UNIVERSITÀ DEGLI STUDI DI PADOVA
DEPARTMENT OF INFORMATION ENGINEERING

MASTER THESIS IN
TELECOMMUNICATION ENGINEERING

Reliable LoRaWAN links: performance analysis

Supervisor
Prof. Andrea ZANELLA

Co-supervisior
Davide MAGRIN

Author
Martina CAPUZZO

Abstract

Nowadays the Internet of Things (IoT) paradigm is attracting interest in the scientific and commercial fields, and many technologies have been proposed as solutions to the increasing demand of connected devices and suitable network infrastructure. Among the proposed standards, Low Power Wide Area Networks (LP-WANs) offer the competitive advantages of long range communication and low power requirements, exploiting license-free frequency bands. This thesis focuses on the emerging LoRa solution and on its network performance when reliable communication is employed.

After a description of the LoRa technology and LoRaWAN standard, the discussion presents the implementation of LoRaWAN reliable communication in the network simulator ns-3 and proposes a mathematical model for the performance of LoRaWAN. Finally, the performance of a LoRa system is evaluated by means of simulations and compared to the theoretical results provided by the mathematical model. The good matching between simulation and analytical results confirms the validity of the proposed model.

Sommario

Al giorno d'oggi il paradigma dell' Internet of Things (IoT) attrae interesse sia in ambito scientifico-accademico che in quello commerciale, e molte tecnologie sono state proposte come soluzione per la crescente domanda di dispositivi connessi e di adeguate infrastrutture di rete. Tra gli standard proposti, le Low Power Wide Area Networks (LP-WANs) offrono i competitivi vantaggi di una comunicazione a lungo raggio e basso fabbisogno energetico, sfruttando bande di frequenza libere. Questa tesi si concentrerà sulla soluzione emergente LoRa e sulle sue prestazioni di rete quando è utilizzata una comunicazione affidabile.

Dopo una descrizione della tecnologia LoRa e dello standard LoRaWAn, la trattazione presenterà l'implementazione di una rete LoRaWAN con connessione affidabile nel simulatore di rete ns-3 e sarà proposto un modello matematico per LoRaWAN. Infine, si valuteranno le prestazioni di un sistema LoRa attraverso varie simulazioni, e le si confronteranno ai risultati teorici ottenuti dal modello matematico. per verificare la validità di quest'ultimo. La coerenza di simulazioni e risultati del modello analitico conferma la validità del modello proposto.

Contents

List of Figures

List of Tables

Acronyms

3GPP Third Generation Partnership Project.

ACK Acknowledgment.

ADR Adaptive Data Rate.

BER Bit Error Rate.

BLE Bluetooth Low Energy.

CAGR Compound Annual Growth Rate.

CR Code Rate.

CRC Cyclic Redundancy Check.

CSS Chirp Spread Spectrum.

DSSS Direct-Sequence Spread Spectrum.

EC-GSM Extended Coverage GSM.

ED End-device.

eDRX Extended Discontinuous Reception.

EIRP Effective Isotropic Radiated Power.

eMTC Enhanced Machine-Type Communication.

ERP Effective Radiated Power.

ETSI European Telecommunications Standard Institute.

FDMA Frequency Division Multiple Access.

FEC Forward Error Correction.

FFT Fast Fourier Transform.

FHDR Frame Header.

GFSK Gaussian Frequency Shift Keying.

GW Gateway.

ISM Industrial Scientific Medical.

ITU International Telecommunication Union.

LBT Listen-Before-Talk.

LP-WAN Low Power Wide Area Network.

LR-WPAN Low-Rate Wireless Personal Area Network.

MAC Medium Access Control.

MFSK Multiple Frequency Shift Keying.

MHDR MAC Header.

MIC Message Integrity Code.

MType Message Type.

NB-IoT Narrow Band Internet of Things.

NS Network Server.

PAN Personal Area Network.

PDR Packet Delivery Ratio.

PER Packet Error Rate.

PHY Physical Layer.

QoS Quality of Service.

RF Radio Frequency.

RPMA Random Phase Multiple Access.

RX1 first receive window.

RX2 second receive window.

SDN Software Defined Radio.

SF Spreading Factor.

SINR Signal plus Interference to Noise Ratio.

SIR Signal to Interference Ratio.

SNR Signal to Noise Ratio.

TDMA Time Division Multiple Access.

UNB Ultra Narrow Band.

WLAN Wireless Local Area Network.

WPAN Wireless Personal Area Network.

Chapter 1

Introduction

Today, the Internet has become an important part of everyday life. Internet connection is nowadays widely available, its cost is decreasing, and more and more devices are created with Wi-Fi capabilities and built-in sensors. The next stage, which is rapidly approaching, is to extend the Internet connectivity to virtually any object, in order to enable their remote control and actioning.

The Internet of Things has been defined as the technologies that will allow *people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service*. A more formal definition has been given in [11]:

> The Internet of Things (IoT) is a network of networks where, typically, a massive number of objects/things/sensors/devices are connected through communications and information infrastructure to provide value-added services.

As written in [12], "anything that can be connected, will be connected ".

The authors of [13] affirmed that in 2017 8.4 billion of connected devices would have been in use, with a growth of 31 % from the previous year, and the prediction was that they will reach 20.4 billion by 2020. This has its effects also in the market: based on data reported in [14], the global IoT market will grow form 157 billion dollars in 2016 to 457 billion dollars in 2020, attaining a Compound Annual Growth Rate (CAGR) of 28.5%, according to [15]. Other sources, still cited in [14], foresee a CAGR of 23%, pointing that most of the applications will be in smart cities and industrial sector. Although other sources in literature and n the web report different numerical results, still there is a general agreement on the significant growth of the sector.

While the IoT paradigm is somehow universal, many particular solutions can be applied in different use cases. Figure 1.1 draws a picture of the applications that could be used in a smart city as, for example, smart parking, smart lighting, air pollution control. Recently, some city-wide services for objects tracking (or even kid and pet monitoring) have been commercialized by some telecom operators. Other

applications are found in the health domain, where wearable devices (some of them already in use) could monitor life parameters, such as heartbeat, quality of sleep or physical activity. In the enterprise fields applications can be the monitoring the status and usage of the machines, of stocks, of energy production or waste and controlling the location of the merchandise or of the fleet of vehicles.

It can be observed that even if there is a variety of applications, most of them share the monitoring feature. In fact, in most of the cases, the *things* that will be connected are intended as sensors and/or other devices equipped with usually not very powerful processing and storage units that communicate through an interface connected to a network. Then, all this information is analyzed by a central unit (server) that makes decision based on them (for example, turning on/off the lights) or communicates data to the user.

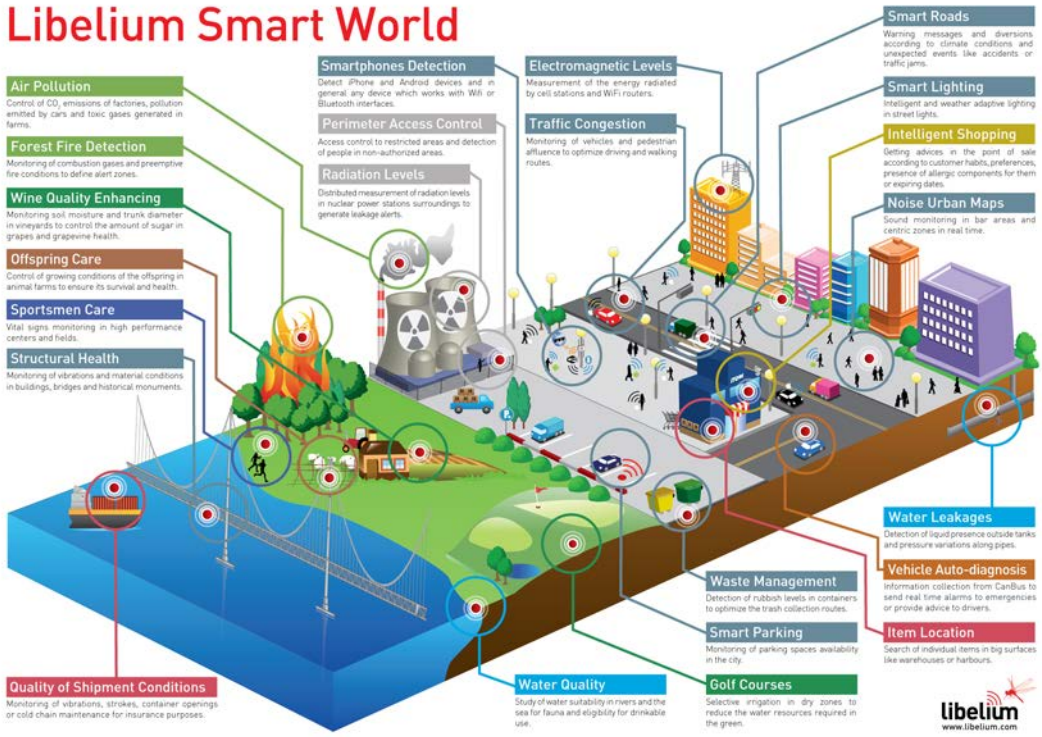


Figure 1.1: Examples of IoT applications.

The features of IoT described till now translate into technical requirements that must be fulfilled by technologies that envisage IoT as target.

Scalability The networks implementing IoT paradigm will handle thousands of nodes generating traffic in a periodic way or occasionally, when an event happens. However, also in this last case, there could be events that activate all the network sensors simultaneously, and this peak of traffic must not result in a network failure. Therefore all the parameters and aspects of the network must be set or chosen so as to support a large number of devices.

Low device cost Since the network will be wide the cost of a single device should be low (3 - 5 $), making the complete implementation more affordable, both

for a company and for a single user. At the same time, since most typical devices will be sensors, they will likely have low requirements in terms of computational capabilities, helping reducing the costs.

Computing power If on one side requiring just little computational capabilities allows to employ basic CPUs maintaining the device cheap, on the other side this requires simple instruction sets, communication protocols and modulation schemes for data transmission.

Long battery life It is expected that many IoT devices will run on batteries. Due to the dimension of the network, an adequate maintenance like that for common devices would cost too much in terms of time and money. For this reason a long battery life (5 - 10 years) should be achieved in order to overcome this problem. Most of the time it is sufficient to transmit data to the central server infrequently: this allows devices to stay longer in sleeping mode, suspending the connection and thus saving energy resources.

Good coverage The utility of reaching a long communication range can be seen easily in tracking application, where a unique server could be able to follow the displacement of the device in a range of some kilometers (usually 5 to 30). This feature is also important when sensors are not subject to mobility, to minimize the required network infrastructure. Furthermore, indoor environments need a deep coverage to enable, for examples, smart home and smart city applications: in fact, many and different obstacles are present especially in urban scenarios, causing shadowing and losses that affect the communication.

Low bit rate Most of the applications of IoT are monitoring and sensing. In these cases messages are short and a periodic data reporting is sufficient.

Different solutions have been proposed in the last years to fulfill these requirements, and they present different features depending on the target application.

Since the IoT is a growing sector that is acquiring more and more importance, the scientific and economic communities are dedicating attention to it. In particular many technical aspects, such as the performance of the technologies, their cost and feasibility should be analyzed before putting them in place.

Among the different emerging solutions, a new class of technologies is the so called LP-WAN, specifically thought for IoT and Machine-to-Machine connections (M2M). In this category, better described in the following chapters, one promising technology is LoRaWAN. Its main characteristic is that it can be configured in an easy way, adapting its performance to the different situations. This, together with the fact that the network protocol LoRaWAN is open-standard and its mature deployment state even if it appeared in the market quite recently (LoRa modulation was patented in 2012), has made LoRaWAN very interesting, and multiple performance analysis in different use cases have been conducted.

Till now, most of the studies on LoRaWAN dealt with the uplink transmission without acknowledgment. As reliable transmissions were not analyzed in detail, we

focus our study on uplink LoRaWAN communication requiring acknowledgment, and propose a mathematical model for LoRaWAN channel access. The investigation is supported by simulations with ns-3 network simulator, conducted in different scenarios.

The rest of this thesis is organized as follows.

- In Chapter 2 we give an overview of the main existing IoT technologies, focusing then on LoRa modulation and LoRaWAN specification for the MAC layer. We will point out the main features that make LoRa attractive.
- Chapter 3 will describe some of the works that already dealt with this technology, highlighting strengths and weaknesses.
- Chapter 4 will describe the assumptions and choices done to model LoRaWAN devices (End Devices, Gateways, Network Server), channel and environment for the implementation in the simulator. Furthermore, we propose the model to analyze LoRaWAN performance.
- Chapter 5 gives a brief introduction of the ns-3 network simulator and explains the structure of the code implementing LoRaWAN and the network used for simulations.
- Chapter 6 will present and discuss the results achieved with model and simulations.
- Chapter 7 will draw the conclusions of this work and suggest possible future developments.

Chapter 2

IoT technologies

As introduced in Chapter 1, many different solutions are competing for the IoT market. This section aims to give an overview of the most important. The technologies are classified according to their coverage and briefly described. Then, the chapter focuses on the LoRa modulation and LoRaWAN standard, describing the elements that will be modeled and implemented in the simulator.

2.1 Main approaches

IoT technologies are often classified in three categories based on their characteristics in terms of coverage range, throughput and cost. In Figure 2.1, for example, the classification is based on the range and power consumption. In the following, we will not treat Wireless Local Area Networks (WLANs), since they are not suitable for IoT because of their short range and high energy consumption.

Up to now it is not clear if one solution will overcome the others, since they all have strengths and weaknesses and could even be applied at the same time for different targets. However, it is likely that consumers will be attracted by the first technologies entering the market, proposing satisfying services at low cost.

2.1.1 Low-Rate Wireless Personal Area Networks (LR-WPANs)

This category (indicated in Figure 2.1 under the name of Short-range Wireless) includes technologies characterized by low bit rate, low power consumption and short coverage range, reaching at maximum a few hundred meters. This range can be extended using a dense deployments of gateways and devices connected in a multihop mesh network. For this reason such a deployment becomes economically unfeasible and technically complex in very large scenarios such as cities or wide open-air areas. We describe now the main solutions.

Bluetooth Low Energy (BLE) BLE was defined in 2010 by the Bluetooth Special Interest Group as a single-hop solution suitable for use cases as healthcare, smart energy, consumer electronics and security. Differently from other technologies of this category, BLE is expected to have high deployment in devices

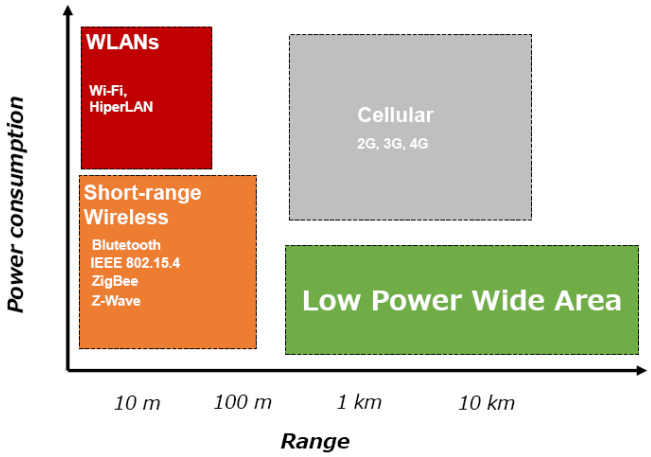


Figure 2.1: LP-WAN vs. legacy wireless technologies [1].

such as smartphones, similarly to the already existing Bluetooh [16]. It operates in the 2.4 GHz Industrial Scientific Medical (ISM) band, defining 40 channels with different functions. The physical data rate is 1 Mbps and the coverage range is typically over various tens of meters. Similarly to Bluetooth, BLE specification defines two device roles: the master and the slave. In the first specification, the slave could be connected only to one master, which handled multiple connections through a Time Division Multiple Access (TDMA) scheme. This has been changed in 2013, with the specification 4.1 that allows slaves to be simultaneously connected to more than one master. To save energy, slaves can turn their radios off for some time intervals. However, the star topology does not allow for some advantages such as path diversity, important to overcome radio propagation impairments and node failures in wireless systems. For this reason, the implementation of a mesh network topology has been suggested by standard development organizations, academic community and industries [17].

IEEE 802.5.4 and ZigBee In 1999, IEEE established the 802.15 working group to develop specific standards for Wireless Personal Area Networks (WPANs). In particular, target group four (802.15.4) was given the responsibility of developing standards for Physical Layer (PHY) and Medium Access Control (MAC) layers enabling the transmission of small flows of data consuming a low amount of energy. By using the standard developed by 802.15.4, ZigBee proposes a complete standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), adding the specifications for network and application layers [18]. This technology works in the ISM 2.4 GHz band and defines 16 channels, which can be selected by the operating device through a frequency hopping mechanism. Three type of devices are defined. The Personal Area Networks (PANs) coordinator and the coordinator are defined as Full Function Devices as they implement all the functionalities of the IEEE 802.15.4 protocol. They transmit beacon frames to provide global and local synchronization

respectively. Simple nodes are connected to the coordinators and have no coordinating functionalities. ZigBee proposes three types of network topologies: star, tree and mesh. While star and tree topology present the disadvantage that, if a link fails, the node can no more communicate, mesh topology is the most reliable, since it provides multiple paths from the simple node to the coordinator. The specification claims a maximum transmission range of a hundred meters, a throughput between 20 and 250 kbps and a capacity of 64000 nodes for each PAN coordinator. The main applications of Zigbee are found in home automation and helthcare [19].

Z-Wave Z-Wave is a proprietary protocol designed by Zensys mainly for home automation applications. It aims to communicate short messages in a reliable manner from a central control unit to one or more nodes in the network using half duplex communication. It uses ISM band around 900 MHz to limit power consumption and interference with other technologies. Its range goes from 30 meters indoor to about one hundred meters in outdoor environments. The protocol defines four layers: the MAC layer, that controls the Radio Frequency (RF) media, the Transfer layer, that controls the transmitting and receiving of frames, the Routing layer and the Application layer. Two basic kinds of nodes are specified: controlling devices that initiate control commands and slave nodes that receive commands, execute them and potentially answer to them. If required, slave nodes can also forward commands to other nodes, extending the communication range of the controller. Controllers also maintain routing tables and one of them (primary controller) has the possibility of including/excluding nodes in the network [20].

2.1.2 Cellular IoT

Cellular technologies for IoT leverage the existing cellular networks to offer new services. One of their main characteristic is that, differently from the other categories, they use the licensed spectrum. For this reason they are very attractive for telecom operators: the network infrastructure is, for most, already installed and usually, only a software upgrade is required. In the following, we give an overview of the three solutions proposed by 3GPP in Release 13 [21].
Providing IoT connectivity is also one of the targets of 5G networks, currently under study.

EC-GSM Extended Coverage GSM (EC-GSM) has been proposed to improve already existing GSM technology. Using repetitions and signal combining techniques, a coverage improvement of 20 dB is reached. EC-GSM adds new control and data channels to legacy GSM and has the possibility of multiplexing new EC-GSM devices with legacy EDGE and GPRS [22]. Extended Discontinuous Reception (eDRX) is used to improve power efficiency and battery life. The achievable rate can vary from 350 bps to 240 kbps depending on the coverage and modulation used [21].

eMTC Enhanced Machine-Type Communication (eMTC), also called LTE-M aims to enhance LTE technology for machine-type communications. By reducing

bandwidth and maximum transmit power, eMTC enables new power-saving functionality, lower device complexity and cost. The coverage is extended and a low deployment cost for network operator is possible: control and data are multiplexed in the frequency domain and therefore it is possible to schedule IoT devices within any legacy LTE system, sharing the carrier capacity, antenna, radio and hardware. A bandwidth of 1.08 MHz is used and frequency hopping technique is adopted. The achieved rate is 1 Mbps both for uplink and downlink communication.

NB-IoT Similarly to the previous technology, Narrow Band Internet of Things (NB-IoT) reuses the already existing LTE infrastructure. The main difference is the narrow bandwidth used: 200 kHz [22]. The number of devices supported by each cell is about 50000 and the coverage range has a sevenfold increase with respect to LTE Release 12. NB-IoT supports three modes of operation according to the position of its carrier with respect to the LTE spectrum: stand-alone, guard band and in-band (we refer to [21, 22] for more details). Power saving mode described in 3GPP Release 12 and eDRX are used to increase battery life. The bit rate is about 250 kbps both in uplink and downlink. Also in this case, the low costs are achieved by lowering the complexity of the devices and by the fact that a software upgrade is sufficient to deploy it in LTE networks.

