 10^{15}$ sub-streams. Each sub-stream has a period of $7.6 \cdot 10^{22}$: this means that the period of the generator amounts to $3.1 \cdot 10^{57}$. Random variables are provided as objects that access one of the independent streams of the underlying random number generator and output random numbers according to a certain distribution described by a set of parameters. The fact that each random variable is assigned a different PRNG stream ensures that there will be no correlation between random variables. A system to perform different "runs" of a simulation, ensuring that each repetition makes use of a distinct stream, is also available.

NS3 also features a tracing system, used to monitor variables during the simulation, and optionally trigger an action when a change is detected. This system is used to gather data during simulations on optimized runs when logging is disabled and no data can be gathered through the program's standard output.

While the models that are specific to a network or protocol are implemented in a module's classes, the topology and models that are to be used in a specific NS3 simulation are described via a C++ or Python program, which typically follows the following steps:

1. ***Creation of a topology*****:** the set of nodes (i.e., devices) that will be used in the network is instantiated as a collection of node objects. A mobility model may be associated to each node, to represent the node's physical position and how it changes with time.
2. ***Models*****:** a certain protocol stack is installed on the previously created set of nodes. This is usually done via *helpers*, classes that are specialized in installing on a node the various objects implementing the required layers of the ISO/OSI stack. This step gives each node the ability to create, send, receive and interpret packets belonging to that protocol.
3. ***Configuration*****:** the models of a protocol are configured to use certain values as their parameters, and links between different nodes are created. Usually, this is done by "subscribing" multiple nodes' PHY layers to the same channel object.
4. ***Execution*****:** the simulation is started, and the simulator class goes through the events and executes the corresponding function calls. During the simulation, trade sources fire and save data either in appropriate data structures or in a file. In some cases, it can be useful to stop a simulation (i.e., schedule a stop event) once enough data has been collected from trace sources.
5. ***Performance analysis*****:** after the simulation is stopped the gathered data can be analysed and visualized.

The next section will focus on explaining how the models for a LoRa network have been created and describes the various ways they can be configured.

3.3 The LoRa Module

In order to model the behaviour of a LoRaWAN, a new LoRa module was created. This module is essentially a collection of classes that work together to describe the behaviour of LoRa EDs and GWs at various levels, from the PHY to the application layer. The set of classes that are needed to simulate the protocol stack on a device can be seen in Figure 3.1. Additionally, to those classes representing a layer in the stack (LoraPhy and LoraMac), some other classes were used to model aspects of the system like losses caused by buildings, correlated shadowing, interference and duty cycle limitations. The structure and interactions of the code will be described briefly in the following section, starting from the upper layers and ending with the class representing the wireless channel.

The work described in this thesis is focused on the ED and GW implementations, since it's at this level that the LoRa modulation is employed. Creation of the NS was not considered a priority, since investigation of the effects of ADR and MAC commands on the network were not part of this thesis' objectives.

3.3.1 PeriodicSender

The application layer class ***PeriodicSender*** consists of a packet generator which creates zero-filled packets of a randomized payload size. It should be noted that creating more "realistic" payload contents would not affect the simulation results, since our link models don't account for the contents of the packets in the link abstraction.

The application transmission period m_interval determines the delay of the "transmit a random packet" event that is scheduled right after a transmission and can be set up as an attribute of the class. Note that, at the application level, transmission simply means that the packet is forwarded down to the LoRa MAC layer ***(PRINCIPLES OF COMMUNICATIONS NETWORKs and SYSTEM*S, n.d.)**.

Since the function call that transmits a packet also schedules the next transmission, this application will keep sending packets until it is stopped via a specific function call. When the application is first started on a node, a random delay for the first packet sending event is chosen via a random variable: $d \sim$ U([0,m_interval]).