2.1.3 Low Power Wide Area Networks (LP-WANs)

The Low Power Wide Area Networks are halfway between LR-WPANs and cellular networs. In fact, they have a larger coverage than LR-WPANs, but lower cost and less energy consumption than cellular technologies. For these reasons they are spreading in the market and have already been adopted in different parts of the world. They are characterized by a star topology where peripheral nodes are directly connected to a concentrator, acting as gateway towards the IP network. Robust modulation allows the use of these technologies also in challenging environments, where cellular communication may fail [23].

Most of the LP-WAN technologies use sub-GHz ISM band, that is unlicensed and less crowded then 2.4 GHz band used by Wi-Fi and Bluetooth. We present now an overview of three LP-WAN solutions that compete with LoRa, which will be described in the following section.

Sigfox Sigfox has been the first LP-WAN technology proposed in the IoT market by the homonymous French start-up founded in 2009. The architecture consists of transmitting devices, gateways and Sigfox back-end. The coverage is deployed in each country by a Sigfox Network Operator that owns the whole infrastructure and requires a yearly fee for each device to provide access to the network. By using Ultra Narrow Band (UNB) communications, Sigfox can transmit a signal occupying a channel of only 100 Hz, thus benefiting from flat fading and very low noise contribution. Therefore, it is possible to have a simpler receiver and to successfully demodulate signals received with extremely low power. The channel access is random both in time and frequency. The randomness in the frequency domain overcomes the imprecision due to electrical

components deterioration and oscillator jitter, and gateways are configured to continuously scan the spectrum, listening at every channel. Since no acknowledgment is used, the reliability of communication is increased by making the device transmit three times the same message, by using each time a randomly selected channel. In this way, the transmission benefits from both time and frequency diversity. Since Sigfox is also subject to regional duty cycle limitations, each device can send a maximum of 140 messages per day in uplink and receive no more then 4 messages per day in downlink. The data rate is 100 bps in uplink and 600 bps in downlink [1]; the claimed achieved range is 30-50 km in rural areas and 3-10 km in urban environments.

Weightless Weightless is a set of three standards proposed by the British company Neul, acquired by Huawei in 2014. Each standard targets different use-cases, but each of them complies to IoT targets of low-power, low-cost and large coverage.
Wightless-N is based on narrow band technology with a star architecture. Transmissions are performed in the 868 MHz ISM band and there is only support for one-way communication that achieves a data rate of 30 - 100 kbps. The coverage range is 5 km also in urban environments, while the device cost is maintained very low [24]. *Wightless-P* improves the previous standard by adding downlink communication, that permits higher reliability by the use of acknowledgments. The channel band is 12.5 kHz and the access is performed with Frequency Division Multiple Access (FDMA) and TDMA schemes. The range is reduced to 2 km in urban environment. *Weightless-W* takes advantage of the available TV white spaces, so it uses the frequency at 470 and 790 MHz. It also enables coverage and data rate adaptation to better fit the actual requirements. Time Division Duplexing is used to provide uplink and downlink pairing, as spectrum is not guaranteed in the TV white space [24]. The range varies from 5 km (indoor) and 10 km (outdoor) and the rate from 1 kbps to 1 Mbps.

Ingenu Ingenu technology was proposed by the American company On-Ramp Wireless. The solution is based on Random Phase Multiple Access (RPMA), transmitting in the 2.4 GHz ISM band, with a typical channel bandwidth of 1 MHz. Data are first encoded, interleaved, and spread by a Golden Code, then the signal is randomly delayed before transmission. The spreading provides a processing gain and makes it possible to adapt the data rate to the propagation conditions. Uplink and downlink transmissions are performed in an half-duplex way and their rate varies from 60 bps to 30 kbps [24]. The communication range has been estimated to be up to 10 km in urban environments.

LoRa The previously described Sigfox technology is one of the most prominent solution in the IoT market. However, it has the limits of a low bit rate, maximum number of messages and is presented with a business model where Sigfox owns the network. Conversely, LoRaWAN technology is considered more flexible and only the PHY modulation is patented. Therefore, each user or company can buy LoRa devices and build its own network. However, from a technical perspective, what mostly differentiates LoRa from Sigfox is the fact that

the first exploits a new spread spectrum design at PHY layer that enables it to adapt range and data rate to the different scenarios. LoRa solution is in reality composed by LoRa modulation at PHY layer and LoRaWAN specification at MAC layer, that is not mandatory, but is the standard suggested by LoRa Alliance to set up the network and exploit the properties of the modulation in the best possible way. The covered range varies from 2 to 5 km in urban areas and up to 15 to 30 km in open spaces, while the range of the raw data rate is between 250 bps and about 5.5 kbps.
Since the objective of this thesis is to simulate and analyze LoRa's performance, we now focus on this technology, exploring LoRa modulation first and LoRaWAN standard next.

2.2 LoRa modulation

LoRa (**Lo**ng **Ra**nge) is a proprietary physical layer modulation used to create long-range wireless links, developed by Semtech and based on the Chirp Spread Spectrum (CSS) modulation technique. Being covered by a patent, most of the information is found in semi-official documents published by Semtech, like [5, 9, 25]. In the last year interest about how the modulation works has risen. Matthew Knight and Balint Seeber succeeded to reverse engineer and implement LoRa via Software Defined Radio (SDN) [4, 26, 27]. In the following discussion we will consider Semtech's official information, integrated with Knight and Seeber's findings.

2.2.1 LoRa's Chirp Spread Spectrum

The principle of CSS is to occupy with the transmission a bandwidth much larger than the one actually needed for the considered data rate. As explained in [24], CSS is a subcategory of Direct-Sequence Spread Spectrum (DSSS) and the receiver can benefit from controlled frequency diversity to recover weak signals and, thus, achieve a higher sensitivity. This way, the covered range is increased at the cost of a lower data rate, according to the LP-WAN IoT paradigm. In CSS data are spread using *chirps*, sinusoidal signals whose frequency linearly increases in time, spanning all the available bandwidth. For example, if we assume that the frequency band that can be used for the transmission is $B = [f_0, f_1]$, a chirp can start at frequency $f_s \in [f_0, f_1]$, increase its frequency linearly till f_1 and then wrap around from f_1 to f_0.

In particular, LoRa symbols are obtained as different circular shifts of the basic upchirp signal. These temporal shifts are slotted into multiples of time $T_{chip} = 1/B$, called *chip* where $B = f_1 - f_0$, and they characterize each symbol. The modulating signal for a generic n-th LoRa symbol can be expressed by the following equation

$$f(t) = \begin{cases} f_1 - n \cdot k \cdot T_{chip} + k \cdot t, & \text{for } 0 \leq t \leq n \cdot T_{chip} \\ f_0 + k \cdot t, & \text{for } n \cdot T_{chip} < t \leq T_s \end{cases} \tag{2.1}$$

where T_s is the LoRa symbol time and $k = (f_1 - f_0)/T$ is the slope of the frequency variations [2]. For the sake of clarity, Figure 2.2 depicts the basic upchirp and three modulated signals.

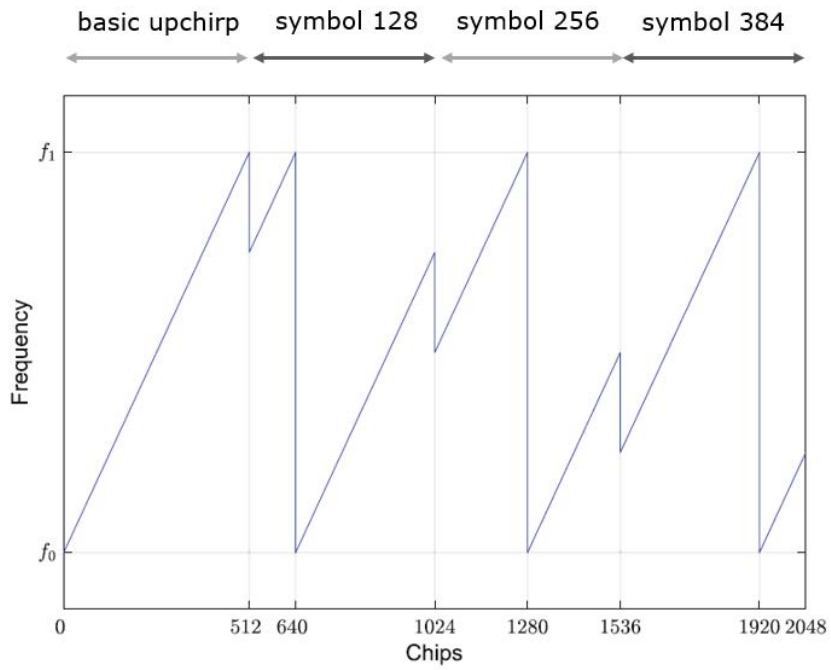


Figure 2.2: Modulating signal with SF = 9 for one basic upchirp and three symbols: 128, 256, 384 [2].

2.2.2 Configurable parameters

The total number of symbols (each one coding SF information bits) is $M = 2^{SF}$, where SF is a tunable parameter called spreading factor and goes from 7 to 12. The symbol time can be obtained as

$$T_s = 2^{SF} \cdot T_{chip} = \frac{2^{SF}}{B}, \tag{2.2}$$

and it can be noted that, for an increase of 1 in the SF, the symbol time doubles. When the symbol duration increases, more energy is put into each symbol, and this makes the signal more robust to noise and interference, reaching longer distances. However, SF also influences the data rate: for the raw data rate R_b at PHY layer it holds

$$R_b = SF \cdot \frac{B}{2^{SF}}, \tag{2.3}$$

and therefore the data rate is lower when a higher SF is used. It can also be noted that each time the SF is increased by one, the number of possible symbols doubles, and the occurrence of symbol errors becomes more likely. Thus, especially with low bit rates it is important to achieve synchronization between transmitter and receiver. Moreover, transmission lasts longer for higher SFs (equation 2.2) and then it is more subject to interference and collisions.

A second parameter that can be set in LoRa is the bandwidth (125, 250, 500 kHz). In general, larger bandwidth translates to a data rate increase and a receiver sensitivity deterioration and is used together with the SF to adapt to channel conditions. The availability of different bandwidths depends on the region the network is operating in, since regional regulations apply different constraints (see section 2.4).

SF	LoRa Demodulator SNR (dB)
7	- 7.5
8	- 10.0
9	- 12.5
10	- 15.0
11	- 17.5
12	-20.0

Table 2.1: SNR values at LoRa demodulator for different SFs [3].

Given these two parameters, we can analyze their influence on the received signal. In the datasheet [3] we find the values of Table 2.1 showing that a higher SF leads to a better SNR at the receiver.

The third configurable parameter is the carrier frequency. LoRa uses sub-GHz ISM band, but the center frequency and channel division depend on regional regulations. The carrier frequency in use can be set by users and can be changed in order to exchange more data provided that the duty cycle limitations are always respected [28].

LoRa provides support for Forward Error Correction (FEC) with configurable Code Rates (CRs). The code rate is defined as $CR = n/k$ where n is the number of information bits in the word and k is the total length of the word. Having $k > n$ makes it possible to add redundant bits to recover and correct errors. In LoRa the code rate is given by $CR = 4/(4+r)$ where $r \in 1,2,3,4$. Therefore, the set of possible code rates is $\{4/5, 4/6, 4/7, 4/8\}$. Reducing the CR provides more protection against error bursts, but also increases the packet transmission time.

2.2.3 LoRa PHY frame

Figure 2.3 illustrates the organization of the physical frame.

- The preamble has a synchronization function and defines the packet modulation scheme, since it is modulated at the same SF as the rest of the packet. It starts with a sequence of constant upchirp symbols, whose length can be variable, followed by two upchirp symbols encoding the sync word, that can also be used to distinguish different LoRa networks [29]. By default, the preamble is 12 symbols long, but in [3] it is suggested to configure identical preamble lengths on receiver and transmitter or, if the value is ignored, to program the receiver with the maximum preamble size.

- The header can be implicit or explicit, depending on the chosen mode of operation. In the second case, it contains the payload length in bytes, the FEC code rate, and signals if the optional Cyclic Redundancy Check (CRC) field is present at the end of the payload. By default, the header is always encoded with CR equal to $4/8$, the highest one. In case the receiver already knows all

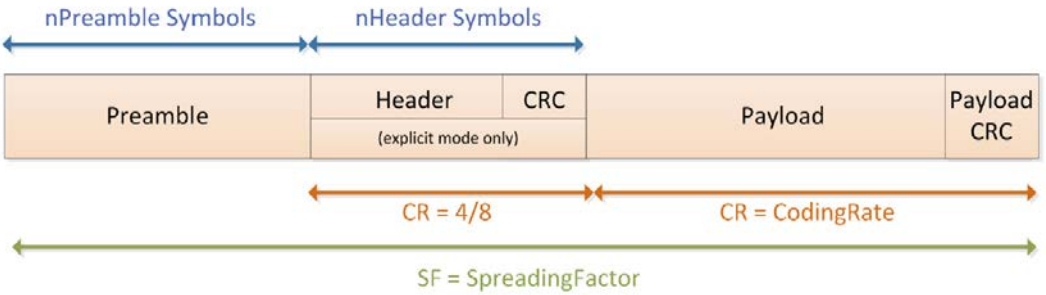


Figure 2.3: LoRa PHY packet structure [3].

the parameters, the header can be omitted (implicit header mechanism). The header also has its own CRC to allow the receiver discarding invalid headers.

- The packet payload has variable size and contains the actual data coded at the code rate specified in the header. An optional CRC may be appended.

For downlink frames, transmitted from the central server to the end node, the structure of the PHY frame is similar: all the fields are identical, except for the CRC field, which is not present. This is done to keep messages as short as possible, with minimum impact on any duty-cycle limitations of the ISM bands used [5] (see section 2.4 for further details).

It is possible to compute the packet time-on-air with the formula given in [3]:

$$T_{packet} = T_{preamble} + T_{payload} \tag{2.4}$$

$T_{preamble}$ is given by

$$T_{preamble} = (n_{preamble} + 4.25) \cdot T_s \tag{2.5}$$

where $n_{preamble}$ is the programmable number of symbols that constitute the preamble.

The payload duration depends on the header mode that is enabled, and is computed with

$$T_{payload} = n_{payload} \cdot T_s \tag{2.6}$$

with the number of payload symbols $n_{payload}$ given by the following formula:

$$n_{payload} = 8 + max\left(\left\lceil \frac{(8PL - 4SF + 16CRC - 20IH)}{4(SF - 2DE)} \right\rceil (CR + 4), 0\right) \tag{2.7}$$

Different parameters are involved:

- PL is the number of bytes of payload,
- SF is the spreading factor,
- IH signals which header mode is used: with implicit header mode IH = 1, with explicit header mode IH = 0,

- DE set to 1 indicates the use of low data rate optimization, otherwise it is set to 0,
- CRC is equal to 1 when the CRC field is present, 0 when not present,
- CR is the programmed coding rate from 1 to 4.

Low data rate optimization is used to increase the robustness of the LoRa link when high spreading factors are used; it is mandated with SF11 and SF12 at a 125 kHz bandwidth.

Figure 2.4 shows the spectrum of the transmission of one LoRa packet: at the beginning, the preamble sequence of constant upchirps spanning the entire available bandwidth is visible. Figure 2.5 represent the results of the decoding process used in [4] to analyze a LoRa modulated signal. To obtain it, the authors first ”de-chirp” the signal, then take its Fast Fourier Transform (FFT) with a number of bins equal to the number of symbols M that can be represented with the spreading factor in use. Secondly, since the signal can now be interpreted as if it was modulated using Multiple Frequency Shift Keying (MFSK), they take multiple overlapping FFTs and, to detect the symbol at each time frame, select the bin with the highest power content. Again, the constant part at the beginning indicates the constant symbols used in the preamble.

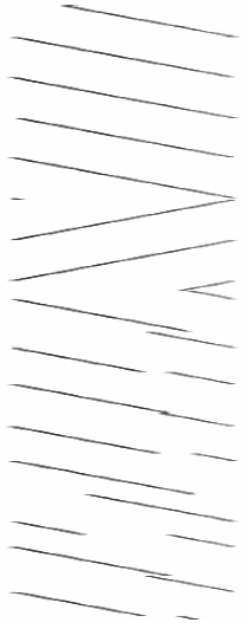


Figure 2.4: Spectrogram excerpt of a LoRa chirp. *x-axis*: Frequency, *y-axis*: Time [4].

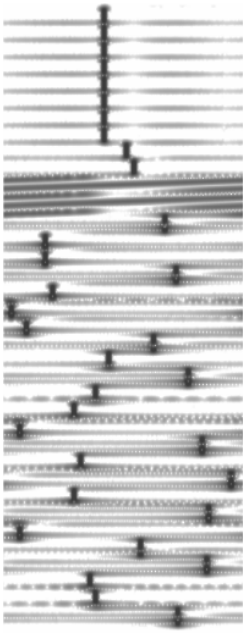


Figure 2.5: Dechirped signal (preamble and body) [4].

To make data more robust, LoRa applies to packets a set of operations before sending them:

Interleaving This procedure scrambles data bits throughout the packet, and is often used with FEC: in this way, if interference affects data, corrupted bits will be spread in the whole packet after the de-interleaving process, avoiding burst of errors and making the correction operation more effective. In [4] it has been found that the interleaver is a diagonal interleaver with the two most significant bits reversed, slightly different from the one described in Semtech's patent.

Forward Error Correction (FEC) LoRa uses Hamming FEC with a variable codeword size in the range from 5 to 8 bits and fixed data size of 4 bits per codeword, as previously described.

Data Whitening This technique is applied to induce randomness in the sequence of symbols, providing more features for clock recovery. It is particularly useful when there are long runs of equal bits. To recover the signal, data are XORed against the same whitening sequence used by the transmitter.

Gray map This encoding procedure maps a block of SF bits into one of the M symbols in the costellation and, by definition, makes sure that two adjacent symbols differs by at most one bit. In this way, when applying FEC it is easier to successfully correct symbols that differ by one bit.

2.2.4 Additional LoRa modulation properties

Besides to the use of CSS modulation, that allows the recovering of signals having very weak power, LoRa modulation presents two particular advantages that make it very competitive in the IoT market: spreading factor orthogonality and capture effect.

Spreading Factor Orthogonality

LoRa spreading factors are pseudo-orthogonal, even when the same channel and the same bandwidth are used. This means that if an ED transmits using spreading factor i, it can be correctly received by the GW even if it is overlapping with another transmission using spreading factor j, as long as $i \neq j$. Even if in [30] the LoRa spreading factor orthogonality is mathematically demonstrated, different works ([2], [31]) state that it is not a complete orthogonality, but, depending on their power, some spreading factors can cause interference. It is also observed that packets sent with higher DRs are affected by interferes more those ones sent with low DRs. However, in the same papers and in [24] it is observed that transmissions arriving at the receiver with a Signal plus Interference to Noise Ratio (SINR) above a certain threshold (also called isolation) that depends on both i and j can be correctly received.

Capture effect

The possibility of recovering a signal when its power is higher than a given value is called capture effect and it is valid also when the interfering signals use the same spreading factor, as long as the condition on the difference of the received powers holds. Literature works, as [32], show the importance of this feature in the total

performance of the network and underline that it should be taken into account when planning or simulating a LoRa network.

Capture effects and spreading factors pseudo-orthogonality make LoRa's performance better than a simple Aloha network, even though the channel access is Aloha-based: while in Aloha two colliding packets are always lost, in LoRa they can survive if they have sufficiently different powers, and the power required for a packet to survive is even lower if the spreading factors used in the transmissions are different.

2.3 LoRaWAN standard

While the PHY layer specification is proprietary and owned by Semtech, the documentation describing the protocol at MAC layer, called LoRaWAN is public and developed by LoRa Alliance, an association initiated by industry leaders, now extended to vendors and researchers, having the objective of spreading LoRa technology and guarantee inginteroperability between different operators proposing a unique open global standard. In the following, we describe the main elements that will be used in this work's analysis, while further details can be found in the last release of the specifications [5].

2.3.1 Topology and devices

LoRaWAN protocol is optimized for battery-powered nodes, which can be either mobile or in a fixed location. The network consists of three types of devices: EDs, also known as motes, GWs, also called concentrators or base stations, and a central Network Server (NS).
A star topology is employed, with GWs acting as transparent message forwarders between EDs and the NS. EDs use a single-hop LoRa connection to one or more GWs, while these last ones use IP links to connect to the NS. Messages sent by EDs (uplink transmissions) can be collected by all the GWs that can hear them; then, the backhaul server will decode the information, potentially discard duplicate packets and send, if needed, answers to the EDs (downlink transmission), choosing the most suitable GW according to a metric of choice (e.g., the one offering better radio connectivity).

LoRaWAN distinguishes three classes of EDs. They all provide bi-directional communication, but differ in functionalities and power constraints. Each device must implement at least the Class A features and optionally the others.

Class A (All) Devices belonging to this class access the channel to transmit the packet received from the application layer in a totally asynchronous fashion, implementing an Aloha Medium Channel Access Control. Each ED's uplink communication is followed by at most two short downlink receive windows, and the second one is opened in a different sub-band previously agreed with the network server in order to increase resilience against channel fluctuations. These are the most power-saving devices as receiving windows are opened only

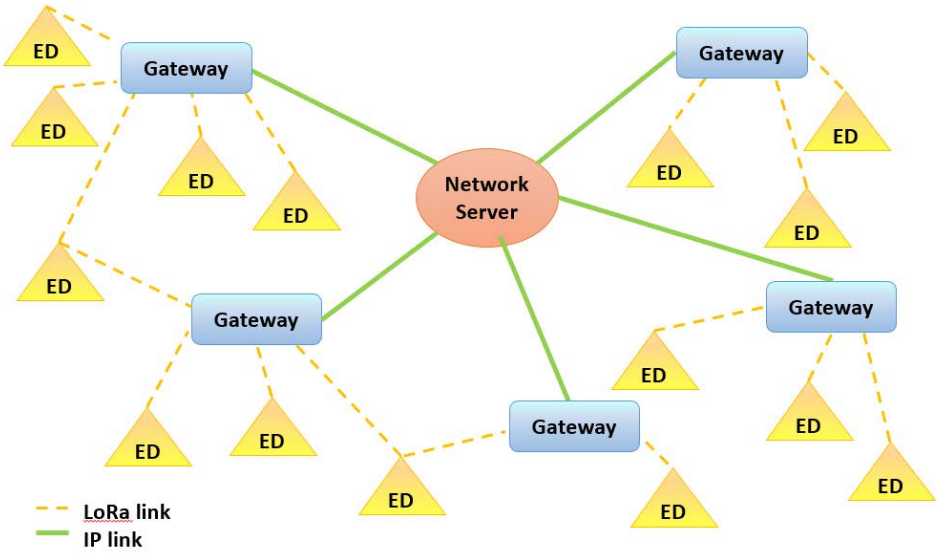


Figure 2.6: Example of a LoRaWAN network topology.

after a transmission starting from the EDs. Downlink communication could, thus, suffer a high latency. These nodes are meant to be used for monitoring applications.

Class B (Beacon) In addition to the functionalities of class A's devices, nodes belonging to class B also open some extra receiving windows at scheduled times receiving a time-synchronized beacon from the GW. This allows the server to know when the device is listening and makes these nodes suitable for remote control applications.

Class C (Continuously listening) These devices are the most power-consuming, as they keep the receiving windows open except when they are transmitting.

In the reminder of this thesis, if not explicitly stated, we will refer to the most common Class A devices, characterized by the fact of being battery powered, in accordance with a typical IoT scenario.

2.3.2 Receive windows

We give here some more information about receive windows, since they represent a key point for the implementation described in this work.

For each uplink message the ED opens two short receive windows, which are the only possibility for the communication from NS to ED. The start times of the receive windows are defined using the end of the uplink transmission as reference (see Figure 2.7).

The frequency and data rate of the RX1 are function of the corresponding values used for the uplink. By default, they keep the same data rate of the last uplink message. The first receive window is opened $RECEIVE\_DELAY1$ seconds after the end of the transmission. The length of the receive windows can be variable, but

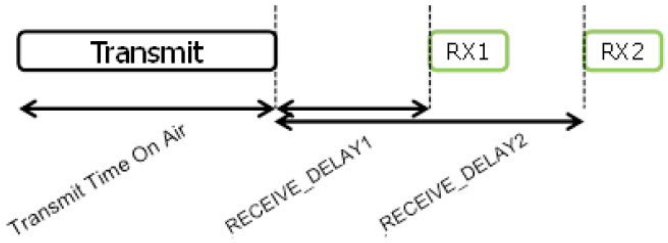


Figure 2.7: End-device receive slot timing [5]

must be at least the time required by the ED's radio to detect the downlink message's preamble. If there is such a detection, the radio receiver stays on until the message is demodulated. If, during the reception, the ED notices, from the address field, that the message is intended for another user, it stops the reception discarding the packet.

In this case or when the first receive window goes empty, the ED must open the second receive window (RX2) $RECEIVE\_DELAY2$ seconds after the end of the uplink transmission. Frequency and data rate of the second receive window have fixed values that can be configured by MAC commands

A transmission from the NS to the ED must initiate exactly at the beginning of one receive window and an ED shall not begin to transmit a new uplink message before it either has received a downlink message or the second receive window has expired.

2.3.3 LoRa MAC frame

We continue here the description of the frame structure, started in Section 2.2.3 with the explication of the PHY layer's frame. For the principle of encapsulation, the packet at MAC layer is the payload of the PHY packet (in Figure 2.8 referred to as PHYPayload), which consists of a 1-byte-long MAC Header (MHDR), a MAC Payload (MACPayload) with variable size, and a Message Integrity Code (MIC) of 4 bytes, checking the integrity of the whole message.

MAC Header

The MAC Header specifies the Message Type (MType) of the packet and according to which version of the LoRaWAN specification the frame has been encoded. In Table 2.2 we have the codes used to indicate the type of the message and the corresponding description. We focus now on the four types used for data messages: they are employed for uplink or downlink communication (*Data Up* and *Data Down*) and can require an acknowledgment from the receiver, or not (*Confirmed* - *Unconfirmed* data). We remark that for the NS the only possibility of acknowledging uplink traffic is during the receive windows, while if an ED is asked for an acknowledgment, it should answer whenever it is possible. MAC commands often require the ED to confirm the message reception and command execution.

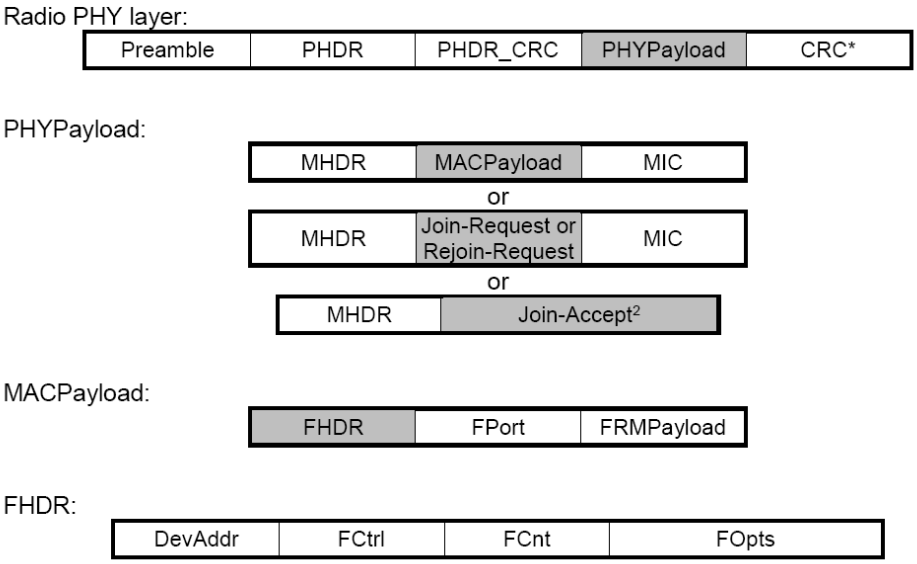


Figure 2.8: Packet structure of a LoRaWAN message [5]

MType	Description
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	Reserved for Future Use
111	Proprietary

Table 2.2: MAC message types [5]

MAC Payload

The MAC Payload consists of a Frame Header (FHDR), a Frame Port (FPort), used to identify which application the message is intended for, and a Frame Payload (FRM-Payload), containing data coming from the application layer.

The frame header contains the device address (DevAddr), a frame control field (FCtrl), a frame counter (FCnt), and an optional encrypted field carrying frame options (FOpts).

Two bits in the FCtrl field implements an important feature in LoRa: the Adaptive Data Rate (ADR) control, used to increase ED's battery life and maximize network capacity. Since each ED is allowed to use any data rate and transmission power, the network, via the NS, has the possibility of controlling them and this is referred to as ADR. This is possible only when the uplink ADR bit in the FCtrl field is set, informing the NS that it can send appropriate MAC commands to modify ED's transmission parameters. The NS will then choose to decrease spreading factor and transmission power if the ED transmissions arrive above sensitivity or increase them if the SNR

margin is too low.

If the ADR bit is unset, it is the ED that controls its own parameters, and the specification [5] states that

- The ED sets its uplink data rate following its own strategy, to better adapt to rapid changes in the radio channel if it is a mobile ED;
- The ED should ignore the uplink ADR bit (keeping it set) and apply normal data rate decay in the absence of ADR downlink commands when it is a stationary ED.

In case the ED is using data rate and transmitting power higher than its default settings and is not receiving any downlink message for a specific time, the ED can set the ADR acknowledgment request bit asking the NS for an answer. If it does not receive it in a fixed interval, the ED must try to regain connectivity, coming back, by steps, to the default transmission power and data rate. It is worth noting that the LoRaWAN specification does not describe any ADR algorithm: its implementation is left to network owners (responsible for the NS) which should find a compromise between computational efficiency and accuracy. Some papers have been recently published about this topic, and will be described in more depth in Chapter 3.

Two other bits in the Frame Control field are used as Acknowledgment (ACK) and Frame Pending bits. When the Frame Pending bit is set in a downlink frame, it means that the NS has something queued for the ED and asks it to open another receive window as soon as possible by sending another uplink message.

MAC commands are used for network administration and they can be either piggybacked in the Frame Option fiels or sent as a separate data frame, contained in the Frame Payload field with the FPort field set to 0. MAC commands are always encrypted and are answered or acked by the receiver. Different MAC commands can be exchanged in both the uplink and in the downlink directions, and enable the following functionalities:

- Resetting;
- Link checking: commands sent by EDs to validate their connectivity to the network and answered by network server with information about the received signal power;
- ADR setting;
- Duty cycle limitations, set by the network server;
- Setting of different parameters (reception slots, transmission power, downlink channel...);
- Exchange of information about the status of the ED (battery level and demodulation margin);
- Creation or modification of radio channels.

2.3.4 Device activation

The specification [5] defines two ways to activate an ED and allow its participation in a LoRaWAN network: Over-The-Air-Activation (OTAA) and Activation By Personalization (ABP).

In the first one the ED sends a request to join the network; the network server checks if that specific device is allowed to join the network (if, for example, the device has a valid subscription or has not been banned from the network for security problems) and, if so, answers with a join accept message. During this procedure, the ED is also given keys (an application key and a network key) for message encryption and decryption. Higher security is given by the fact that these keys are computed basing on root keys specific to the ED and assigned to it during fabrication: if a key is extracted from an ED, it only compromises that specific ED and not the whole network. The over-the-air procedure can be repeated multiple times (for example, every day) and each time new keys are computed and provided to the device, further increasing the security of the network. With the activation, the ED is also given a device address (DevAddr) that uniquely identifies it in the network.

With the Activation-by-Personalization procedure, the ED can participate in the network as soon as it is started, without requiring the joining procedure, since the device address and all the keys are already stored into the device.

2.3.5 Retransmission procedure

Since this work will deal with confirmed traffic and retransmissions, we provide here some additional detail about this procedure, as specified in the standard [5].

When a device (both ED and NS) receive a confirmed data message, it shall answer with a data frame having the ACK bit set: the server will answer in one of the two receive windows while the ED will reply at its own discretion, by sending an empty frame with ACK immediately after the reception or piggybacking it to a following data message. Uplink messages are transmitted for a maximum of *NbTimes*, until a valid downlink is received following one of the transmissions, and consequent retransmissions should be done at least $ACK\_TIMEOUT$ seconds after RX2. The parameter *NbTimes* can be set by appropriate MAC commands and its value ranges from 0 to 15. It is used to control the redundancy and obtain a given Quality of Service (QoS) that can depend on the application in use. The value of $ACK\_TIMEOUT$ is randomly chosen in the interval [1, 3] s.

It is suggested to use frequency hopping also in the retransmission procedure and to delay retransmissions until the receive windows have expired. Moreover it is recommended to stop further retransmissions if the corresponding downlink acknowledgment frame is received.

In the previous version of LoRaWAN [33] it was also remarked that in addition to selecting a different frequency channel, retransmissions could also use a different (lower) data rate: the recommended scheme was to decrement the data rate by one every couple of failed transmissions, until the lowest possible data rate is reached. Any further transmission will employ the last data rate used, and the recommended maximum number of transmissions is 8. For example, let's suppose that an ED sends a confirmed frame first using DR5. If it doesn't receive any ACK, after the eighth

Region	**Frequency band (MHz)**
Europe (EU)	863 - 870 and 433
United States (US)	902 - 928
China (CN)	779 - 787 and 470 - 510
Australia (AU)	915 - 928
Asia (AS)	470 - 510
South Korea (KR)	920 - 923
India (INDIA)	865 - 867

Table 2.3: Regions and frequency bands [10].

transmission, it will be set to use DR2 and the packet will be dropped. A following application packet will initiate a new transmission cycle, where the first two attempts will be done at DR2, then the end device will switch to DR1, then to DR0.

2.4 Regional Parameters

In addition to the LoRaWAN specification, the LoRa Alliance also publishes a document complementary to the standard, that describes how LoRaWAN parameters should adapt to the different regions. As already mentioned, this is due to the fact that the available ISM band is not the same everywhere, and the local regulations can be more or less strict. Table 2.3 shows the different regions and the frequencies of the ISM band. In the following, we will focus our discussion on the EU 863-870 MHz ISM band.

Even if the ISM band is unlicensed, users have to respect some regulations dictated by different entities: the National Administrators, organisms at the European level and the International Telecommunication Union (ITU) at the worldwide level [34]. In particular, in European countries, LoRa is required to respect the European Telecommunications Standard Institute (ETSI) regulations. The restrictions concern the physical medium access, such as maximum transmission time. Radios are required to either adopt a Listen-Before-Talk (LBT) policy or duty cycled transmissions, to limit the rate at which messages are generated. The duty cycle is defined as the ratio of the maximum transmitter "on" time over one hour, expressed as a percentage. For example, a device with a 1% duty cycle, can perform 10 transmissions of 3.6 seconds within one hour, while if the same device had a 10% of duty cycle, the transmissions could last 36 seconds. To keep ED simple and save energy, the duty cycle option is adopted in LoRa. This can have an important influence when the packets are generated frequently or when the transmitter uses high SFs (and thus longer transmission times) since it limits the activity of the device.

Network operators can freely assign channels provided that they are respectful of this constraint. However, the specification defines a minimum set of channels (see Table 2.4) that must be implemented and cannot be modified, so that to guarantee in any occasion a minimal common channel set between ED and GWs. In fact, each GW should always listen to at least these three channels.

Modulation	Bandwidth [kHz]	Channel Frequency [MHz]	LoRa DR / Bitrate	Number of Channels	Duty Cycle
LoRa	125	868.1 868.3 868.5	DR0 to DR5 / 0.3 - 6 kbps	3	<1%

Table 2.4: Mandatory channels in EU863-870 ISM band [10].

DataRate	Configuration	Indicative physical bit rate [bit/s]
DR0	LoRa SF12	250
DR1	LoRa SF11	440
DR2	LoRa SF10	980
DR3	LoRa SF9	1760
DR4	LoRa SF8	3125
DR5	LoRa SF7	5470

Table 2.5: Data rate table for EU863-870 band with bandwidth of 125 kHz [10].

ETSI also limits the effective power radiated by the devices. We first give the definition of Effective Isotropic Radiated Power (EIRP): it is the power that an isotropic antenna should employ in order to get the same field strength that the tested device produces at the same distance. It is defined by the following formula:

$$EIRP_{dBm} = 10\ log\left(\frac{E^2 \cdot r^2}{0.03}\right) \tag{2.8}$$

where E is the electrical field strength at a specified distance from the transmitting antenna and r is the distance.

The Effective Radiated Power (ERP) has a similar definition to the EIRP but it uses a half-wave dipole instead of an isotropic antenna, that has a gain at most equal to 2.15 dBi. Thus, the relation between the two is

$$ERP = EIRP - 2.15 \tag{2.9}$$

In its regulations, ETSI uses the ERP metric to describe the limitations of the devices' transmission power. In [10] the possible values for the transmission power are given, and the maximum is equal to 16 dBm. The document also gives the values of the bit rates that can be achieved when using the different bands. In Table 2.5 we report the ones relative to the mandatory channels in the EU868-870 ISM band.

In Table 2.6 we report the default values of some parameters that have been mentioned in the previous sections, but that take different numerical values according to the region. The values we list are the ones for EU863-870 band.

Parameter	Value
RX1 frequency	same as uplink transmission
RX1 data rate	same as uplink transmission
RX2 frequency	869.525 MHz
RX2 data rate	DR0 (SF12, 125 kHz)
$ACK\_TIMEOUT$	2 ± 1 s
$RECEIVE\_DELAY1$	1 s
$RECEIVE\_DELAY2$	2 s = $RECEIVE\_DELAY1 + 1$ s

Table 2.6: Default values of some parameters in the EU863-870 band [10].

It is interesting to note that when LoRa devices are activated in territories using different frequencies, they should be able to automatically identify the frequency to use before sending their messages, especially when the Over-The-Air-Activation is used. To do that, LoRaWAN specification [5] suggests that EDs should use GPS location if they are equipped with it, or search a Class B beacons and identify the region using its frequency or a combined method.

Chapter 3

State of the art

A number of works studying LoRa and its performance have been published in the literature in the last few years. In this chapter, we will give an overview of the most important contributions related to the analysis carried out in this thesis. The contributions are grouped according to specific topic: experimental results, considerations on scalability and SF orthogonality, evaluation of ADR algorithms, mathematical models and simulators.

3.1 Experimental measures

Various works present empirical results on the coverage of LoRa. In [35] a channel model based on experimental measures of LoRa is developed. The experiments are conducted in both urban and open air scenarios and show that the packet loss ratio depends on the distance and on the environment. To this goal, an EDs equipped with a Semtech SX1272 transceiver was set to use SF 12 and a 125 kHz bandwidth. The base station sensitivity for this setup was -137 dBm. In the urban scenario the maximum achieved range was 15 km. However, the communication was not reliable, as only 74 % of the packets were not correctly retrieved at the base station. Better performances were achieved at lower ranges, where the packet loss ratio was 12 % and 15 % in the ranges of 0-2 km and 2-5 km, respectively. To verify the achieved range in open air scenarios, similar measures were performed on the sea, with the EDss mounted on a boat. Here, the packet loss ratio was much smaller: 31 % in the range 5-15 km and 38 % from 15 to 30 km thanks to the line of sight between GW and EDs.
Another work that analyzes LoRa's coverage is [28]: authors claim that a single LoRa GW can cover a cell of about 2 km of radius using the highest SF value. Therefore, to avoid that high SFs affect too much the performance of the network, authors assume a nominal cell-range of 1.2 km and estimate that 30 GW can properly serve a city of about 100 square kilometers and 200 thousands inhabitants.
More recent evaluations are those of [36, 37]. In [36] the maximum range achieved in the city was about 2.2 km and it was verified how environmental obstacles as hills, buildings, walls influenced the signal reception. However, it is also stated that the problem could be overcome by employing multiple GW.

3.2 Scalability and SFs' pseudo-orthogonality

A feature that is often investigated in LoRa is how much it is scalable. It is commonly recognized that an increase in the number of EDss accessing the networks leads to a lower throughput when a high traffic is involved, since collisions happen more frequently. Authors of [38] use the LoRaWANSim simulator to analyze a network with downlink transmissions. They underline how the optimal number of transmissions depends on the application and on the considered scenario and that there are scalability problems if a large number of devices requires confirmed messages, since ACKs occupy both channel and GW, that, being subject to duty cycle constraints, are not able to answer all the requests. At the same time, another parameter that influences the network throughput is the spreading factors used by the EDss. In [6] it is noted that, in addition to the time-on-air, large SFs also increase the off-period caused by the duty cycle regulations and that the problem if further increased by the fact that high SFs are used more often. However, as illustrated in Figure 3.1, the duty cycle also prevents the network to exceed a given load and stabilizes the throughput when a high transmission rate is used. In the analysis presented in 3.1, SFs are considered orthogonal and the capture effect is not taken into account.