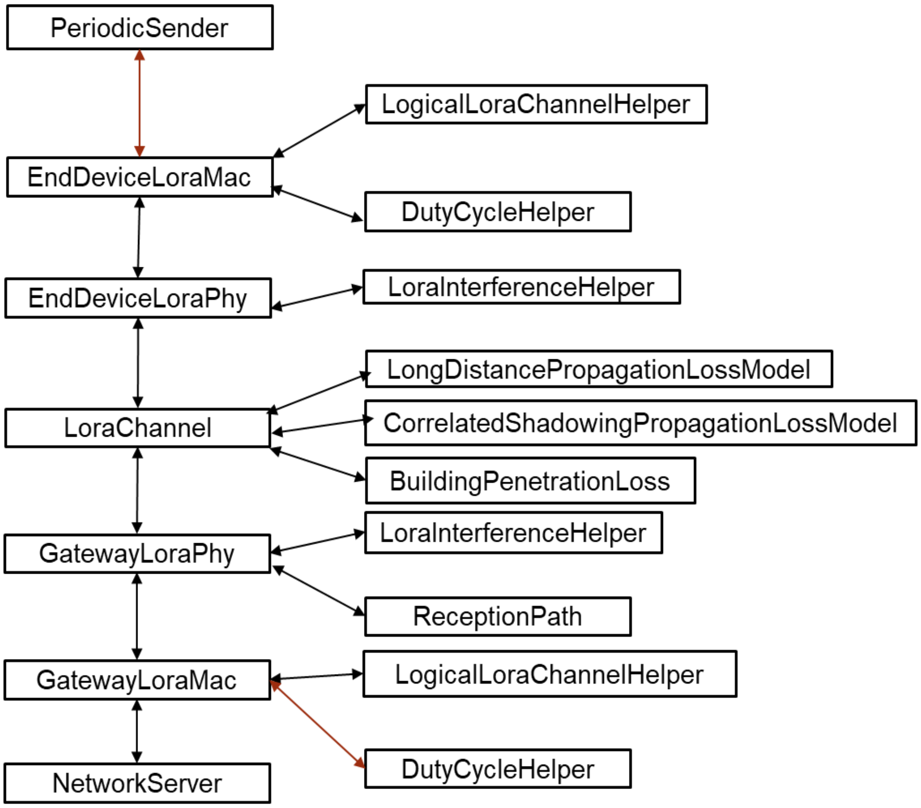

Figure 3.1: The LoRaWAN stack as it was represented in the LoRa module

3.3.2 End Device LoRaMac

The LoRaMac class models the MAC layer of a LoRaWAN device. This class is used to keep track of the available network channels via a Logical Lora Channel Helper object, and to account for the duty cycle limitations demanded by the regulations: it’s this class’s responsibility not to send messages coming from the upper layer if this would mean breaking the duty cycle rule, and to queue them and send them at a more appropriate time. In order to keep track of different duty cycles on different sub-bands and to separate the duty cycle logic from the class, a Duty Cycle Helper class was created.

More specifically, the two subclasses End Device Lora Mac and Gateway Lora Mac implement behaviours that are specific to an ED and a GW, respectively.

End Device Lora Mac in that it implements a simpler, forward-only mac layer. These classes are also responsible for the interpretation of MAC commands that are either piggybacked in the FOpts field or contained in the FRM Payload, even though these functionalities are not implemented now. LoRa packets, with their specific structure, were implemented as extensions of the basic packet class.

Procedure 3.1 Sending procedure in the End Device LoRaMac class

Input: The packet variable contains the application-level payload

```
Add the Frame and Mac headers to packet
$M$ = number of available channels
Channels = $[1,\ldots,M]$
Shuffle channels
Shortest Wait Time = $\infty$
For every channel c in the channels list
t = wait time on the c channel, taken from Duty Cycle Helper
    If $t <$ Shortest Wait Time
        Shortest Wait Time = $t$
    If $t = 0$
        Send packet to the device's LoraPhy class
        Notify Duty Cycle Helper of the new transmission
  Return
        Cancel previous send event
        Schedule retransmission of packet after Shortest Wait Time seconds
Return
```

3.3.3 EndDeviceLoraPhy

The EndDeviceLoraPhy class models the physical layer of a Semtech LoRa device. Specifically, this is the class that simulates the behaviour of the SX1272 and SX1301 LoRa chips in EDs and GWs, respectively. When the device has to send a message, this class's role is to take the packet from the MAC layer and deliver it to the channel class. Furthermore, it decides whether a packet obtained from the channel is correctly received, based on its power and the interference the device is experiencing.

The class makes use of an m_state attribute, representing the state of the chip that can take one of the following values:

1. TX when transmitting a packet;
2. RC during reception of an incoming packet;
3. IDLE when listening for incoming packets; and
4. SLEEP when in low power consumption mode.

Each one of the states above can be linked with a different voltage and current consumption by the energy model, which takes note of the energy expenditure of a device and can subsequently give an estimate of the device's battery life. Even though the energy model has not yet been fully integrated in the NS3 simulator, particular care was taken in the design of the classes so that combining the LoRa and energy models in future iterations of this work will be trivial.

As with the design of the LoraMac classes, LoraPhy also features two subclasses, End Device LoraPhy and GatewayLoraPhy, which represent in greater detail the PHY layers of EDs and GWs. Both implementations compute interference similarly, via a LoraInterferenceHelper class that keeps track of every signal that arrives at the device, and then uses this knowledge to compute whether or not a given packet was impaired by interference.

Algorithm 3.2 shows the steps that are taken by an EndDeviceLoraPhy object whenever it is notified by the LoraChannel class that a signal is arriving at the antenna. First of all, the object performs an Add call to its LoraInterferenceHelper instance, so that the helper will create an instance of the Event class to represent the signal, and add it to a list that keeps track of all events (i.e., packets) that arrived at that device. An Event holds all the information that is needed to perform interference computations, namely the time window during which the packet was impinging on the antenna, its spreading factor, received power and logical channel it was sent in. After the packet is kept track of by the interference helper, its power is compared with the device's sensitivity to that spreading factor, and the end of the reception event is scheduled only if the device is, in fact, able to receive the packet. Furthermore, the above procedure is performed only if the underlying PHY is in its IDLE state. If the device is sleeping, receiving or transmitting; reception of another packet is impossible. It must be noted that, even if the signal is under the sensitivity, an event is still added to the list of interferers, since it will still need to be accounted for when computing. This reasoning also applies to the state of the device: even if a device is sleeping, arriving signals must still be registered since they would become influential if the device were to wake up and start listening before the signal ends.

Procedure 3.2 Beginning of reception of a packet

```
Input: packet = payload of the packet that is being received
Input: sf = spreading factor of the signal
Input: sensitivity = minimum power, in dBm, that is needed to receive a packet of SF = sf
Input: rxdBm = reception power of the signal
Input: d = duration of the signal
1: Notify the LoraInterferenceHelper of the impinging signal with spreading factor sf
2: if state = IDLE
3:     if rxdBm < sensitivity
4:         return
```

```
    else
        Switch to the RX state
        Schedule the end of the reception of the packet after d seconds
return
```

The end of the reception procedure is shown in Algorithm 3.3: first of all, the EndDeviceLoraPhy's LoraInterferenceHelper instance is queried to ascertain whether the current packet was destroyed by interference or not. Based on this function call's result, the device either forwards the correctly received packet up the stack or does not, before going into SLEEP mode in either case. The procedure that is used to determine whether a packet suffers fatal interference by other signals can be seen in Algorithm 3.4: the set of interfering signals is grouped according to the spreading factor, then for each SF the cumulative interference energy is computed as the sum of the energies from each signal. Energies are obtained as the product of received power and overlapping time with the intended signal.

The implementation of the GatewayLoraPhy class is slightly more complicated than that of EndDeviceLoraPhy, since multiple reception paths need to be implemented. In order to do this, the GatewayLoraPhy class features, as a member variable, a list of ReceptionPath objects. This class represents a reception path: its variables indicate whether or not it is receiving an Event as well as the channel to which it is listening. When the StartReceive method of a GatewayLoraPhy is called by the LoraChannel, the procedure is similar to that represented in Algorithm 3.2, but can be performed only if the reception path list contains an instance of ReceptionPath that is free and listening to the channel of the incoming packet. If the arriving transmission is above the sensitivity, the reception path is marked as busy and linked to an Event representing the signal. Similarly, the EndReceive procedure is similar to Algorithm 3.3, but contains some added steps to free the reception path that was linked to the incoming transmission.