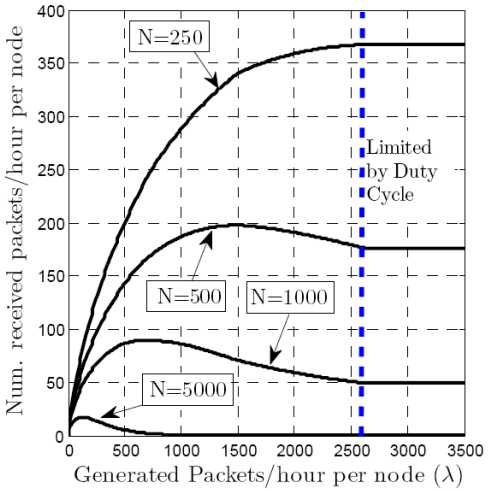


Figure 3.1: Number of received packets per hour and per node, for 250, 500, 1000, 5000 EDs and 3 channels, as a function of the packet generation frequency [6].

Different works analyze the pseudo-orthogonality of the SFs. In [31] this feature is studied using first a Matlab code, then it is compared to measures taken from commercial devices, and finally the open-source LoRaSim simulator is extended to consider inter-SF collisions. The authors find that SFs are not perfectly orthogonal and therefore do not form a set of independent logical channels. Smaller SFs need a higher SINR to achieve an acceptable Packet Error Rate (PER). The same result is also observed in [39] where empirical measures were taken. Here, Mikhaylov et

al. also state that, when two devices are transmitting with the same data rate, they interfere, but one transmission can survive if its receive power is sufficiently larger then the other. When different SFs are used, the power gap should be at least 6 dB. They also compare LoRa and Gaussian Frequency Shift Keying (GFSK) modulations and verify that the first one is more robust to interference.

Another evaluation of LoRa virtual channels orthogonality is given in [40]: with experimental simulations conducted using three Semtech's SX 1272 modems, a function generator and a LoRa packet forwarder using Semtech SX1301 transceiver, authors analyze the effect of delay and SF in a packet collision. Their results show that packets using the same SF are more subject to collisions, while acceptable PER values can be achieved by using different SFs and a time distance of 4 ms between the transmissions. Therefore, they point out that the use of different SFs is an effective approach for increasing the number of nodes in the same area.

3.3 ADR algorithm

More recent investigations focus on the ADR algorithm. In [41] authors evaluate the open ADR algorithm used by The Things Network [42], and propose some modifications. With empirical evaluation of the relation between PER and Signal to Noise Ratio (SNR) they obtain a channel model that is later used in Matlab simulations. Some changes in the algorithm, make it possible to achieve better error performance in harsh channels and reduce the number of both retransmissions and downlink commands from the network server.

In [43] it is shown how a SF assignment based on the distance between GW and EDs brings to unfair performance: in order to gain connection, farther devices will be assigned high SFs and therefore will be penalized with a longer transmission time, higher collision probability and longer off-period. Moreover, the capture effect favors nodes located near the GW, as the received power will be higher. Different solutions have been proposed to cope with this problem.
In [7] the objective of the discussion is to find a way to assign SFs in order to enhance the packet error rate fairness inside a LoRaWAN cell by optimizing power and SF for each node; in particular, power control is important because a threshold SNR is required to successfully decode simultaneous transmissions. The discussion firstly considers the optimal SF distribution with unconstrained power control, which will counteract path losses especially for nodes far from the GW, and results in all nodes achieving the same collision probability. Secondly, discrete and constrained power control is considered: in this scenario the collision probability also depends on the distance from the GW, besides the SF used by other nodes, as most powerful transmissions will be selected. Authors note that in this case, if the maximal difference of path loss is below 30 dB, the combination of power control and spreading gain of CSS is able to provide the required SINR to all nodes. If the power difference is higher than 30 dB, the node with lower received power will lose its packet. Therefore, close nodes will minimize their impact on the network by using the lowest SF to minimize collision probability. As a large number of nodes will use the same low SF, authors propose a scheme to split traffic over different channels and frequencies, writing an algorithm that should be run by the network at regular time intervals. Fig-

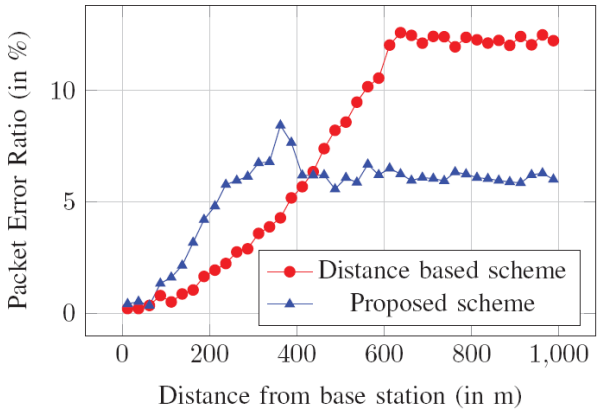


Figure 3.2: Comparison of PER achieved by a distance based scheme and the algorithm proposed in [7].

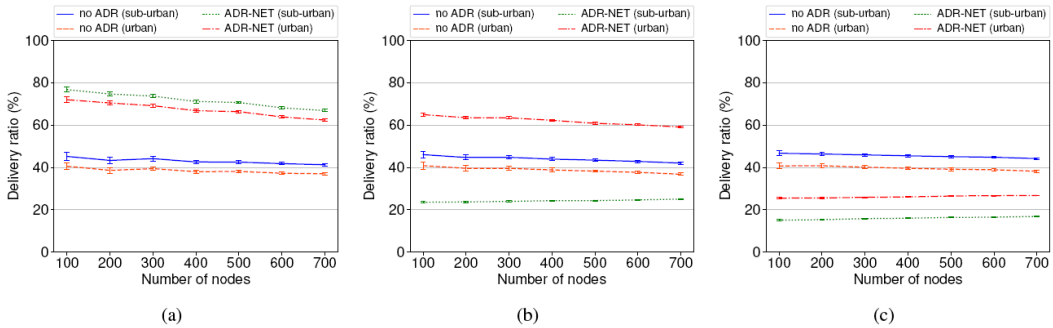


Figure 3.3: Impact of ADR-NET on the delivery ratio in networks with different channel conditions: (a) ideal, (b) moderate variability and (c) typical variability [8].

ure 3.2 shows how the proposed scheme improves performance of nodes far from the station, while the PER for devices close to the GW increases because their traffic is grouped on the same channel, to let other channels free for the farther nodes. Authors also admit that this improved fairness comes at the cost of an increased energy consumption compared to that of the distance based scheme.

In [8] authors analyze ADR performance whith different channel conditions by developing a LoRa simulator in OMNeT++, called FLoRa. They note that ADR is effective in increasing the network delivery ratio when the channel is stable, while it is deterimental in case of variable channel conditions (Figure 3.3). Authors modify the ADR algorithm used by The Thing Network (indicated in the paper as ADR-NET), proposing a new version of the algorithm, called ADR+. In particular, while ADR-NET algorithm estimates the link quality by using the maximum SNR value of the last 20 samples, ADR+ uses the average function to solve the problem of high variability in fast-fading conditions. As it can be seen in Figure 3.4 the modified algorithm achieves better results even when the channel variability increases. Authors also point out some weak points in the ADR algorithm proposed by The Thing Network and note that the two algorithms choose SF and transmission power based on each link: this approach, however, does not consider the collision probability and

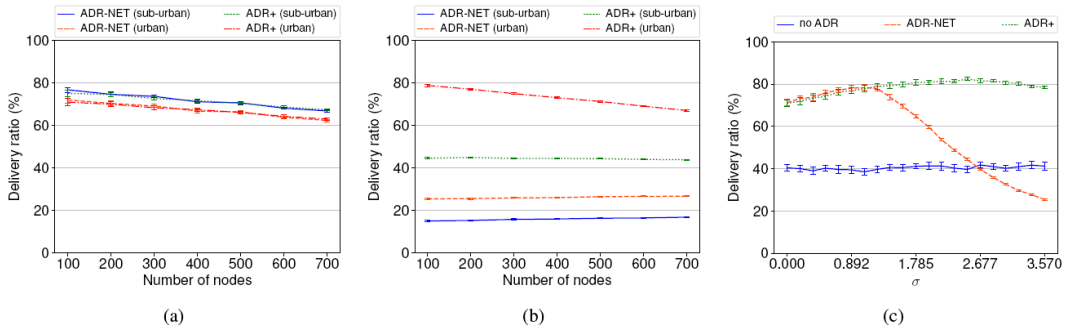


Figure 3.4: Comparison of delivery ratio of ADR-NET and ADR+ in networks with different channel conditions: (a) ideal, (b) typical variability. (c) Delivery ratio in urban scenarios (100 nodes) as a function of the path loss variance σ [8].

an algorithm that assigns parameters based on the knowledge of the whole network should be employed.

This consideration is taken into account in [44]: the approach is similar to the one applied in [7] but here, instead of applying the fair ADR algorithm to the whole network, authors propose to execute it over regions. They implement their solution in LoRaSim simulator, considering also the effect of imperfect SF orthogonality. With the proposed changes, fairness is improved and energy consumption is lowered with respect to the results of [44].

3.4 Models and simulations

While a part of the literature explores LoRa performance based on empirical evaluations, another approach consist in the mathematical modeling of a LoRa network to predict its performance. A first step is done in [45], where authors start from empirical evaluation on interference to show how the collision probability depends on both power and delay of the transmissions. In particular, it is observed that, depending on the overlap, a packet can be lost, correctly received or received (e.g., recognized by the destination node), but with errors in the CRC. The distinction between the last two cases is done because, they claim, the fact that a packet is received by a given GW could be useful in tracking applications. However, to the best of our knowledge, the device address is contained in the MAC payload, and therefore protected by the CRC code. Hence, in case of CRC failure, it would be impossible for the gateway to retrieve the sender address, and the packet would be useless. The simple model proposed by the authors shows that LoRa performs better than pure Aloha, and that a configuration using multiple SFs and multiple channels should be preferable to a setting where a unique SF - or a unique channel - is used.

In [46] a stochastic geometry model is used to jointly analyze interference in the time and frequency domain. The model is applied both to the Sigfox and LoRa technologies and it results that Sigfox can support more devices but achieves a throughput similar to LoRa, which can transmit larger packets. Moreover, it is observed that, for LoRa, the repetition mechanism reduces the outage probability (e.g., when each

packet is repeated n times, the outage probability is defined as the probability that the SINR value of all the n transmissions is lower than a threshold), but also limits the average throughput because of the introduced redundancy.

The author of [47] proposes closed-form expressions for collision and packet loss probabilities, based on the assumption of perfect SF orthogonality and without considering the capture effect (a time overlap in the transmissions corresponds to a collision, while a packet is considered successful when the corresponding ACK is received by the ED). Furthermore it is assumed that the GW is not able to receive any packet when it is transmitting an ACK. With simulations, the author of [47] shows that the results obtained using the formulas proposed in the paper are more accurate than those achieved with a Poisson distributed process.

The analysis conducted in [43] investigates the performance of LoRaWAN uplink transmissions in terms of latency, collision rate and throughput, considering duty cycle constraints and assuming exponential inter-arrival times, no capture effect and orthogonality between SFs. The solution is based on queue theory and channel access is modeled as Aloha, with scaling factors in the traffic load. In the results, authors observe that lower latencies and higher capacities could be achieved by using sub-bands and combinations of sub-bands with high duty cycles and that the collision rate depends on how the traffic is split in the different channels. Also, this paper shows that the devices reach their capacities before the collision rate comes close to 1, as the duty cycle limits the collision rate for each band and arrivals beyond the capacity of each device would be dropped.

A model that inspired part of the work in this thesis is that developed by Bankov et al. in [48] where the authors give a first model of LoRaWAN channel access procedure considering also retransmissions. The model is extended in [49] by taking into account the capture effect. In the discussion, the authors assume perfect orthogonality between SFs, a Poisson process for packet generation at the nodes, and systematic transmission of two ACKs: one for each receive window. Instead the authors do not consider the multiple receive-path architecture at the GW. In this thesis, we will follow a similar approach for the mathematical evaluation, but we will base our work on different assumptions and considerations.

In [50] an analysis of LoRaWAN performance is conducted based on the results obtained with simulations in ns-3. With the help of Matlab, authors build an error model for the channel based on baseband simulations of a LoRa transceiver over an AWGN channel and describe the effects of path loss and intra-LoRa interference. They write a ns-3 module implementing the LoRa PHY and MAC layers. Simulations are run to observe how the type of traffic (confirmed or unconfirmed) and the addition of downlink messages impact on the performance. It is observed that a SF distribution based on PER values performs better than a random or a fixed distribution; the Packet Delivery Ratio (PDR) decreases when confirmed traffic is used, while sporadic downstream data traffic has a limited influence. It is also shown that the use of multiple GW increases the performance of the network, as the distance of the nodes to the closest GW is reduces, allowing for a smaller SF, so that the transmission time of both uplink messages and ACKs sent in RX1 decreases.

Even if using the same network simulator and considering similar performance metrics, the approach taken in this thesis is different from the one followed in [50]. Our work is based on the ns-3 module developed in [34], where the author studies the performance of a LoRaWAN network whith only uplink traffic by observing the influence of SFs and number of GWs on throughput, success probability and coverage in presence of shadowing and buildings. The module has been extended to support the retransmission procedure and the ADR algorithm running in the nodes. Extensive simulations have been carried out to see how retransmissions and ADR influence network performance when different scenarios are considered. Moreover, we develop a mathematical model to predict packet success probabilities and verify its accurancy by comparison with simulations.

Chapter 4

Network model

As many aspects influence the real behavior of the network, depending on both the scenario in which the network is deployed and on the features and devices characterizing the network itself, some assumptions and simplifications need be made to obtain a feasible model, leading to computationally viable simulations while being however accurate and meaningful in the description. The LoRa model used in this work is presented in the following, with a particular focus on the contribution of this thesis. Further information on the already existing implementation can be found in [34].

4.1 The channel

When a transmission takes place, different factors influence the correct reception of the packet. The elements that should be considered are fast fading, shadowing and path loss, together with the architecture of the transceiver module and the features of the devices, the transmit power P_{tx}, and the antenna gains of transmitting and receiving devices (G_{tx} and G_{rx} respectively). Assuming that the fast-fading Rayleigh-distributed component is averaged out by the chip modulation, the received power of a signal can be computed as

$$P_{rx} = \frac{P_{tx} \cdot G_{tx} \cdot G_{rx}}{L} e^{\xi} , \tag{4.1}$$

where L is the path loss and e^{ξ} is a factor modeling the shadowing via a lognormal random variable with $\xi \sim N(0, \sigma^2)$. Writing (4.1) in the logarithmic domain, with $10\,\xi\, log_{10} e = 4.34\,\xi$, we get

$$P_{rx}^{dB} = P_{tx}^{dB} + G_{tx}^{dB} + G_{rc}^{dB} - L^{dB} + 4.34\,\xi . \tag{4.2}$$

The path loss L^{dB} results by the sum of two contribution:

$$L^{dB} = PL^{dB} + L_{buildings}^{dB} \tag{4.3}$$

where PL represents the *propagation loss*, with the attenutation depending on the distance between transmitter and receiver, while $L_{buildings}$ is the *building penetration loss*, the attenuation caused by buildings walls when urban scenarios are considered.

4.1.1 Propagation loss model

To compute the propagation behavior, different models have been proposed in the literature. We will use the Log-distance propagation and the Okumura-Hata path loss models.

Log-distance path loss model

According to [51], the propagation loss can be computed as

$$PL_{LD}^{dB} = 40(1 - 4 \cdot 10^{-3} \cdot h)\, log_{10}R - 18\, log_{10}h + 21\, log_{10}f + 80 \qquad (4.4)$$

where $h \in [0, 50]$ m is the GW antenna elevation measured from the rooftop level, R is the distance in kilometers and f is the carrier frequency in MHz. Assuming $f = 868$ MHz and $h = 15$ m, Eq. (4.4) yields

$$PL_{LD}^{dB} = 7.7 + 10 \cdot 3.76\, log_{10}R, \qquad (4.5)$$

where R is the distance expressed in meters. This equation respects the more general formula for Log-distance path loss model with no fading:

$$PL_{LD}^{dB} = L_0^{dB} + 10 \cdot \gamma log_{10}\left(\frac{R}{d_0}\right) \qquad (4.6)$$

where L_0^{dB} is the path loss at the reference distance d_0 and γ is the path loss exponenth.

Okumura-Hata path loss model

A different way to represent the propagation loss is given by the Okumura-Hata model, also used in the analysis presented in [49]. The path loss is described by

$$PL_{OH}^{dB} = A + B\, log_{10}(d) \qquad (4.7)$$

where

$$\begin{aligned} A &= \alpha - 13.82\, log_{10}(h_{GW}) - a(h_{mote}), \\ \alpha &= 69.55 + 26.16 log_{10}(f) \\ B &= 44.9 - 6.55\, log_{10}(h_{GW}) \end{aligned}$$

and $a(h_{mote})$ is a correction factor depending on the city size. In our case, its value is

$$a(h_{mote}) = 3.2\,(log_{10}(11.75\, h_{mote}))^2\,.$$

4.1.2 Building penetration loss model

For each device, the building penetration loss $L_{buildings}^{dB}$ is given by the sum of three contributions:

1. losses caused by external walls of the buildings, indicated as EXL;
2. losses caused by internal walls of the buildings;

Probability	Range r
0.25	[4, 11] dB
0.65	[11, 19] dB
0.1	[19, 23] dB

Table 4.1: Possible distributions of the EWL uniform random variable

3. gain in the received power given by the fact that a device is located higher than the first floor, indicated as G_{FH}.

EWL is modeled as a uniform random variable taking values in a certain range: EWL $\sim U(r)$. The ranges and the probability that a node experiences that kind of loss are given in Table 4.1. The variety of values that this variable can take is intended to model walls built of different materials and with various thicknesses.

The contribution of internal walls is expressed as the maximum between two values, Γ_1 and Γ_3. Γ_1 represents the loss due to the number of internal walls:

$$\Gamma_1 = W_i \cdot p\,, \tag{4.8}$$

where W_i models the material of the walls and takes values in the [4, 10] dB range according to a uniform distribution, and p is the number of penetrated walls. The value of p is assumed to be equal to 3 with 15% of probability, while it is equally distributed among $\{0, 1, 2\}$ with complementary probability. The second value, Γ_3, is needed to model internal wall loss:

$$\Gamma_3 = \beta \cdot d\,, \tag{4.9}$$

where β= 0.6 dB/m is the penetration distance coefficient, and d is the penetration distance, uniformly distributed in the [0, 15] m range. The G_{FH} contribution is given by

$$G_{FH} = n \cdot G_n\,, \tag{4.10}$$

where n is the floor number, assumed uniformly distributed in the range [0, 4], and G_n = 1.5 dB/floor is the gain caused by increase in height given by one floor.

Then, the total loss due to buildings for an indoor end device is:

$$L^{dB}_{buildings} = EWL + max(\Gamma_1, \Gamma_3) - G_{FH} \tag{4.11}$$

4.1.3 Correlated shadowing

In the literature, many studies like [52] show how shadowing correlation affects various aspects of wireless networks, such as interference power and handover behavior. Since the shadowing modeling and implementation are the same of [34], in the following we will only give an overview of the main characteristics.

In particular, two kinds of correlation should be considered:

1. When a transmitter sends a message to two devices that are close in the space, we expect that these devices will experience similar shadowing: if the receivers are close in the space, their line of sight will likely be blocked by the same obstacles and therefore the shadowing at these two positions will be correlated.

2. A similar situation happens when two transmitters are close to each other and communicate to the same node. The receiver will likely receive two signals affected by correlated shadowing: since two close devices are communicating to the same point, it is likely that a big obstacle situated between transmitters and receiver will block both signals.