Procedure 3.3 End of reception of a packet

```
Input: packet = payload of the packet that is being received
Input: sf = spreading factor of the signal
Input: rxdBm = reception power of the signal
Input: d = duration of the signal
lost = whether packet was destroyed by interference or not (as determined by LoraInterferenceHelper)
if lost
    Switch to SLEEP state
else
    Pass the correctly received packet up to the LoraMac class
    Switch to SLEEP state
return
```

3.3.4 LoraChannel

The LoraChannel class models the wireless channel that is shared by all devices in a LoRaWAN: this class is responsible for taking packets that a PHY layer wants to transmit and delivering them to a set of LoraPhy objects, with a reception power that is computed according to the Link Measurement Model. During the configuration phase, LoraPhy objects are “added” to the channel object, which registers them in a list. The LoraChannel class allows interconnection between registered PHYs via two methods: Send and Receive. When a PHY needs to send a message in the channel, it can do so by calling the Send function with parameters such as the spreading factor of the message, the PHY-layer packet itself, the duration, transmission power and channel number. Upon calling of this method, the channel goes through the list of PHYs, and schedules a Receive event for those nodes that are registered as listening to that communication’s channel. In order to determine the time at which to schedule a given Receive event, the channel uses a PropagationDelayModel, a default NS3 abstract class that, given the MobilityModel (i.e., the positions) of the transmitter and the receiver can compute the delay according to various models.

In this thesis, the delay model that was used is the ConstantSpeedPropagationDelayModel, which simply computes the time of flight given the distance between the two devices. The Receive event is scheduled with a set of parameters that the target PHY needs to know in order to perform the set of function calls described in algorithm 3.2: signal duration, spreading factor, channel and power at the receiver location. This last parameter is computed by the LoraChannel by resorting to the PropagationLossModel, an object which computes the power loss based on the transmission power and the locations of transmitter and receiver.

It’s important to notice that LoraChannel is completely unaware of issues such as sensitivity and interference: its only task is computing the received power and delay at the location of every other device in the network, given the transmission power and position of the transmitter. In order to speed up the simulations featuring nodes that span a very large area, a power threshold can be introduced so that every transmission that would arrive under that threshold value won’t even be delivered to the PHY layer. In addition to this measure, a way to greatly reduce the computational time needed for the simulation of UL only networks is to check whether the PHY device the channel is delivering the message to is an ED or a GW, and only performing the necessary computations of the received power and schedule the receive event if the receiving PHY is a GW. In fact, it’s useless to perform interference computation at the end devices if all traffic is expected to flow in the UL direction, since an ED will never open a receive window.

3.3.5 LoraNetDevice

The LoraNetDevice class models a “LoRa network card”: this NetDevice can be attached to a node, whose applications can then use the card to send data to other LoRa devices. A LoraNetDevice is essentially used to hold together all the LoRa objects that need to be aggregated to a node: a LoraPhy and a LoraMac. One consideration must be made regarding the abstract class NetDevice, and its orientation towards IP communications.

It is stated in the LrWpanNetDevice API documentation that “[t]he ns3::NetDevice includes IP-specific API such as GetMulticast, Send and Receive methods, which do not map well the 802.15.4 MAC MCPS DataRequest primitive. So, the basic design is to provide, as much as makes sense, the class ns3::NetDevice API, but rely on the user accessing the LrWpanMac pointer to make 802.15.4-specific API calls”. Likewise, the LoRa module uses the NetDevice as an encapsulating class, and only leverages a generic Send version that is adapted to handle the underlying MAC layer. No support for concepts such as multicast and IPv6 addresses is provided.

3.4 Helpers and Tests

Aside from the fundamental building blocks described above, a set of “helper” classes was also implemented to make configuring a Lora network easier. Helpers are an NS3 design element, intended to assist script creators in setting up topologies and nodes that are fully configured to use the desired module.

Whenever an instance of the PeriodicSender application is created, its period must be set. In order to make the correct configuration of many of these classes easier, applications can be deployed on a set of nodes by using the PeriodicSenderHelper class. This helper decides a node’s reporting period according to the Orange (2016) and Palattella et al.’s (2016) specifications, so that the appropriate distribution of periods among nodes described above is respected.

Another set of helper procedures was then written in order to install and configure the Lora stack on a given set of nodes at the same time: the classes LoraHelper, LoraPhyHelper and LoraMacHelper were designed to work in synergy to create and deploy LoraNetDevice, LoraMac and LoraPhy objects on a wide set of nodes, making sure that layers are configured to communicate appropriately with one another and that the PHY layers are correctly connected to the LoraChannel instance they share.

A set of tests was also written, along with the module, in order to ensure the correctness of the software after each update and to comply with the NS3 guidelines. Elementary message delivery is tested to ensure that the PHY layer at an end device is able to receive a message from a device within range. Channel separation tests make sure that a device listening to channel *i* will not receive communications sent in channel *j* when $i \neq j$. Interference checks verify that a packet can be destroyed by an interferer with sufficiently high power. Furthermore, they also verify that communications on different channels do not interfere. Finally, gateway parallel decoding capacity tests make sure that a gateway can receive no more than 8 messages in parallel.

Procedure 3.4 Determine whether a packet was destroyed by interference or not

```
Input: packet = the packet we want to receive
Input: duration = the time taken to receive the packet
Input: interferers = list of registered interferers so far
Input: rxPowerdBm = minimum receive power of the packet
Input: sf = packet spreading factor
packetLost = false
for currentsf $\in$ 7,...,12, until packetLost = false
    cumulativeInterferenceEnergy = 0
    for interferer $\in$ interferers
        if interferer’s channel number = packet’s channel number OR interferer = packet
            Skip this event
        Remove the current interferer from the list if it is older than a threshold
        if interferer’s sf = currentSf
            overlap = overlap between the interferer and packet windows
            interferenceEnergy = overlap * interferer’s power
            cumulativeInterferenceEnergy += interferenceEnergy
```

```
        if cumulativeInterferenceEnergy = 0
            snir = snir computed according to Eq. (3.2)
            snirIsolation = isolation from Eq. (3.20), according to sf and currentSf
            if snir $\geq$ snirIsolation
                Continue with the loop
            else
                packetLost = true
return packetLost
```

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Performance Evaluation

This section analyses various metrics obtained in chapter 3. Firstly, some results on the network's throughput performance were evaluated. Then a network was set up with a realistically modelled environment and traffic model in order to evaluate the probability of losing a packet in a LoRaWAN. Some experiments were then performed to evaluate the performance of each single spreading factor on interference, and how packet success rate changes with spreading factor. Finally, some tests on coverage in a realistic setting were performed, in order to evaluate the gateway density that is required to cover a city with deep shadowing caused by buildings.

4.2 Throughput Performance

The first simulation campaign aims at evaluating the network throughput S as a function of the network offered traffic G. The scenario is characterized by a single central GW and N EDs that are uniformly distributed in a circle of radius r = 7500 m around it. This particular radius value was chosen because r is the maximum distance at which the gateway and an end device using SF = 12 are able to communicate above sensitivity only considering the propagation loss.

For the computation of throughput, we suppose that EDs $i = 1,\ldots,N$ generate every τ_i seconds a packet which occupies the channel for $t_{p,i}$ seconds in order to be transmitted. In fact, we are interested in testing the LoRaWAN access scheme in itself, regardless of the local regulations that may modify its performance. We compute the network offered traffic as described in *(PRINCIPLES OF COMMUNICATIONS NETWORKS AND SYSTEM*S, n.d.):

$$G = \sum_{i=1}^{N} \frac{t_{p,i}}{\tau_i} \,. \qquad (4.1)$$

The offered traffic is, in other words, a measure that expresses the fraction of time the channel is occupied by transmissions by end devices. $G < 1$ means that the channel is underutilized, since there are times at which no transmission is going through the channel. On the other hand, $G > 1$ means that, even with a perfect synchronization of the devices, some packets will necessarily try to use the channel at the same time, causing a collision. For a given value of G, throughput S is then obtained as:

$$S = G \cdot P_{\text{succ}} \, , \qquad (4.2)$$

where the probability of success of a given packet is approximated as the P_{succ} ratio between the number of successfully received packets and the total number of sent packets during a simulation. A network offering a traffic of 1 and featuring perfect synchronization between devices, so that collisions are avoided and no packets are lost, will yield a throughput of 1. Of course, a perfect synchronization between devices is impossible to achieve, so it is expected that $S < 1$. Especially for networks where devices access independently the channel with no coordination such as time or frequency slotting, as is the case of LoRaWAN, throughput is expected to follow the shape of the ALOHA medium access protocol.

As a first validation of the simulator described in Chapter 3 we expect that, under ideal channel conditions, the shape of the throughput curve for a varying offered traffic will be that of a typical ALOHA network. If we turn off the link measurement model, all end devices are configured to transmit with $SF = 7$ and all packets have the same ToA (provided the payload length is fixed) and are received with the same power at the central gateway. We also assume that overlapped packets always fatally collide and are therefore lost.