The most common model for shadowing correlation is a decaying exponential of distance, which describes the correlation between the shadowing experienced by nodes i and j as

$$\rho_{i,j}(d_{i,j}) = e^{-d_{i,j}/d_0}, \tag{4.12}$$

where $d_{i,j}$ is the distance between nodes i and j and $d_0 > 0$ is a tunable parameter called decorrelation distance, representing the distance at which the shadowing correlation is under the e^{-1} threshold, and thus shadowing can be considered uncorrelated.

To describe correlated shadowing in a region of the space, a common way in literature is to generate shadowing maps or 2D functions describing the shadowing at each point in the map for a given transmitter position. Since these methods are computationally expensive, especially when the number of needed points is large, as in our simulations, an heuristic approach is used. Assuming a shadowing decorrelation distance $d_0 = 110$ m, a regular grid of squares of side d_0 is generated. Then, at each vertex of the grid, independent Gaussian random variables are generated. The shadowing values of each point in the map are computed as a weighted interpolation of the values given by the Gaussian random variables.

4.1.4 Interference model

In this work, the performance of a standalone LoRa network is analyzed, and therefore possible interference coming from other technologies is not taken into account. With this assumption, the study of interference focuses on how transmissions performed by different devices, at different powers and different spreading factors influence each other. It is at this moment that the pseudo-orthogonality between different spreading factors comes into play, allowing different overlapping packets to be received simultaneously.

For our discussion, we introduce the Signal to Interference Ratio (SIR) threshold matrix presented in [24] that describes how different spreading factors interact with each other:

$$\mathbf{T} = \begin{bmatrix} 6 & -16 & -18 & -19 & -19 & -20 \\ -24 & 6 & -20 & -22 & -22 & -22 \\ -27 & -27 & 6 & -23 & -25 & -25 \\ -30 & -30 & -30 & 6 & -26 & -28 \\ -33 & -33 & -33 & -33 & 6 & -29 \\ -36 & -36 & -36 & -36 & -36 & 6 \\ . & & & & & \end{bmatrix} \tag{4.13}$$

The element $\mathbf{T}_{i,j}$ represents the SIR margin (in dB) that a packet sent with SF = $x = i + 6$ must have to be correctly decoded when the interferer uses SF = $y = j + 6$. For example, if a packet is sent using SF 9 and an interferer uses SF 10, in order

to be successfully received the following condition must hold in the dB domain: $P_{tx} > P_{interf} - 23$, where P_{tx} of the transmitter and P_{interf} is the power of the interferer.

It can be noted that matrix $\mathbf{T}$ is coherent with what observed in [31, 39]: transmissions using higher spreading factors are more resistant to interference, as the SIR needed to survive is lower than the one needed by packets using higher spreading factors. For example, observing the matrix, we see that when the transmitter uses SF 7, it can survive to an interferer signal modulated with SF 12 and up to 20 dB stronger; on the other hand, when the transmitter employs SF 12 and the interferer uses SF 7, the signal can be correctly decoded even if the interference is up to 36 dB stronger than the signal. In case the same spreading factors are employed by transmitter and interferer, a correct reception only happens if the power of the transmitter is at least 6 dB above that of the interferer.

In case of multiple interfering transmissions, the considered packet survives if it satisfies the margin conditions with all the interfering packets, summing the values of the received power for each SF. Therefore, if the packet is sent with SF x and received with power $P_{rx,0}$, and we indicate with I_y the set of interferes using SF y, for each couple of spreading factors (x, y), the SIR can be written as

$$SIR_{x,y} = \frac{P_{rc,0}}{\sum_{k \in I_y} P_{rc,k}}, \tag{4.14}$$

and the considered packet is correctly received if the following condition holds:

$$SIR_{x,y}^{dB} > \mathbf{T}_{i,j}. \tag{4.15}$$

It must be observed that the elements in matrix $\mathbf{T}$ have been computed assuming a complete overlap between packets. Since this is not verified in most of the cases, the interfering power used for the SIR computation must be equalized with respect to the packet duration. Moreover, it is assumed that the interleaver will spread out the interference on the entire encoded signal. Thus, calling T_x the duration time of a packet sent with SF x, the equalized interfering power can be expressed as

$$P_{rc,y} = \frac{E_y^{interf}}{T_x} = \frac{P_{rc,y} \cdot t_{ol}}{T_x} \tag{4.16}$$

where $E_y^{interf} = P_{rc,y} \cdot t_{ol}$ is the interfering energy in the overlapping time t_{ol} and y is the spreading factor used by the interferer.

We also assume that, thanks to the channel coding technique used by LoRa and described in Section 2.2.2, a packet is always correctly received when it is above the receiver sensitivity and survives interference. This assumption is justified by the fact that the curves of the Bit Error Rate (BER) versus SINR decline sharply and the BER becomes negligible as SINR grows above the thresholds reported in matrix $\mathbf{T}$ in (4.13).

With this analysis, we can observe that two signals employing the same SF and with similar received power can be correctly decoded if they overlap for a time small enough to make the SINR computed with (4.15) and (4.16) above 6 dB for both the transmissions.

4.2 Network components

To model the network devices (EDs, GWs) the Semtech datasheets have been used as reference, while the NS is represented as a simpler node with an application running on top.

4.2.1 End-devices (EDs)

The datasheet [3] has been used as reference point to model EDs. In this document two versions of the LoRa transceiver are described: SX1272 and SX1273. The versions of the product are similar, but SX1273 has some limitations on the number of spreading factors that can be used and, therefore, on the maximum sensitivity and achieved range. For this reason, the SX1272 has been chosen both in [34] and in this work. Two aspects are especially interesting for our discussion: the sensitivity achieved by the ED, which will be discussed in Section 4.2.3, and the behavior of the node whith confirmed packets. Figure 4.1 shows the sequence of events when a confirmed transmission takes place. When modeling this procedure, the following points should be taken into account:

- The device must know which kind of message is being sent (unconfirmed or confirmed).
- If the message is confirmed, the device must wait for an ACK.
- If the device was waiting for an ACK and after the second receive window the confirmation message is not received, a retransmission needs to be scheduled by the MAC layer.
- Attention must be paid to the fact that, when the ED closes its receive windows, it could be in the receiving or decoding phase. Different cases should be distinguished:
  - No packet received in the receive window
  - Packet received and already decoded
  - Packet in reception / not already decoded .
- If the ACK is successfully received and intended for the considered device, the retransmission process stops, otherwise another transmission of the same packet is performed after a randomly chosen delay.
- The retransmission process must be stopped and the packet marked as failed after a maximum number of failed transmission attempts.
- The transmission of new packets generated by the applications should preempt the retransmission of the previous packet, which would hence be dropped.

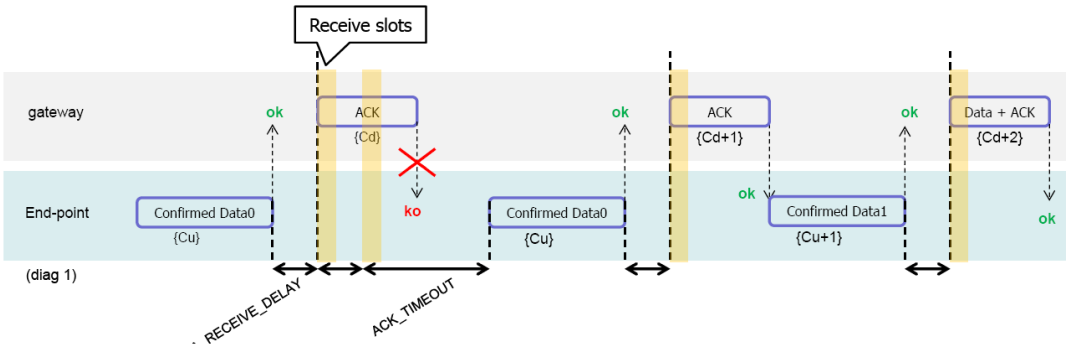


Figure 4.1: Uplink timing diagram for confirmed data messages [5].

4.2.2 Gateways

The model of the GWs follows the SX1301 datasheet [9]. An important aspect is that the GW chip contains 10 programmable reception paths, which work in parallel and are connected to the same antenna, as shown in Figure 4.2.

In particular, the IF8 LoRa channel can be configured to demodulate signals arriving at a specific SF, and is intended to serve as high-speed backhaul link to other GWs or infrastructure equipment. The IF9 (G)FSK channel presents similar features to the previous channel, but it is connected to a GFSK demodulator. The IF0 to IF7 LoRa channels are used to connect the GWs to the EDs, and have the following characteristics:

- the channel bandwidth is 125 kHz and can not be configured;
- the center frequency of each receive path can be individually configured, therefore it is possible to have multiple paths listening on the same radio channel;
- any data rate can be received without prior configuration on each receive path;
- the receiver can decode as many overlapping packets as the number of paths listening to that channel. In other words, when a packet arrives on a given channels, it will ”lock” only one receive path, while the other paths remain available to receive other incoming signals that can even overlap. Therefore, the chip is able to demodulate up to 8 packets simultaneously, and any combination of SF and channel is possible;
- The previous consideration remains valid also if the packets arriving on the same channel have the same SF, as they can be decoded by different path listening on that channel;
- If a packet arrives at a certain LoRa channel and there are no receive paths available on that channel, the packet is lost.

For our purposes, only the modeling of the LoRa channels IF0 to IF7 is important, thus we will consider 8 receive paths.

Moreover, we will assume that if the GW is in receive state and is asked by the network server to forward a packet to an ED, it will give up the receptions and

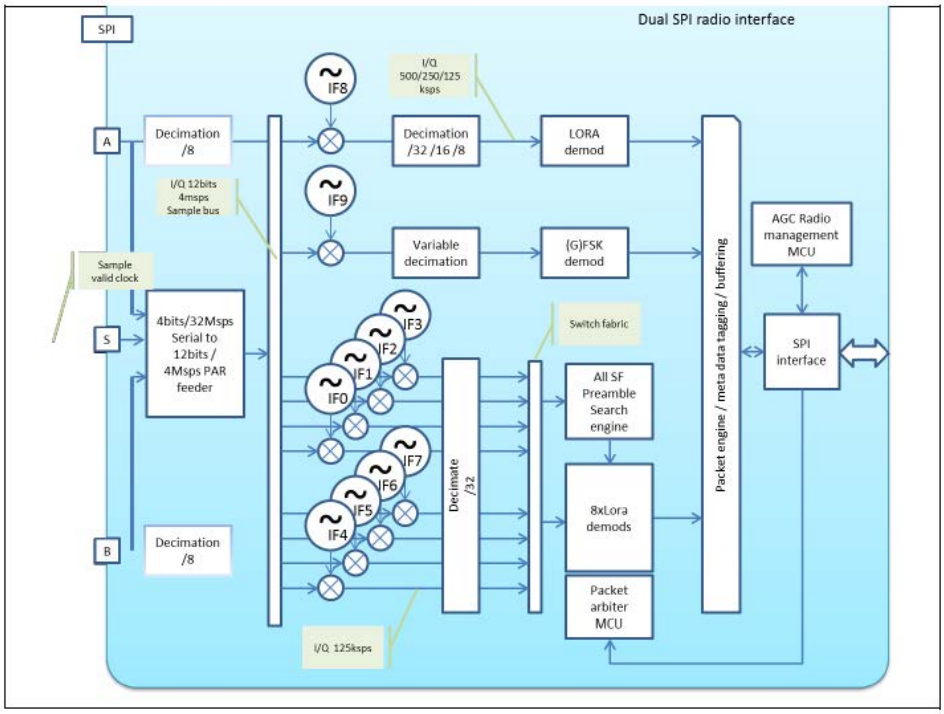


Figure 4.2: SX1301 digital baseband chip block diagram [9].

Channel index	Channel frequency (MHz)	Number of receive paths
1	868.1	3
2	868.3	3
3	868.5	2

Table 4.2: Receive path distribution among the three European mandatory channels

transmit the downlink packet. This is done for two reasons: first of all, to have the possibility of being successfully retrieved by the EDs, downlink packets must be sent when the receiving windows open. If that time frame is missed, it is useless to send the downlink message, as the end node will not be listening. Second, if the missed downlink packet is an ACK, a more conservative choice is to send it, thus it stopping the retransmission procedure rather than letting the node retransmit its packet, possibly multiple times. Even if a number of packets is potentially dropped at the GW when this behavior is adopted, this aspect is influent in the network performance when the traffic is high, as having many devices retransmitting multiple times leads to collisions, receive paths exhaustion and throughput decrease. Third, it must be considered that the frequency channel of RX2 is only used for downlink, and therefore when the ACK message is sent in RX2 it will always be successful.

For simplicity, in [34] only the three European mandatory channels are implemented. As the LoRaWAN standard defines that an ED should select the channel to use randomly, the available receive paths are equally distributed among the three channels so to maximize the coverage of each channel, as shown in Table 4.2.

SF	End-device sensitivity (dBm)	Gateway sensitivity (dBm)
7	- 124	- 130.0
8	- 127	- 132.5
9	- 130	- 135.0
10	- 133	- 137.5
11	- 135	- 140.0
12	- 137	- 142.5

Table 4.3: Sensitivity of ED (module SX1272 [3]) and GW (module SX1301 [9]) with 125 kHz mode.

4.2.3 Packet reception model

Table 4.3 summarizes the sensitivities achieved by EDs and GWs. For both devices, the sensitivity increases with the spreading factor being used, and the GW achieves a better sensitivity than EDs and, therefore, can decode weaker signals.

A packet will be detected by a device if its received power is above the sensitivity level: for example, let's consider two nodes, the first using SF 7 and the second employing SF 12. Let's suppose that they are located at the same distance from the GW, experiencing the same path loss. If the power received at the GW is -137.5 dBm for both signals, the transmission employing SF 12 will be successfully decoded, while the signal modulated with SF 7 will not even be detected.

In our model we also assume that the received power of a packet is constant for the duration of the whole packet: if the power is sufficient to make the decoding process start, the packet will is assumed to be decodable until the end of the reception.

Finally, it is assumed that if two weak signals arrive simultaneously at the receiver they can not be detected even if the sum of their powers is above the sensitivity. The reason of this choice is the fact that, the signals interfere, but it is very unlikely that they collide with constructive interference. Therefore, destructive interference is assumed and the packet would be lost even if the receiver locked on it.

4.2.4 Applications

End-device application model

EDs are configured to generate packets at periodic time intervals τ, as it is expected from many IoT sensing applications. Each device is assigned an initial random delay, to avoid the case of having synchronization among all nodes. The application periods can also be set in a uneven way, as described in Table 4.4. This distribution is taken from the Mobile Autonomous Reporting (MAR) periodic reports model. [51] also suggests to use a Pareto-distributed random variable to select the packet size; to adapt to LoRa specification, a minimum payload size of 10 bytes, a shape parameter $\alpha = 2.5$ and a cutoff of 50 bytes have been selected.

Consider the values in Table 4.4. It can be observed that almost half of the devices transmit very infrequent messages (1 packet/day) while a small percentage of nodes

Packet interarrival time τ	Percentage of devices
1 day	40 %
2 hours	40 %
1 hour	15 %
30 minutes	5 %

Table 4.4: Distribution of packet interarrival times 4.4.

generate 1 packet every 30 minutes, meaning 48 packets every day. So, two devices generating packets at the highest rate load the network with a traffic that is almost the same of 100 nodes transmitting once every day. This consideration shows the importance of the periodicity settings in the network configuration and its influence on the traffic load and, therefore, on the performance of the entire network.

Gateway and network server application models

The application implemented at the GW is a simple forwarder that transmits the packets arriving from the EDs to the network server and vice versa.

The network server application creates a downlink packet each time an uplink packet requiring confirmation is received. Then, the network server selects the GW that received the uplink packet with the highest power, and sends the downlink packet to it. Since in this work downlink packets are only used as ACK messages, they are built with zero payload and ACK bit set to 1.

4.3 System model

The mathematical model that we propose to analyze the performance of a LoRaWAN network at the MAC layer is based on that presented in [49], where propagation losses and capture effect are taken into account. Both models aim to find the transmission reliability of the network in an open air scenario, where only propagation loss is present, and is characterized by the Okumura-Hata propagation loss model (Section 4.1.1). However, different assumptions are considered, which make our model closer to the realistic network behavior:

- We consider that GWs are subject to duty cycle. ACK transmissions at the GW will also be limited by the duty cycle. We also note that the channel used in the second receive window allows a larger duty cycle.

- The model in [49] always considers the transmission of two ACKs for each data packet received at the GW: one in the RX1 and one in RX2. We think that even if this choice increments the probability that one of the two ACKs is correctly received by the ED, the second transmission could be useless if the first one is successful, and would only "consume" a transmission possibility in RX2. As the policy that should be adopted is not explicitly indicated in the

LoRaWAN specification [5], we chose to make the GW transmit only once: the window in which the ACK must be sent is selected by the NS, that knows when the GW last transmitted and in which channel. If the ACK can not be sent in either of the two windows because of the duty cycle, the transmission is not performed.

- [49] considers that "the GW cancels ACK transmission if it is receiving a data frame". As indicated in Section 4.2.2, we chose to give priority to ACKs: when the NS asks the GW to transmit an acknowledgment, the GW interrupts the reception of all incoming packets and transmits the ACK.

- We take into account the 8 receive paths at the GW: an uplink packet can be decoded only if at least one receive path is available on the frequency channel it is using.

- We recall that the frequency and SF employed in RX1 are the same as those used by the uplink packet that requires confirmation, while RX2 employs a separate downlink channel, at a different frequency and a fixed SF equal to 12.

- As in [49], we assume perfect orthogonality between SFs. A packet is lost if it collides with another signal using the same channel and the same spreading factor.

For this analysis, we will not take retransmissions into account. The model describes network performance in the case of confirmed data messages when a unique transmission attempt is employed, introducing realistic features that were not previously considered in the literature.

We consider a LoRaWAN network that consists of one GW and N motes, uniformly placed in a circle around the GW. The spreading factors are assigned according to the received power and, since only path loss is considered, this translates into a condition on the motes' distance from the central GW. We call p_i the probability that the mote employs SF i. We consider all motes transmitting a frame payload of 51 bytes in the acknowledged mode, while ACKs carry no payload. We call T_i^{data} and T_i^{ack} the time on air of a data packet and an ACK message sent with spreading factor i, while T_0^{ack} is the time duration of an ACK packet using SF 12. $F = 3$ channels are used for uplink, while a single channel with better duty cycle restrictions is used for downlink in RX2. We assume that motes generate frames according to a Poisson process with total intensity λ (the network load). Then, the load in a given channel for spreading factor i is

$$r_i = \frac{\lambda p_i}{F} \tag{4.17}$$

The capture effect is modeled considering a frame is successfully delivered if for its entire duration, its power is greater than the noise plus the power of interfering frame transmitted in the same channel and at the same spreading factor by at least CR dB. CR is the co-channel rejection parameter, specified in LoRa chip datasheets [3,9].

Since only path loss comes into play, the probabilities that a signal is stronger then the interfering signal only depend on the position of nodes. We consider the following probabilities:

- $\mathbb{S}_i^{GW}$ as the probability that, at the GW, the received power of the transmitted signal is greater than that of an interfering ED by at least CR dB, with both nodes transmitting at SF i;
- $\mathbb{S}_i^{mote}$ as the probability that, at the considered ED, the GW's signal transmitted with SF i is more powerful than the signal coming from another ED by at least CR dB;
- $\mathbb{F}_i^{both}$ as the probability that, if two devices are transmitting in the same channel and at the same SF i, neither of them is successful, because their received powers at the GW are similar.

The probabilities $\mathbb{S}_i^{GW}$ and $\mathbb{S}_i^{mote}$ are computed as in [49] considering only collisions between two packets. If more packets collide, the probabilities are assumed equal to zero. Moreover, since when two packets collide either they are both lost or only one of them survives collision, we compute $\mathbb{F}_i^{both}$ as

$$\mathbb{F}_i^{both} = 1 - \mathbb{S}_i^{GW} . \tag{4.18}$$

We observed that $\mathbb{S}_i^{GW}$ is different from zero only for SF 7. In fact, SFs are assigned to EDs based on the sensitivity: by taking two devices at the same SF, the maximum difference in their received power will be around 3 dB (which is given by the constraint imposed by sensitivity), and therefore there is no possibility of having two devices whose power difference is at least $CR = 6$ dB, as required by Semtech datasheets.