Using this setting, the expression for the traffic offered by a network of N devices can be expressed as:

$$G = \frac{N \cdot t_7}{\tau_i} \qquad (4.3)$$

where t_7 is the ToA for a packet using SF = 7 and a fixed payload length.

As expected, the performance result of this test, follows the typical ALOHA throughput trend *(PRINCIPLES OF COMMUNICATIONS NETWORKS AND SYSTEM*S, n.d.), for which an exact expression exists:

$$S = G \cdot e^{-2G} \qquad (4.4)$$

After this first validation, we can evaluate the impact of using a variety of SFs and a real wireless channel, by using the log-distance component of the proposed link measurement model: indeed, the presence of a real channel motivates the usage of all possible SFs to allow devices further away to communicate with the GW. The fact that multiple SFs are used also allows the network to leverage the quasi-orthogonality of transmissions using different SFs. We also studied the impact of $SF = 12$ transmissions on the performance of a LoRa network, especially on interference. In order to do so, a simulation would be created in which EDs configured to use SF 12 were not allowed to transmit, even though their traffic still counted towards the generated traffic total. This way, while some of the generated packets will always fail transmission and count as lost packets in the P_{succ} ratio, they will also generate no interference with other transmissions in the network. The computation results show that excluding end devices with the highest SF is beneficial when the system load is high, because the collisions with other end device transmissions are reduced, and thus the success probability increases up to $S_{gain} = 0{:}12$ with respect to the scenario in which SF = 12 devices are allowed to transmit. On the other hand, for low offered traffic values, when interference is not the limiting factor for throughput, the fact that some packets are not transmitted affects throughput in a negative way, causing a loss of up to $S_{loss} = 0.1$. This behaviour is in line with the mandate by the LoRa Alliance to exclude from public networks end nodes which only transmit at SF = 12 and refuse to change their SF: while these nodes will have higher probabilities of successfully delivering a packet given the better sensitivity of the GW to SF = 12, their effect on the network as a whole would be detrimental.

4.3 Comments

The results derived in this chapter highlight the qualities of the LoRa modulation. Firstly, it was shown that, by leveraging the quasi-orthogonality between different SFs, a network employing LoRa devices achieves a throughput that is higher than that of a standard ALOHA system, without adding any burden to the MAC scheme, like coordination and synchronization between devices or carrier sensing. This feature allows LoRa networks to scale well, while maintaining a low complexity that is suitable for IoT devices. Furthermore, it's thanks to the modulation and its high sensitivity values that LoRa networks can be deployed in an urban scenario with a reduced density with respect to proposed CIoT networks. Finally, SFs represent another feature of the network that the NS can tune to optimize the performance of the network, finding the right balance between the resistance to interference brought by SF diversity and the maximum coverage possible. The fact that the LoRaWAN standard does not define gateway cells means that EDs are not tied to a single gateway. Gateways can be placed freely, and without necessarily following a predefined deployment scheme: this is a key feature that allows crowdsourced solutions like The Things Network to grow so fast, encouraging users to deploy their own gateway and increase the coverage and performance of the whole network. This characteristic, however, leaves room for potential issues, tied to the fact that an ED in an area covered by a certain GW could, because of shadowing, find it easier to communicate with the NS via another cell's GW. This behaviour can be seen as an advantage, since it allows devices to find multiple GWs to which to deliver their packets, and it also avoids the burden of handover procedures. On the other hand, this mechanism may disadvantage the GWs that have better exposure. In fact, since the NS forwards DL messages through the "best" gateway (i.e., the gateway that received a UL message with the highest power), a GW placed in a favourable position to cover a wide area may be burdened with DL communications that are intended for EDs that are, in fact, closer to other GWs.

CHAPTER FIVE
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This work covered one of the most promising LPWAN technologies, LoRa, and introduced a framework that allows the evaluation of network level performance of a LoRaWAN system.

After an introduction, that briefly covered the requirements and challenges of the IoT paradigm, the LoRa modulation and the LoRaWAN standard were covered, along with the regulatory framework in which these technologies are meant to operate.