We define P_i^{data} and P_i^{ack} as the probability that the data packet and the acknowledgment sent with SF i is correctly received, respectively. Furthermore, we define

$$P_i^{S,1} = P_i^{data} P_i^{ack} , \tag{4.19}$$

as the overall success probability of the first transmission attempt, with SF i. To determine P_i^{ack} we first define P_{RX1} and P_{RX2} as the probability that the GW *can* transmit an acknowledgment in RX1 and RX2 respectively. ACKs with SF i are generated for each successfully received data packet at the GW in any frequency channel. Therefore, the ACK generation rate can be computed as:

$$\sum_{f=1}^{F} r_i P_i^{data} = \lambda \, p_i P_i^{data} . \tag{4.20}$$

The traffic that will be sent in RX1 and RX2 is hence given by $\lambda \, p_i P_i^{data} P_{RX1}$, and $\lambda \, p_i P_i^{data} (1 - P_{RX1}) P_{RX2}$, as RX2 is selected only if RX1 can not be used for transmission. Note that P_{RX1} and P_{RX2} do not necessarily sum to 1, as it could happen that in conditions of high traffic, at a given time both the receive windows can not be

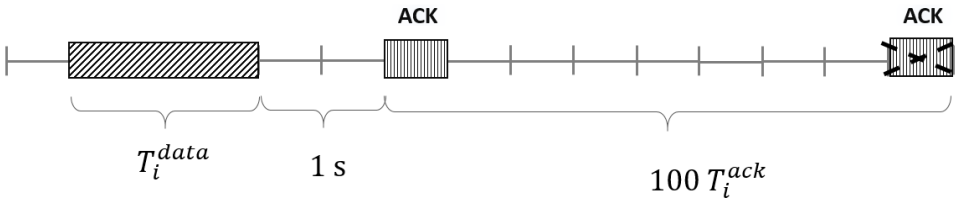


Figure 4.3: Example of case in which ACK is not sent in RX1 because of duty cycle.

used because of duty cycle limitations.
P_{RX1} is given by

$$\begin{aligned} P_{RX1} &= Pr[\text{duty cycle allows tx}] \cdot Pr[\text{GW is not already tx in RX2}] \\ &= e^{-\sum_i^{SF}(T_i^{data}+1+100\,T_i^{ack})\,P_i^{data}\,\lambda\,p_i\,P_{RX1}} \cdot e^{-\sum_i^{SF} T_0^{ack}\,P_i^{data}\,\lambda\,p_i\,(1-P_{RX1})\,P_{RX2}} \end{aligned} \tag{4.21}$$

where the first term considers the duty cycle constraint: the GW can transmit in RX1 only if no data packet caused an ACK transmission in the previous 99 slots, which means that no data packet should arrive in the interval $(T_i^{data} + 1 + 100\,T_i^{ack})$ where 1 s is the time delay between the packet transmission and the opening of RX1 in the EDs (see Figure 4.3). The second term describes the fact that the GW must not be already transmitting in RX2, and in this case the rate of ACKs is given by the fact that the GW could not transmit in RX1 *and* selected RX2 because it had the possibility of transmitting in it.

Similar considerations lead to the computation of P_{RX2}, where 2 s is the time delay between the packet transmission and the opening of RX2 in the EDs:

$$\begin{aligned} P_{RX2} &= Pr[\text{duty cycle allows tx}] \cdot Pr[\text{GW is not already tx in RX1}] \\ &= e^{-\sum_i^{SF}(T_i^{data}+2+10\,T_0^{ack})\,P_i^{data}\,\lambda\,p_i\,P_{RX2}\,(1-P_{RX1})} \cdot e^{-\sum_i^{SF} T_i^{ack}\,P_i^{data}\,\lambda\,p_i\,P_{RX1}} \end{aligned} \tag{4.22}$$

The probability that an ACK transmitted in RX1 at SF i is successfully received by the ED is

$$P_i^{ack1} = e^{-(T_i^{data}+T_i^{ack})\,r_i} + (T_i^{data} + T_i^{ack})\,r_i\,e^{-(T_i^{data}+T_i^{ack})\,r_i}\,\mathbb{S}_i^{mote} \tag{4.23}$$

where the vulnerability interval is $(T_i^{data} + T_i^{ack})$ because the ACK collides if it finds the channel already busy with a data transmission or if an ED starts a packet transmission while the ACK is being sent. The first summand describes the probability of no collisions, the second is the probability of colliding with exactly one packet whose power is lower then the power of the ACK one minus CR dB. The success probability of an ACK sent in RX2 is independent of the SF used in uplink and is $P^{ack2} = 1$, since the channel of RX2 is only used for downlink and transmissions depend on a single GW, which transmit a single packet at a time.

Therefore, the probability that an ACK is successfully received by an ED that transmitted using SF i is

$$\begin{aligned} P_i^{ack} &= P_{RX1} \cdot P_i^{ack1} + P_{RX2}\,(1 - P_{RX1}) \cdot P^{ack2} \\ &= P_{RX1} \cdot P_i^{ack1} + P_{RX2}\,(1 - P_{RX1}) \end{aligned} \tag{4.24}$$

A data frame sent with SF i is successfully received by the GW if it survives to possible collisions in the channel, finds an available receive path at the GW and the GW does not have to send any ACK while it is receiving the data frame.

The first term is given by the probability of no collision plus the probability of colliding with one packet having weaker power:

$$P_i^{data\,rx} = e^{-(2T_i^{data})\,r_i} + (2T_i^{data})\,r_i\,e^{-(2T_i^{data})\,r_i}\,\mathbb{S}_i^{GW} \tag{4.25}$$

The second term is the probability P_{rxPath} of finding at least one free receive path. As we are using the channel allocation described in Table 4.2, and the channel selection is random and uniformly distributed, the ED selects a channel associated to 3 receive paths with probability 2/3 and a channel associated with 2 receive paths with probability 1/3. This implies that

$$P_{rxPath} = \frac{2}{3}\cdot(P_{0path} + P_{1path} + P_{2paths}) + \frac{1}{3}\cdot(P_{0path} + P_{1path}), \tag{4.26}$$

where the probability P_{npath} is the probability that a packet arriving at the GW finds n paths busy. Note that the rate is multiplied by P_{rxPath} because the fact that a packet is occupying the receiving path depends on the fact that that packet found the path free.

$$\begin{aligned} P_{0path} &= Pr[\text{no packet arrival}] \\ &= e^{-\sum_i^{SF} T_i^{data} r_i P_{rxPath}} \end{aligned} \tag{4.27}$$

$$\begin{aligned} P_{1path} &= Pr[\text{exactly one packet arrival}] \\ &= \sum_i^{SF} (T_i^{data}\, r_i\, P_{rxPath})\, e^{-T_i^{data} r_i P_{rxPath}} \cdot e^{-\sum_{j\neq i}^{SF} T_j^{data} r_j P_{rxPath}}. \end{aligned} \tag{4.28}$$

The coefficient 1/2 in P_{2paths} in the case when two packets employing different SFs occupy the receive path avoids to consider twice the same couple of SFs, as we are not interested in the order (for example, $i = 7, j = 8$ and $i = 8, j = 7$ must be considered only once).

$$\begin{aligned} P_{2paths} &= Pr[\text{exactly two packet arrivals}] \\ &= Pr[\text{two packets with the same SF}] + Pr[\text{two packets with different SF}] \\ &= \sum_i^{SF} \frac{(T_i^{data}\, r_i\, P_{rxPath})^2}{2}\, e^{-T_i^{data} r_i P_{rxPath}} \cdot e^{-\sum_{t\neq i}^{SF} T_t^{data} r_t P_{rxPath}} \\ &+ \frac{1}{2}\sum_i^{SF}\sum_{j\neq i}^{SF} (T_i^{data}\, r_i\, P_{rxPath})\, e^{-T_i^{data} r_i P_{rxPath}} \cdot (T_j^{data}\, r_j\, P_{rxPath})\, e^{-T_j^{data} r_j P_{rxPath}} \\ &\cdot e^{-\sum_{t\neq i,j}^{SF} T_t^{data} r_t P_{rxPath}}. \end{aligned} \tag{4.29}$$

The data packet is not dropped during reception if the GW is not transmitting an ACK when the packet is sent and will not ne send ACKs while the data packet is being received. This probability takes into account the transmission of ACKs both in RX1 and RX2 caused by any SF:

$$\begin{aligned} P_i^{no\,ack} &= Pr[\text{no ACK sent in RX1}] \cdot Pr[\text{no ACK sent in RX2}] \\ &= e^{-\sum_j^{SF}(T_j^{data}+T_j^{ack})P_j^{data}p_j\lambda P_{RX1}} \cdot e^{-\sum_j^{SF}(T_j^{data}+T_0^{ack})P_j^{data}p_j\lambda(1-P_{RX1})P_{RX2}}. \end{aligned} \tag{4.30}$$

Finally, the probability that a data frame is correctly received at the GW is computed as

$$P_i^{data} = P_i^{data\,rx} \cdot P_{rxPath} \cdot P_i^{no\,ack} \tag{4.31}$$

and the total success probability is

$$\begin{aligned} P_{succ} &= \sum_i^{SF} p_i \cdot P_i^{S,1} \\ &= \sum_i^{SF} p_i \cdot P_i^{data} \cdot P_i^{ack}. \end{aligned} \tag{4.32}$$

Chapter 5

Simulation set up

This chapter focuses on the description of the implementation of the LoRaWAN protocol in the network simulator ns-3. After a short introduction presenting the simulator and its main features, we will describe the `lorawan` ns-3 module already implemented in [34] and the new features that were added to simulate confirmed transmissions and the retransmission procedure.

5.1 The simulator

The network simulator used in this thesis is ns-3 [53]. It is a open-source discrete-event simulator software written in the C++ programming language and intended primarily for research and educational use.

The simulator keeps track of a number of *events* that are scheduled to be executed at a specific simulation time. Examples of events can be "packet transmission" or "packet reception". The events are executed in sequential time order, and each event can schedule some other events (e.g., the event "packet reception" will schedule the event "packet decoding"). "Discrete-event" means that when an event *a* is scheduled at a given time, and the next event is *b*, the simulator will execute *a* and then immediately jump to *b* independently from the time difference between *a* and *b*. In this way, it is possible to simulate what in reality would happen in different days in just a few minutes, while keeping the simulations realistic, as the state of the network does not change in the meanwhile. Since events can schedule other events, as is done in the retransmission procedure, there is no guarantee that the queue of the events that the simulator has to execute gets empty, and the simulation could never end. To avoid such a situation, a special event is usually set to make the simulator stop at a given time.

Most of the time, random variables and random numbers need to be used in the simulations to implement parameters that don't have a deterministic value (e.g., position of the nodes, transmission times, payload size in the packets). For this purpose, ns-3 provides a built-in Random Number Generator (RNG) that produces a long sequence of pseudo-random numbers. The length of the sequence is called period, after which the RNG will repeat itself This sequence can be partitioned into disjoint *streams*, which are uncorrelated. The RNG used in ns-3 is the MRG32k3a generator

by Pierre L'Ecuyer, and is described in [54]. The generator provides $1.8 \cdot 10^{19}$ independent streams of random numbers, which can be divided into $2.3 \cdot 10^{15}$ substreams. The period of the entire generator is $3.1 \cdot 10^{57}$, while the period of each substream is $7.6 \cdot 10^{22}$. To obtain randomness across multiple runs, either the seed of the RNG or the run number can be set. The suggested method is to use a fixed seed and use different streams and substreams for each independent run of the simulator. We will use this method to get averaged results in our simulations.

A useful feature provided by ns-3 is a tracing system that allows to keep track of changes in the variables during the simulation, and optionally record their values or trigger an action. This functionality makes it possible to collect data in an automatic and easy way when the logging component is disabled to speed up simulations in ns-3's optimized mode.

ns-3 already implements multiple modules frequently used for network simulations, such as the `WiFi`, `LTE`, `pointToPoint`, and `energy` modules. Each module groups up several classes implementing the modeling of the different aspects: for example, the module `energy` contains the classes `DeviceEnergyModel`, `EnergySource`, `EnergyHarvester`. In [34] the simulator has been expanded with the `lorawan` module that we will describe in the following section.

5.2 The `lorawan` module

The `lorawan` module simulates the behavior of a LoRa network, from the devices with their applications, to the channel and its features, such as losses and shadowing. The organization of the main classes can be seen in Figure 5.1. Moreover, other classes are:

- `LoraInterferenceHelper`: to handle interfering events and decide if a packet survives or not. This class is connected to the channel and to the classes describing PHY layers, checking that packets can be successfully received by the device;
- `DeviceStatus` and `GatewayStatus` represent the network server's knowledge of the devices;
- `LogicalLoraChannel` represents a logical LoRaWAN channel, characterized by a central frequency and a range of data rates that can be sent on it and can be marked as enabled or disabled for uplink transmission. It is different from `LoraChannel`, that only delivers packets among PHY layers and computes the power and delay used in the scheduling of the reception event;
- `LoraDeviceAddressGenerator` to generate unique LoRa network addresses for the devices;
- `LoraFrameHeader` and `LoraMacHeader`, to manage packet headers at different layers;
- `LoraNetDevice` models a "LoRa network card"attached to each node. It is used to link application instances and specific LoRa classes. In this way the

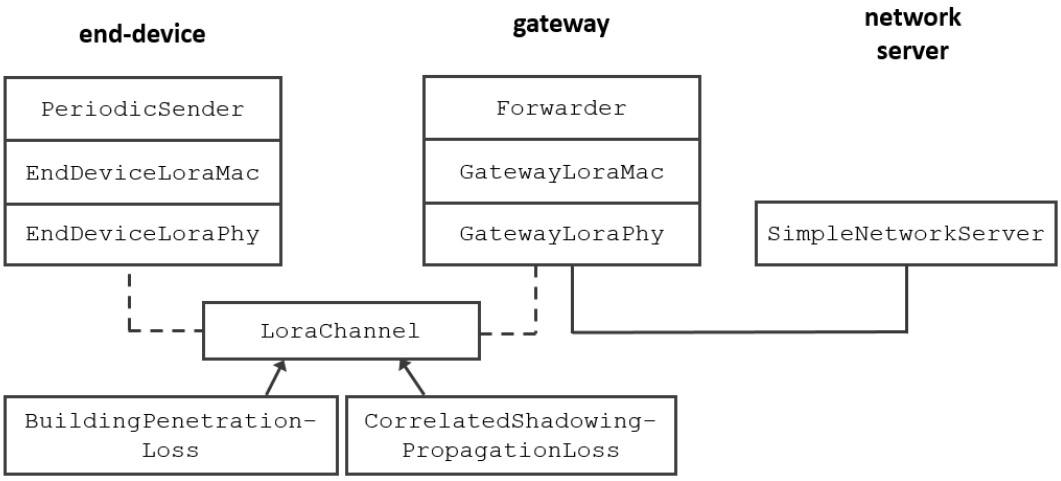


Figure 5.1: Main classes of the LoRaWAN implementation in the `lorawan` module.

application layer only generates packets, while `LoraNetDevice` takes care of forwarding them to the MAC layer's sending method with the appropriate parameters.

5.2.1 Application layers classes

`PeriodicSender`

This class implements the model described for the ED's application layer introduced in Section 4.2.4. Packets with zero-filled payload are sent at periodic intervals by each device. An attribute of the class contains the value of the application period `m_interval` that separates consecutive transmissions. The initial time at which the first transmission is performed is chosen when the application is first started on a node and selected via a random variable $d \sim U(0, \texttt{m\_interval})$. As each transmission event also schedules the packet sending event, a specific function sets the stop time for the application. Note that, at the application level, transmission simply means that the packet is forwarded down to the MAC layer (where the appropriate methods of the class implementing MAC layer functionalities will be called).

`Forwarder`

This application runs on the GWs. Since GWs are linked to EDs via the LoRa channel, and communicate with the network server via an IP link (represented in ns-3 by a point-to-point link), they also have two net devices. The forwarder application forwards packets incoming from EDs to the point-to-point net device, and packets arriving from the network server to the LoRa net device.

`SimpleNetworkServer`

This application is installed on top of a simple node equipped with links connecting it to the GWs. The first version of this application could reply with ACKs to confirmed

uplink messages. For the purposes of this thesis, some features have been modified to correctly handle multiple confirmed packets sent by EDs during the retransmission procedure.

5.2.2 `LoraMac`

The `LoraMac` class is intended to model the MAC layer of the devices, and is extended by the subclasses `EndDeviceLoraMac` and `GatewayLoraMac`. The class forwards packets from the application layer to the physical layer, taking into account duty cycle limitations. Therefore, this class will send the packets to the PHY layer only when the duty cycle allows it, otherwise the packet is queued and will be sent at a more appropriate time.

In addition to the duty cycle limitations, the `EndDeviceLoraMac` class also controls the state of the device, waking up the radio when receive windows need to be opened or switching its state when transmission or reception operations are performed. In the case of a transmission, the send method of this class chooses the frequency channel that will be used to transmit at PHY layer. In this work, the class `EndDeviceLoraMac` has been enhanced to support retransmissions and add a random delay between consecutive transmissions, as required by the LoRaWAN standard.

The class `GatewayLoraMac` implements a simple forwarder-only MAC layer and is responsible for the interpretation of MAC commands, piggybacked in the FOpts field or contained in the FRMPayload. Since this functionality was out of the scope of our analysis, it has not been implemented.

5.2.3 `LoraPhy`

This class implements the physical layer of EDs and GWs as described in the datasheets of SX1272 and SX1301 respectively [3,9]. It takes the packet sent from the MAC layer and deliver it to the channel class. Furthermore, the class decides if the packet delivered by the channel has been destroyed by interference or if it has been correctly received and can be forwarded to the upper layer.

Similarly to `LoraMac`, this class is extended by two subclasses that implement the behaviors of EDs or GWs. For every signal arriving on the channel the device is listening to, two operations are performed: the comparison of the received power with the device's sensitivity and the interference computation. Both classes compute interference via a `LoraInterferenceHelper` class that keeps track of every signal arriving in the channel the device is listening to, recording their spreading factor and received power, and compares the received power with the device sensitivity at that specific SF to decide if the signal is successfully received by the device. Then, when the packet reception ends, it is verified if the packet survived interference with the computation described in Eq. 4.16. If this control is successful the packet is forwarded to the upper layer, otherwise the packet is ignored.

The `LoraPhy` class also applies packet tags: ns-3 provides a customizable data structure, called tags, to contain information associated with a packet, to which it can be attached. We use tags to store the spreading factor used by a packet and about the spreading factor causing a loss when the packet is destroyed by interference.

The EndDeviceLoraPhy class employs the m_state attribute to distinguish different states of the chip:

- TX when the ED is transmitting a packet;
- RX when the ED is receiving an incoming packet;
- IDLE when the ED is listening to the channel for incoming packets;
- SLEEP when the ED is in low power consumption mode. In this state the device can not lock on a packet and start reception as its radio is turned off.

After transmission and reception the device automatically turns to IDLE mode, while it is the MAC layer that will subsequently put the device in the SLEEP mode. We remark that the procedure for packet reception in EDs is performed only when the device is in the IDLE state.

The GatewayLoraPhy class also needs to implement the presence of multiple paths. Therefore, a variable for each path is employed to indicate if that path is busy with reception or not. It the variable indicates that the channel is free and a transmission arrived above the sensitivity, the path is marked as busy and the packet reception can start. The variable is reset when the path is released from the reception.

5.2.4 LoraChannel

This class models the LoRa wireless channel shared by all the network devices. As the channel is the same, only one instance of the class is created for each simulation and all the devices are connected to it. This class is responsible of taking packets from the PHY layer of a transmitting device and delivering them to the PHY layers connected to the channel. In fact, the same signal will arrive also to devices that are not the intended receiver, but are listening to the channel (different channel frequencies are handled by LoraLogicalChannel in the receiving method of the physical layer).

The methods that interconnect PHY layers and LoRa channel are Send and Receive methods of the LoraPhy class. When a packet needs to be sent, the sending method of the PHY layer is called with the required parameters: the packet to send and its time-on-air, the transmit power, the spreading factor, and the channel number. The channel schedules a Receive event for all the nodes listening to the channel at a time computed with the aid of the PropagationDelayModel class. This class is provided by ns-3 and computes the propagation delay between two nodes given their MobilityModel (e.g., position) and the model assumed. In this work, the ConstantSpeedPropagationModel simply computes the time of flight between two nodes. The scheduled Receive event also contains information about channel, signal duration, spreading factor and received power. This last value, is computed with an object from the PropagationLossModel class, that computes the power loss based on the transmit power and the location of the nodes. It implements what described in Section 4.1 by concatenating three loss models: LogDistancePropagationLossModel, available by default in ns-3, Correlated-

ShadowingPropagationLossModel and BuildingPenetrationLoss as implemented in [34]. Once computed, the propagation loss for a given ED remains the same throughout the entire simulation since there is no mobility.