A review of the literature on LoRa systems and their performance with regards to range and ability to scale was done. After this introduction to the state of the art, various models of the different components of a LoRaWAN system were described, with particular attention to the modelling of interference and the usage of realistic traffic and propagation models.

After a brief introduction to the NS3 network simulator, the new LoRa extension module, developed as part of this thesis, was described in detail along with the standard NS3 classes that were used to implement some of the models.

Moreover, the LoRaWAN architecture was proved to scale well, mainly because an increase in the number of gateways enhances the coverage and reliability of the uplink.

Finally, a simulation involving a network featuring multiple gateways and a realistic traffic model resulted in a packet success rate above 95% for a gateway serving approximately 15,000 end devices, confirming some of the claims that were made by the companies supporting this new system.

5.2 Future Work

Simulation to be extended for the future work using NS3 described above.

REFERENCES:

Actility. (2013). ThingPark Wireless PHY and MAC layer specifications, 1–24.

Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2012). A review of smart homes - Past, present, and future. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *42*(6), 1190–1203. https://doi.org/10.1109/TSMCC.2012.2189204

Alhaj, A. S. A. (2015). Home Automation System Using Arduino, *13*(July), 1–5.

Ali, M., Zaidi, Z. A., Zia, Q., Zakaria, M., Ahsan, S. M., & Rameez, M. (2012). Intelligent load management scheme for a residential community in smart grids network using FEDRP. *WSEAS Transactions on Power Systems*, *7*(3), 104–113.

Amato, G., Caruso, A., Chessa, S., Masi, V., & Urpi, A. (n.d.). State of the Art and Future Directions in Wireless Sensor Network ' s Data Management Project : Management of Data in Wireless Sensor Networks (MaD-WiSe), 1–31.

Bədicə, C., Brezovan, M., & Bədicə, A. (2013). An overview of smart home environments: Architectures, technologies and applications. *CEUR Workshop Proceedings*, *1036*(i), 78–85.

Bregman, D., & Korman, A. (2009). A universal implementation model for the smart home. *International Journal of Smart Home*, *3*(3), 15–30.

Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, *23*(5), 60–67. https://doi.org/10.1109/MWC.2016.7721743

COMMUNICATIONS NETWORKS AND SYSTEMS PRINCIPLES OF COMMUNICATIONS. (n.d.).

Cook, D. J., & Das, S. K. (2007). How Smart are our Environments ? An Updated Look at the State of the Art The Role of Physical Components in Smart Environ ments. *Journal of Pervasive and Mobile Computing*, *3*(2), 53–73. https://doi.org/10.1016/j.pmcj.2006.12.001

De Silva, L. C., Morikawa, C., & Petra, I. M. (2012). State of the art of smart homes. *Engineering Applications of Artificial Intelligence*, *25*(7), 1313–1321. https://doi.org/10.1016/j.engappai.2012.05.002

Ding, D., Cooper, R. A., Pasquina, P. F., & Fici-Pasquina, L. (2011). Sensor technology for smart homes. *Maturitas*, *69*(2), 131–136. https://doi.org/10.1016/j.maturitas.2011.03.016

Engineering, T. (n.d.). Università degli Studi di Padova Network level performances of a LoRa system.

Ericsson, AB. (2016). Cellular networks for Massive IoT – enabling low power wide area applications. *Ericsson*, (January), 1–13. Retrieved from https://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf

Erol-Kantarci, M., & Mouftah, H. T. (2011). Wireless Sensor Networks for Cost-Efficient Residential Energy Management in the Smart Grid. *IEEE Transactions on Smart Grid*, *2*(2), 314–325. https://doi.org/10.1109/TSG.2011.2114678

European Telecommunications Standards Institute. (2002). Electromagnetic compatibility and Radio spectrum Matters (ERM); Electromagnetic Compatibility (EMC) Standard for Radio Equipment and Services; Part 1: Common Technical Requirements. *Intellectual Property*, *1*, 1–35.

Fsk, G., Lora, D. D. R., & Lora, G. F. S. K. (2017). WIRELESS & SENSING PRODUCTS Datasheet, (May), 1–40.

Goursaud, C., & Gorce, J. M. (2015). Dedicated networks for IoT: PHY / MAC state of the art and challenges. *EAI Endorsed Transactions on Internet of Things*, *1*(1), 150597. https://doi.org/10.4108/eai.26-10-2015.150597

Homes, S., & Tedblad, R. (2015). Reliable Wireless Sensor Networks in Smart Home, (September).

Hu, Q., & Li, F. (2013). Hardware design of smart home energy management system with dynamic price response. *IEEE Transactions on Smart Grid*, *4*(4), 1878–1887. https://doi.org/10.1109/TSG.2013.2258181

Hussain, I., Mohsin, S., Basit, A., Khan, Z. A., Qasim, U., & Javaid, N. (2015). A Review on Demand Response: Pricing, Optimization, and Appliance Scheduling. *Procedia Computer Science*, *52*(Seit), 843–850. https://doi.org/10.1016/j.procs.2015.05.141

Knight, M. (2016). Decoding LoRa : Realizing a Modern LPWAN with SDR. *Proceedings of the 6th GNU Radio Conference*,.

Knight, M. (2016). Reversing LoRa. *Bastille Networks*, 128. Retrieved from http://www.jailbreaksecuritysummit.com/s/Reversing-Lora-Knight.pdf

L’Ecuyer, P., Simard, R., Chen, E. J., & Kelton, W. D. (2002). An Object-Oriented Random-Number Package with Many Long Streams and Substreams. *Operations Research*, *50*(6), 1073–1075. https://doi.org/10.1287/opre.50.6.1073.358

Lê, Q., Nguyen, H. B., & Barnett, T. (2012). Smart Homes for Older People: Positive Aging in a Digital World. *Future Internet*, *4*(December), 607–617. https://doi.org/10.3390/fi4020607

Madakam, S., & Ramaswamy, R. (2015). Smart Homes (Conceptual Views). *Proceedings - 2014 2nd International Symposium on Computational and Business Intelligence, ISCBI 2014*, (December 2014), 63–66. https://doi.org/10.1109/ISCBI.2014.21

Mendes, T., Godina, R., Rodrigues, E., Matias, J., & Catalão, J. (2015). *Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources*. *Energies* (Vol. 8). https://doi.org/10.3390/en8077279

Nagar, R., Nadu, T., Nagar, R., & Nadu, T. (2016). Smart Home Control Systems Based on Wireless Sensor Networks Using ZIGBEE and, *5*(3), 108–117.

ns-3 Tutorial. (2017).

Orange. (2016). LoRa Device Developer Guide, 42. Retrieved from https://partner.orange.com/wp-content/uploads/2016/04/LoRa-Device-Developer-Guide-Orange.pdf

Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., & Ladid, L. (2016). Internet of Things in the 5G Era : Enabling Technologies and Business Models. *IEEE Journal on Selected Areas in Communications*, *34*(3), 1–9. https://doi.org/10.1109/JSAC.2016.2525418

Saad al-sumaiti, A., Ahmed, M. H., & Salama, M. M. A. (2014). Smart Home Activities: A Literature Review. *Electric Power Components and Systems*, *42*(3–4), 294–305. https://doi.org/10.1080/15325008.2013.832439

Satpathy, L. (2006). Smart housing: Technology to aid aging in place. New opportunities and challenges. *ProQuest Dissertations and Theses*, *1441072*(August), 192-192 . Retrieved from http://ezproxy.net.ucf.edu/login?url=http://search.proquest.com/docview/305315332?accountid=10003%5Cnhttp://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+theses&sid=ProQ:ProQuest+Dissertations+&+T

Semtech. (2015). LoRa Modulation Basics, (May), 1–26. Retrieved from http://www.semtech.com/images/datasheet/an1200.22.pdf

Singh, M. P., & Singh, D. K. (2010). Routing Protocols in Wireless Sensor Networks –, *1*(2), 63–83.

Torunski, E., Othman, R., Orozco, M., & El Saddik, A. (2012). A review of smart environments for energy savings. *Procedia Computer Science*, *10*, 205–214. https://doi.org/10.1016/j.procs.2012.06.029

Ullo, S., Vaccaro, A., & Velotto, G. (2010). The role of pervasive and cooperative sensor networks in smart grids communication. *Proceedings of the Mediterranean Electrotechnical Conference - MELECON*, 443–447. https://doi.org/10.1109/MELCON.2010.5476236

Whitening, S. (2013). Implementing Data Whitening and CRC Calculation in Software on SX12xx Devices Table of Contents Index of Figures, (October), 1–14.