A useful feature of ns-3 is that each of these three loss models can be switched off or substituted with another one in a very simple way, making it easy to simulate and compare the performance of the network with different combinations of models. Furthermore, some models, as LogDistancePropagationLossModel and OkumuraHataPropagationLossModel are already implemented in ns-3 and only need configuration.

5.3 Contribution

The purpose of this work was to implement the retransmission feature in the lorawan module. The main steps can be listed as:

- implement the retransmission procedure on the ED's MAC layer;
- activate an $ACK\_TIMEOUT$ random delay;
- implement ADR as suggested in [33];
- modify the NS implementation to correctly behave when retransmissions are involved.

Moreover, the implementation will respect the following assumptions:

1. For each ED, the MType set on the messages will remain the same for the whole simulation. Therefore, if during the network configuration an ED is set to transmit confirmed messages, *all* the packets sent by that device will require an ACK;
2. if an application packet arrives during a retransmission procedure, the retransmission is abandoned and the new application packet is sent. The packet whose retransmission did not end is marked as failed and will never be considered anymore;
3. if a packet arrives at the MAC layer but can not be transmitted immediately, for example because of duty cycle limitations, the transmission is postponed. This happens for any packet transmission, including retransmissions;
4. if the ED is waiting for an ACK and receives a downlink message that does not contain an ACK, the message is processed but the retransmission procedure goes on.

In the following, we will describe these points in a detailed manner, dwelling on the most difficult and delicate passages. Particular attention has been paid to singular cases in which unexpected events happened: in fact, they are the most difficult to model correctly.

5.3.1 Retransmissions

Retransmissions are controlled by the MAC layer, which sends the correct packet (a new application packet or a packet to be transmitted once more) to the PHY layer. In the following, we will refer to the first transmission of an application packet with the term *transmission (tx)*, and we will call *retransmission (retx)* the transmission of a packet that has already been sent at least once.

When developing the code, particular attention must be paid to the possible combination of events and method calls: this analysis clarifies in which sequence functions are called and, therefore, where parameters should be updated to obtain the correct results. Figure 5.2 presents some of the most frequent cases and gives an idea about how the analysis has been carried out. The vertical arrow indicates a packet arrival from the upper layer.

a) Case of an unconfirmed packet arriving when the ED has the possibility of transmitting it immediately (not limited by duty cycle). In this case, the packet sending is performed right away and no other events are scheduled.

b) Case of an unconfirmed packet arriving when the ED has run out of duty cycle. In this case the simulator event corresponding to the call of the sending function with that packet is postponed at the time in which the device will have the ability of transmitting.

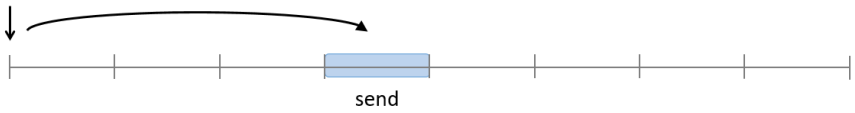
c) Case of a confirmed packet arriving from the application layer when the duty cycle does not limit the transmission. Here, the packet is sent immediately and the two receive windows are scheduled. If a downlink packet is received in one of the two receive windows, the message is decoded and parsed. If an ACK for that device is found, no retransmission are performed, otherwise the sending of the same packet is scheduled as soon as the ED can transmit again. Note that in that case, the same sequence of events is repeated, with the only difference that the packet sent is not the one received from the application layer but the one coming from the previous (unsuccessful) transmission.

d) Case of a confirmed packet delivered from the application layer and sent immediately. If a downlink packet is not received, another sending event is scheduled after having verified that both the receive windows went empty.

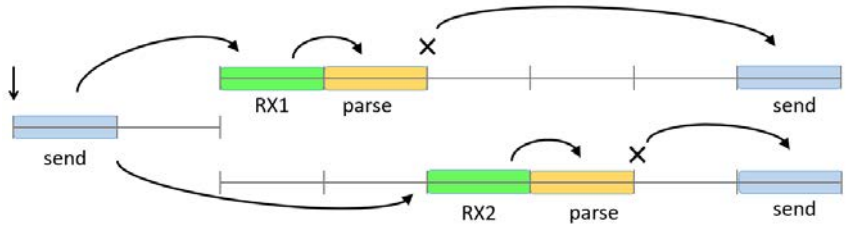
e) Case of an application packet that can not be sent immediately because of duty cycle. Its transmission is postponed but in the meanwhile, another application packet arrives at the MAC layer. In this case, the transmission of the oldest packet is deleted and the newer packet will be transmitted when the duty cycle will allow it.

f) In this case a confirmed application packet is successfully sent by the MAC layer and the two receive windows are opened. However, the ACK is not received and a retransmission is scheduled. When a new application packet arrives (both during the receive windows or while waiting for the retransmission), the scheduled retransmission is deleted and the transmission of the newer packet is scheduled out of duty cycle.

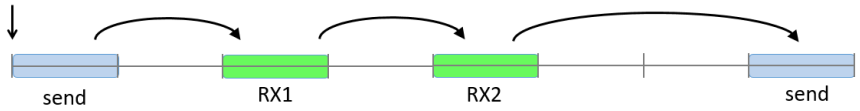
(a) Unconfirmed packet, duty cycle OK.



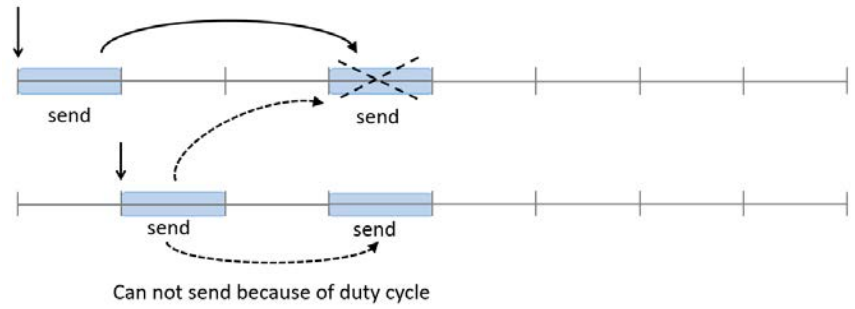
(b) Unconfirmed packet, duty cycle KO.



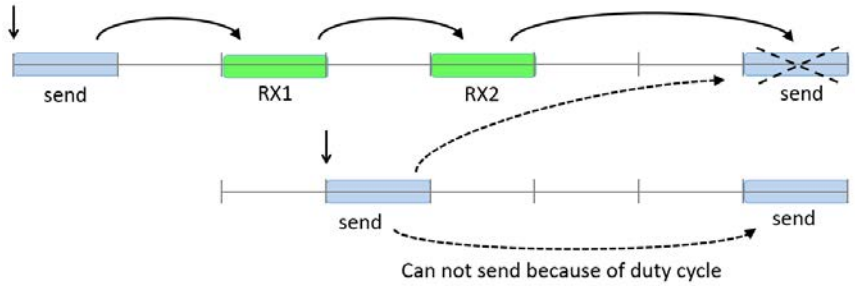
(c) Confirmed packet, duty cycle OK, downlink message received.



(d) Confirmed packet, duty cycle OK, downlink message not received.



(e) Duty cycle KO, new application packet.



(f) Duty cycle OK for the first packet, ACK not received, new application packet.

Figure 5.2: Schedule of events in different situations.

		packet = `LRP-packet`	
		true	***false***
`waitingAck`	***true***	retransmission	transmission of a new confirmed packet
	false	error	transmission of a new unconfirmed packet

Table 5.1: Possible cases of variables `waitingAck` and `LRP-packet` and their meaning.

A data structure called `LoraRetxParams` has been introduced in the `EndDeviceLoraMac` class to keep track of the retransmission parameters. `LoraRetxParams` contains four elements, whose initialization value has been indicated in parenthesis:

- `firstAttempt` (*0 seconds*) the time at which the first physical transmission of the packet took place;
- `packet` (*0*) a pointer to the packet that is going to be transmitted. For the sake of clarity, we will indicate it as `LRP-packet`;
- `waitingAck` (*false*) a boolean variable set to true when confirmed packets are sent and reset to false when a successful ACK is received;
- `retxLeft` (*maxNumbTx*) an integer number indicating the number of times that the packet can be transmitted. Its value can be set at each simulation with the parameter `maxNumbTx`.

The value of these variables is first initialized when a new packet is going to be transmitted and is updated in different moments to indicate the state of the retransmission. Since the same sending function is called both during the retransmissions procedure and from upper layers, it is important to distinguish the two cases, and this is done by checking two variables: `waitingAck` and `LRP-packet`, that will be compared to the packet that is going to be sent. If the variable `waitingAck` is set to true and the packet to be sent is the same of the one saved in the the structure, it is the case of a retransmission. If the packet is different from the one recorded in the structure, the ED is going to send a new application packet. Table 5.1 summarizes the possible cases.

The sending procedure has been split in different methods to simplify it.

Figure 5.3 describes the procedure implementing the first phase of the sending method. When the packet arrives at the `Send` method of the `EndDeviceLoraMac` class, the packet size is checked, to establish if the message can be sent in accordance to the LoRaWAN specification or is too big for the considered data rate, in which case it is discarded. Then, the method `GetNextTransmissionDelay` obtains the first time at which the message can be sent being respectful of duty cycle. If the packet can not be transmitted immediately, the transmission is postponed by an appropriate delay. Otherwise, it is checked if the packet has not run out all its possible

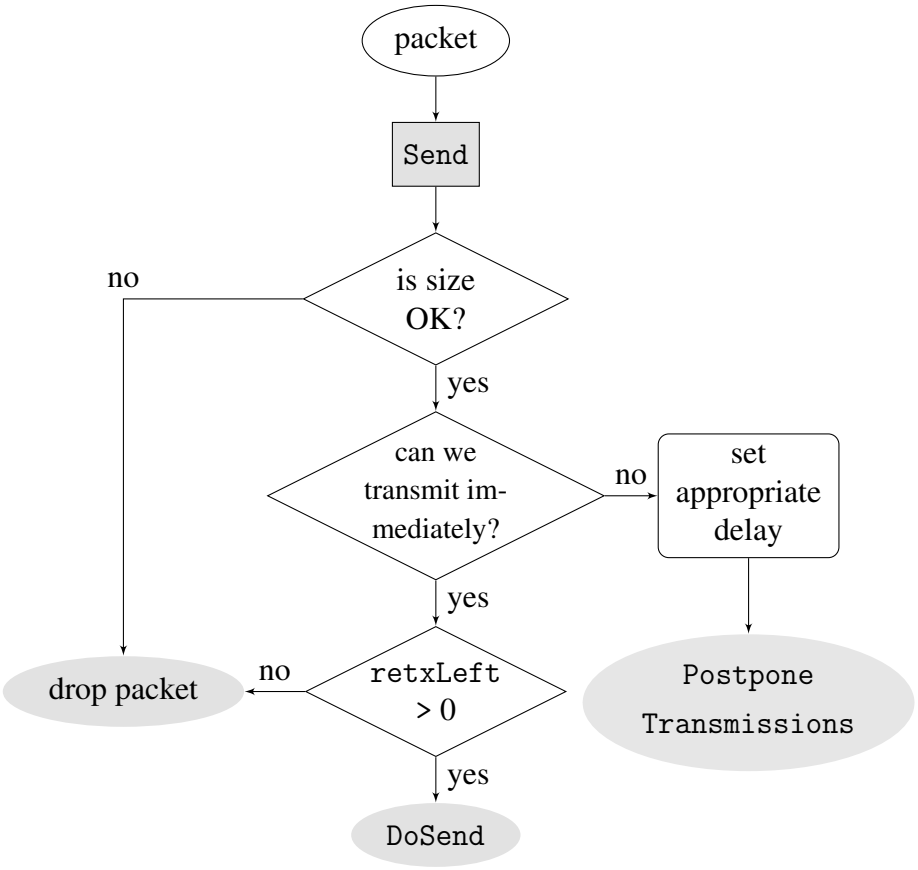


Figure 5.3: Method Send.

transmissions: if the packet is new, retxLeft will be equal to m_numbTx, otherwise it will be lower. This establishes if the packet is dropped or can continue the sending procedure, calling the DoSend method.

If the transmission is delayed, the postponeTransmission function deletes already scheduled transmissions or retransmissions (if present) and schedules a new transmission at the delay received as input parameter with the packet.

The DoSend function, illustrated in Figure 5.4, verifies if the operation that must be performed is a transmission or a retransmission with the aid of LoraRetxParams structure and applying the cases of Table 5.1. If it is a retransmission, the retxLeft number is decremented by one and the packet saved in the LoraRetxParams structure is delivered to the SendToPhy method. Otherwise, since a new application packet is going to be transmitted, it is checked if the variable waitingAck is set to true, meaning that we were in a retransmission procedure (case f of Figure 5.2). In this case, the old packet is dropped and scheduled events are deleted, the LoraRetxParams are reset and then set with the appropriate parameters for the new packet. If the ED was not in a retransmission procedure, the parameters of the new packet are set on the structure. Finally, the SendToPhy method is called. This function will call the sending function of the PHY layer that adds headers, prepare transmission parameters and sends the packet on the channel selected with the method GetChannelForTx. The packet trasmission is also registered to be considered in the duty cycle computation; then, the SendToPhy function prepares the ED for down-

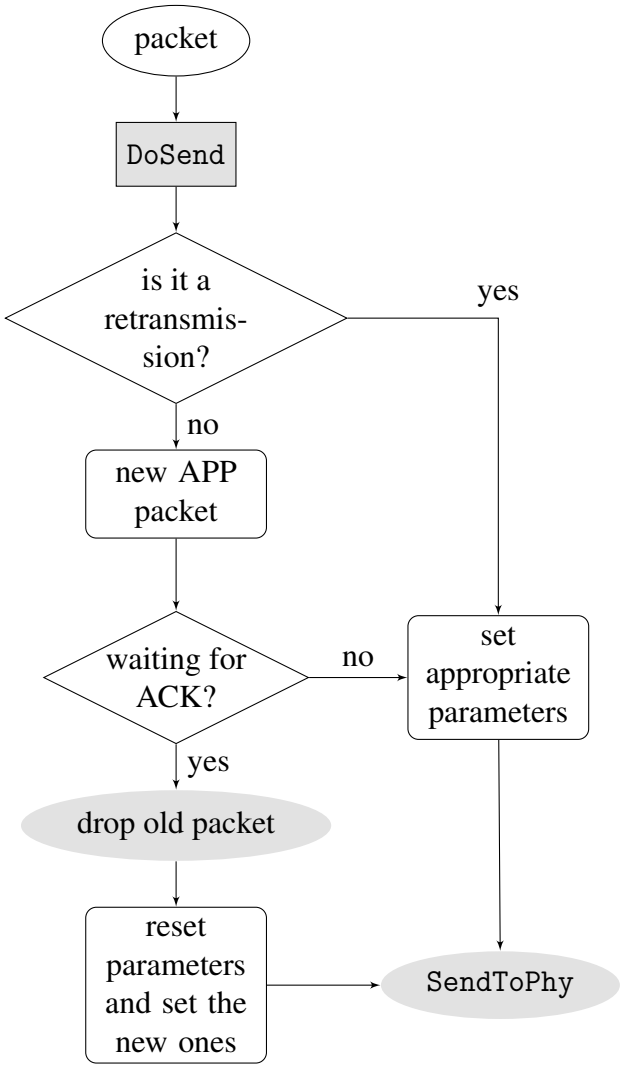


Figure 5.4: `DoSend` method.

link, indicating the frequency and data rate used in the transmission to correctly open RX1 with the same parameters.

When a confirmed packet is sent, RX1 and RX2 opening and closing are also scheduled. The `CloseFirstReceiveWindow` function checks the mode of the device and if it is in `IDLE` mode, it switches the device to `SLEEP` mode, otherwise this is done by the the receiving function at the PHY layer. In addition to this, the `CloseSecondReceiveWindow` function also has to establish if a retransmission is needed, according to the procedure of Algorithm 1. If at that moment the ED is receiving a packet, potentially further retransmissions will be scheduled by the receiving method. If the ED successfully received an ACK before `CloseSecondReceiveWindow` was called, the variable `waitingAck` was set to false during the reception and here the device is prepared for another transmission resetting the data structure. Otherwise, if the ED is still waiting for an ACK but no successful downlink packet has been received, a retransmission is scheduled if possible, or the packet is marked as failed and this failure is notified to the simulation script via a callback; then the

device is prepared for a new transmission by resetting `LoraRetxParams`.

Algorithm 1: `CloseSecondReceiveWindow` method.

```
Result: Close RX2 and schedule retransmissions.
check ED's mode;
if RX then
    Receive will handle the result.
    Return.
else
    No reception: switch to SLEEP if not already in this mode;
    if waitingAck then
        if retxLeft > 0 then
            retransmission: Send LRP-packet;
        else if retxLeft = 0 & ED is not receiving then
            packet has been unsuccessful;
            mark packet as failed and prepare for new transmissions;
        else
            error;
    else
        reset LoraRetxParams.
```

The MAC layer's `Receive` function is responsible for decoding downlink packets arriving at the device and deciding if a successful acknowledgment was received. Figure 5.5 illustrates the procedure adopted by this function. As LoRaWAN does not distinguish packet preambles between uplink and downlink directions, it can happen that EDs erroneously lock on messages sent from other nodes to the GW. Therefore, as first operation it is checked if the message detected by the ED was in the uplink direction. If this is the case, if possible, another retransmission is scheduled, otherwise the packet is dropped. On the other hand, when the packet is in the downlink direction and the ED is the intended receiver, RX2 is deleted if still scheduled, and the `ParseCommand` function is called to analyze the packet content.

When the device is waiting for an ACK, the `ParseCommands` function checks if the ACK flag is set in the received packet and, if so, notifies a success, reset `LoraRetx- Params` and deletes already scheduled retransmissions if present.

Finally some minor changes have been made to the `Receive` method of the NS. In this function, when the NS receives a packet from an ED for the first time, it creates a reply and sends it in the first available slot (RX1 or RX2). The function has been modified to send the already created packet to the ED even if it was not the first time that the node contacted the network server, for example when multiple transmissions - or retransmissions - take place because of an ACK loss. Moreover, the parameters used for the transmission of the downlink packet need to be updated with the ones of the last uplink transmission.

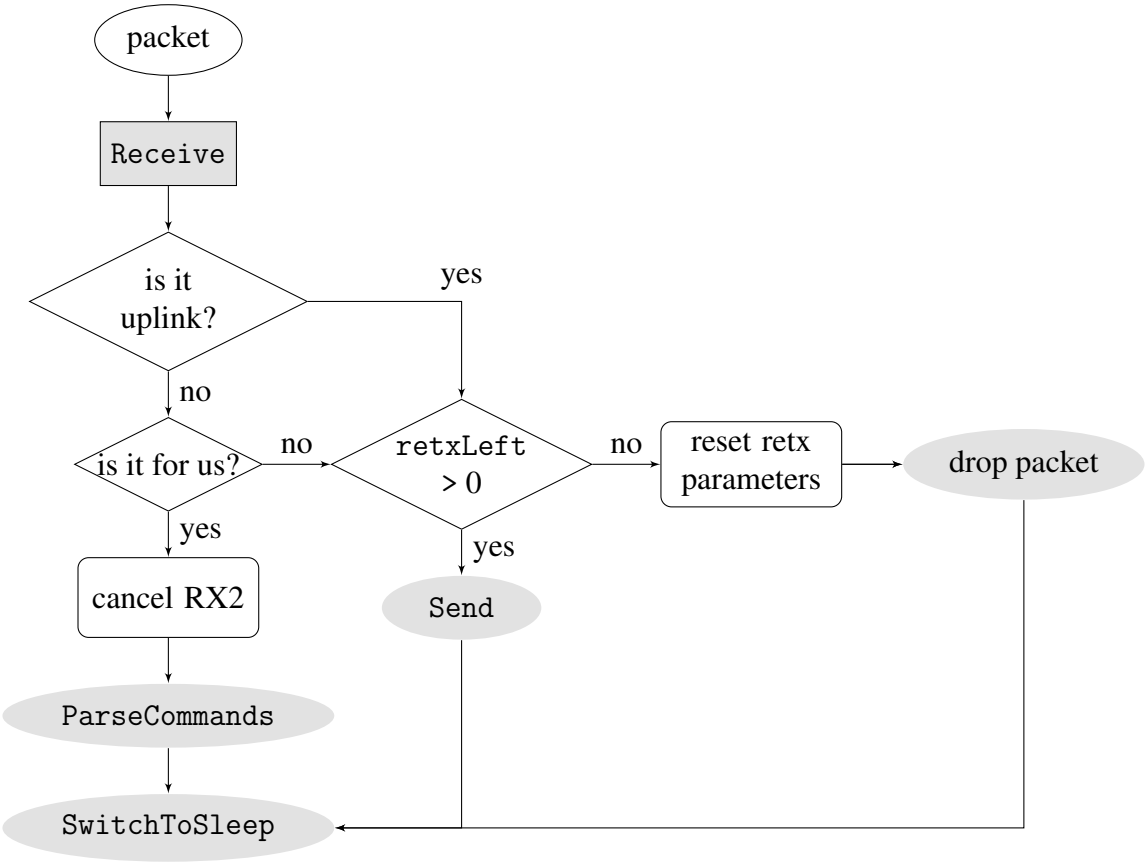


Figure 5.5: Method `Receive`.

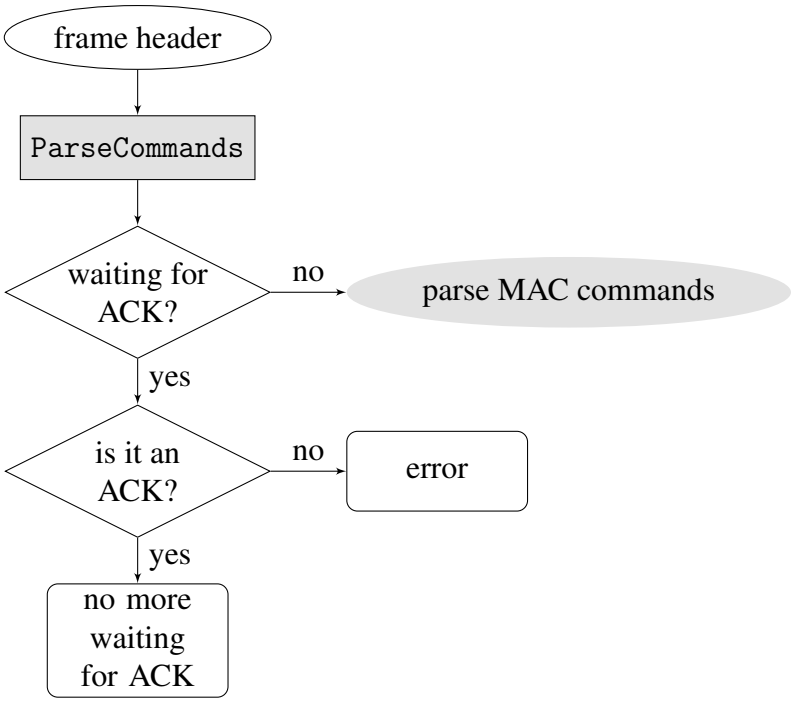


Figure 5.6: Method `ParseCommands`.

5.3.2 Random delay and ADR

To avoid consecutive collisions when retransmissions are employed, the LoRaWAN specification [5] suggests to add a random delay, called $ACK\_TIMEOUT$ to the time at which the retransmission is performed. This has been implemented in the `Send` function of the `EndDeviceLoraMac` class. When the `GetNextTxDelay` function is called to know if the packet can be sent immediately or if it must wait for the silent period imposed by the duty cycle to finish, it is also checked if the ED is performing a retransmission. If that is the case, a random time chosen uniformly between 0 and 3 seconds is added to the delay at which the transmission is postponed.

The ADR implementation is based on the one described in [33], where it is suggested to decrement by one the data rate each time two consequent retransmissions fail, and to maintain that data rate for the following transmissions of the ED. This is applied in the `SendToPhy` function: if all the following conditions hold, the data rate is decremented by one.

- Data Rate Adaptation is enabled;
- The data rate is larger than the minimum one (SF $<$ 12);
- The device has not already performed the maximum number of transmissions;
- The number of retransmissions left is a multiple of 2.

Note that, in the following, any reference to ADR will always consider this specific implementation.

5.4 Simulations

Classes in ns-3 implement specific network models and protocol; on the other hand, to describe the network topology and models to be used in a ns-3 simulation, a C++ or Python program is needed. Its structure is based on the following points:

1. **Creation of the topology** Gateways and EDs are created in the desired number as collections of *node* objects. The network server is created too and a `MobilityModel` is associated to each node to establish its position in the simulated space.

2. **Models** The LoRaWAN protocol stack is installed on each node, specifying the device type. This is done with the aid of *helpers*, classes specialized in installing the various objects implementing the needed layer of the ISO/OSI stack in multiple nodes.

3. **Configuration** The models of the protocol are configured to use certain values, for example the message type, the maximum number of transmissions allowed and whether or not to use ADR. The LoRa channel is created with the desired propagation loss model, presence of shadowing and buildings, together with the point-to-point links connecting NS and GWs.

4. **Execution** The simulation starts executing events and the corresponding function calls. During the simulation, trace sources fire and save data in appropriate data structures. The simulation ends when all the scheduled events have been executed or when a stop event, scheduled before starting the simulation, is met.

5. **Performance analysis** The data saved thanks to trace sources are finally analyzed and visualized.

The procedure used in the script is described in a more detailed manner in Procedure 2. When analyzing the network performances for confirmed traffic whith a maximum number of transmissions attempts greater then zero, it is important to consider the network at operating speed. Indeed, if it is not the case, devices transmitting first will not suffer from any (re)transmission already underway.

Let's take as unit period the maximum period of the applications running on the EDs in the network in a given simulation. We need to simulate multiple periods and discard the first ones. Also, the application packets sent in the final periods and requiring multiple retransmission may never end successfully, as the retransmission procedure may continue beyond the moment at which the simulation is stopped. Therefore, for our performance analysis we will only consider the central periods, ignoring the *transient periods* at the beginning and at the end of the simulation. In particular, we will look at the packets sent inside the central period, how many times they have been transmitted and if they finally succeed (even if the success happens outside the central period).

In the simulations, EDs are preconfigured to be able to communicate in the network, so that there is no need to perform any join procedure. EDs are 1.2 m high, while the GW is located at a height of 15 m.

Finally, an additional script has been written to perform multiple simulation runs for each network configuration, and get averaged results. For example, to analyze the performance of the network with a given application period at the EDs and a fixed number of nodes, different runs allow to change the nodes' position, spreading factor and transmission time, leading to different results in collisions and delays that need to be averaged to get results that are representative of the network performance.

Procedure 2: Simulation setup script

Input: r = simulation radius;
n number of EDs;
application period;
`maxNumbTx`;
ADR enable;
scenario (Urban or open air);
number of periods to simulate and transient periods

1 Create the `LoraChannel` object
2 Configure the channel according to the desired scenario and delay model
3 Create n nodes representing EDs
4 Set EDs' position chosen uniformly inside a circle of radius r
5 Install LoRa stack to each node
6 Connect EDs to the channel
7 Connect the callbacks of the ED trace sources to local functions
8 Create the GW at the position (0, 0)
9 Install LoRa stack on the GW
10 Connect the callbacks of the GW trace sources to local functions
11 if the input scenario is "urban", create buildings
12 Set spreading factors up
13 Create NS and connect it to EDs and GW
14 Install and start applications on all network devices
15 Start the simulation
16 Save simulation results

Chapter 6

Results

This chapter discusses the results of the simulations and the analysis of the LoRaWAN network via an analytical model. Before illustrating the final outcomes, we present the metrics in which we were interested and the variables that were considered and played a role in the simualtions.

To analyze the performance of the whole network, we consider aggregate performance, without analyzing the outcomes of each specific device. We already know that the network will be penalized by devices far from the central GW, as they use lower data rate and are more exposed to collisions. Nevertheless, they are taken into account as being part of the network, since pruning these devices (as done in [34]) would artificially increase the network performance introducing even more unfairness.

In the following, we will give the definitions used to analyze the simulation outcomes. For the discussion about the model, the assumptions taken into account are described in section 4.3.

6.1 Simulation metrics and variables

The metrics of interest used to analyze the network performance are the following:

Spreading factor distribution Different scenarios modify the distribution of the spreading factors. The assignment of the SF is based on the power received at the GW: the minimum spreading factor is selected so that the ED can still reach the GW by using the SF that minimizes the packet transmission time.

Success probability We define success probability (P_{succ}) at the MAC layer the probability that a data packet sent by an ED is successfully received at the GW *and* that the corresponding ACK is received by the node. It is computed as:

$$P_{succ} = \frac{\sum_{i=1}^{\texttt{maxNumbTx}} \text{number of successful packets that used } i \text{ transmissions}}{\text{total MAC sent packets}} \tag{6.1}$$

A packet is successfully received at the GW if it does not collide with any other signal or if it collides but is not destroyed by interference. Moreover, a

receive path must be available to lock on the incoming signal and the packet must not be dropped because of an ACK arrival at the GW. When only uplink is considered, the success probability does not consider ACKs collisions, and is given by:

$$P_{succ-uplink} = \frac{\text{number of successfully received packets at the GW}}{\text{total sent MAC packets}} \tag{6.2}$$

Packet outcome From the GW's point of view, the outcome of each PHY packet is recorded. The outcome can be

- *received*, if the packet is successfully decoded;
- *interfered*, when the packet was destroyd by interference and could not be correctly retrieved by the GW;
- *noMoreReceivers*, when the packet but could not be received by the GW because no reception paths were available on its channel;
- *underSensitivity*, when the packet arrived at the GW but its power was below the GW's sensitivity at that spreading factor. In most of the simulations, except when explicitly stated, the maximum distance at which the EDs could be positioned was set so that no devices were out of range and, therefore, this value will almost always be zero.

Delays The *average delay* is the average time between the first packet transmission at the MAC layer, and its successfully reception at the GW. The *average ACK delay* considers the delay between the first transmission of a packet at the MAC layer until the moment in which the corresponding ACK is received by the ED. Both are averaged over the total number of MAC packets sent in the central period (ignoring the transients) and successfully received.

Note that for the computation of all the above metrics, only the central period of the simulations has been considered, ignoring the transient periods.

To study the evolution of these network metrics, a set of variable parameters, configurations and scenarios have been considered. Here, we list all the ones that can be taken into account, while in the results section it will be indicated which combination of parameters produced the considered outcome.

- Scenario: different scenarios have been considered: open air environments, where only path loss influence the signal propagation, and urban scenarios, where shadowing effect and buildings' presence have been used. The path loss is usually computed with a log distance propagation model; the Okumura-Hata propagation model is only applied in the simulator when comparison with the mathematical model are made;

- Network scale: we will always analyze a single LoRaWAN cell, a network in which a unique GW is employed and the number of EDs can be changed, leading to a higher probability of interference between two signals for higher device densities;

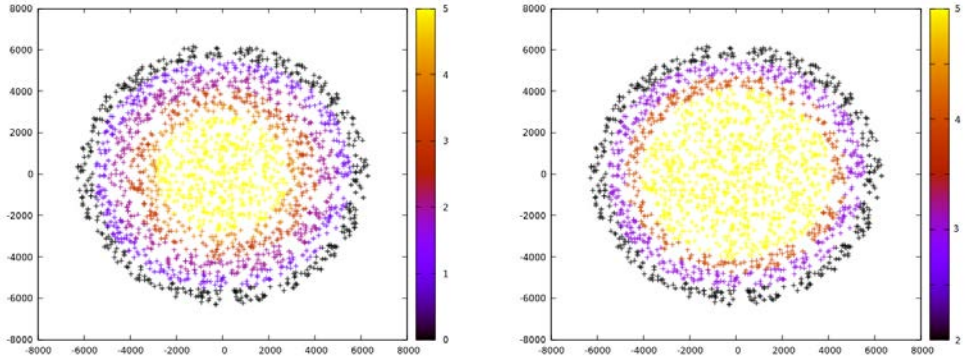
- Model of the traffic being generated at the application layer, determined by the application period (appPeriod). For a fixed number of EDs, a smaller appPeriod will increment the offered traffic in the network and therefore the collision probability;
- Radius of the circle around the GW in which EDs are placed;
- Uplink and downlink traffic: the type of messages sent by EDs influences the offered traffic in the network;
- Maximum number of transmissions (`maxNumbTx`) allowed, that determines the QoS offered by the device but also influences the amount of traffic flowing in the network;
- Activation of ADR in the ED, influencing the data rate employed by EDs;
- $ACK\_TIMEOUT$ random delay between retransmissions, affecting the delay of packet delivery but also the probability of collision when retransmitting;
- Payload length of the downlink messages, that has an impact on the duration of the packet transmission.

6.2 Simulations results and discussion

6.2.1 Spreading Factors distribution

As in LoRa SFs determine the employed data rate, to analyze the distribution of SFs we can observe the data rate in use by EDs. Figure 6.1 shows the distribution of the data rates of EDs distributed around a GW positioned at coordinates (0, 0), when only path loss is considered. The data rate is encoded as a color: higher data rate, corresponding to smaller spreading factors, are shown in the picture with lighter colors, while the darker points represent nodes using higher spreading factors (data rates DR0 and DR1). Each node is assigned the minimum SF that makes it possible to connect it to the GW, resulting in a received power larger than the sensitivity. Note that, as only path loss is considered, the received power decreases with the distance, and this implies the need to use lower data rates when the ED is further away from the GW.

The distribution of spreading factors depends on which sensitivity is used as reference. In a network with only uplink traffic, a connection from EDs to GWs is sufficient, and the spreading factor can be determined from the GWs sensitivity. On the other hand, when the network needs to carry also downlink traffic, the EDs sensitivity should be used, being is lower. In fact, as the sensitivity capabilities are different for GWs and EDs, also the ranges achieved with a given spreading factor are different: the maximum distance that allows the signal to be decoded is about 8850 m when the GW's sensitivity is used, while it is reduced to about 6315 m when using ED's sensitivity. When the ED's sensitivity was used to set the spreading factors, uplink transmissions were successful, while downlink packets were not received by the EDs because arriving under ED's sensitivity, even if the received power was the same.



(a) Distribution considering EDs' sensitivity

(b) Distribution considering GWs' sensitivity.

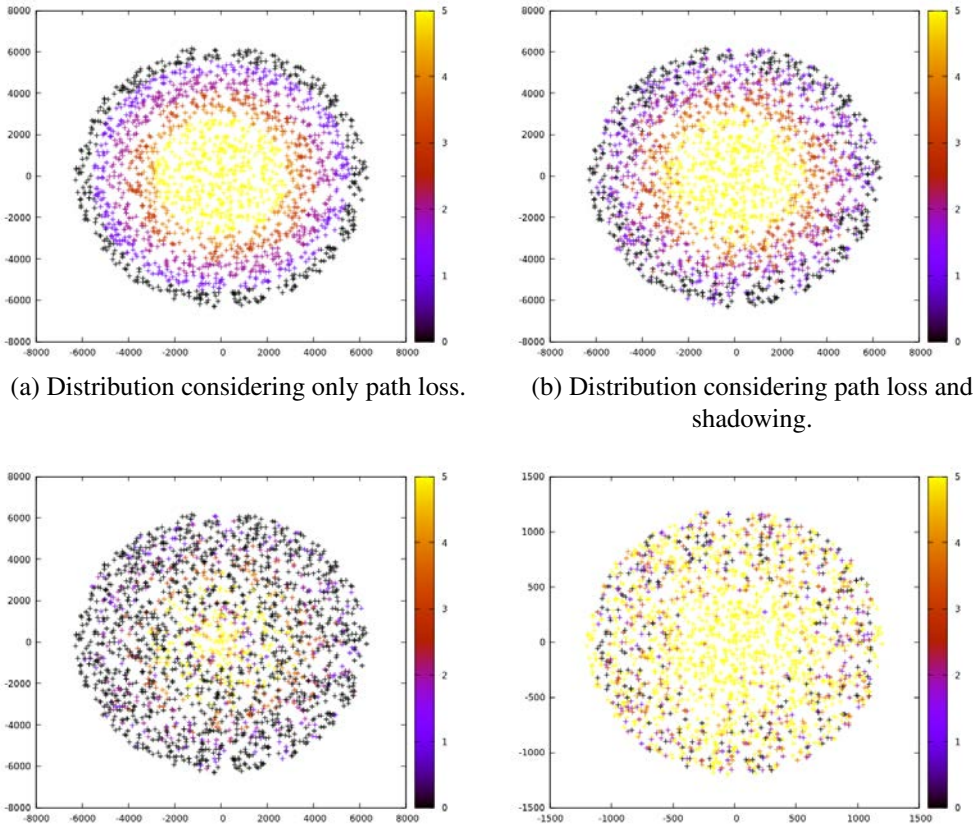
Figure 6.1: Distribution of the spreading factors with different sensitivities taken as reference.

Figure 6.2 shows the assigned data rates when also shadowing and buildings are considered in the computation of the received power. When only path loss is considered, the distribution of the SFs form different concentric annuli around the central GW. When shadowing is added to path loss in the propagation model the borders between the sets of devices that use the same SF become blurrier: the shadowing makes some devices experience a slightly worse channel, and this difference is sufficient to make these nodes transmit with an higher SF. Analogously, other EDs could experience a better channel and select a lower SF. When also buildings are introduced, the SF distribution completely changes. In fact, the high losses caused by walls force a large number of devices to decrease the necessary data rate to achieve connectivity. For the same reason, when considering the same area then the previous cases, many devices become unreachable. Therefore, the performance analysis in this case is done by placing the EDs closer to the GW, as shown in Figure 6.2d. Here, most of the devices can employ lower spreading factors because they are positioned near the GW or outdoors, while some devices, placed inside the buildings, need to employ high SFs.

6.2.2 Unconfirmed traffic

The first simulation campaign considered only traffic transmitted in the uplink direction, not requiring a confirmation from the network server. Figure 6.3 analyzes network scalability showing the probability that a packet is successfully received at the GW when the number of EDs is increased and different application periods are considered. We see that P_{succ} decreases as the network is more crowded for any application period, but the effect is more significant for small periods, as the offered traffic is higher: for example, with 5000 EDs, an appPeriod equal to 5 minutes corresponds to an offered traffic $\lambda = 16.67$ packet/s, while an appPeriod of 1 day results in about $\lambda = 0.06$ packet/s.

Figure 6.4 shows the causes of packets losses when different appPeriod are considered. We see that when short application periods are considered, the most frequent



(a) Distribution considering only path loss.

(b) Distribution considering path loss and shadowing.

(c) Distribution considering path loss, shadowing and presence of buildings.

(d) Distribution considering path loss, shadowing and presence of buildings, in a shorter radius.

Figure 6.2: Distribution of the spreading factors in different environments.

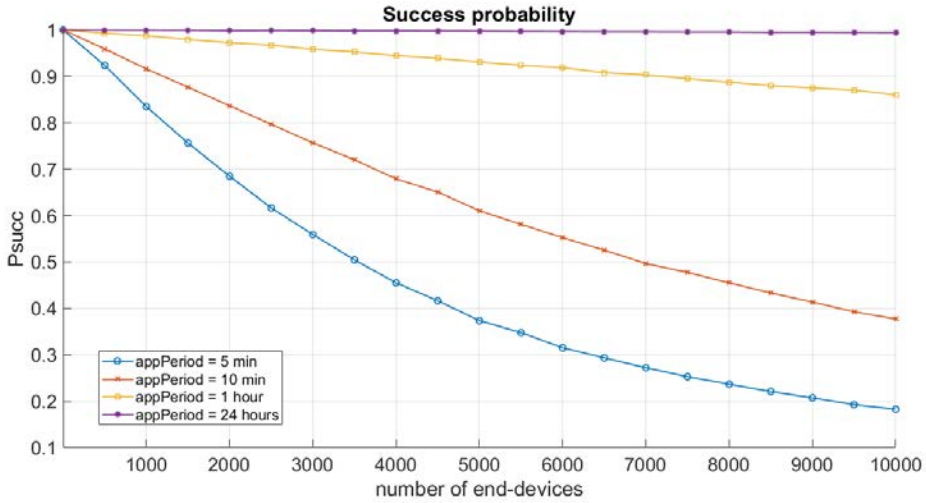


Figure 6.3: Network scalability for unconfirmed data traffic with different application periods.

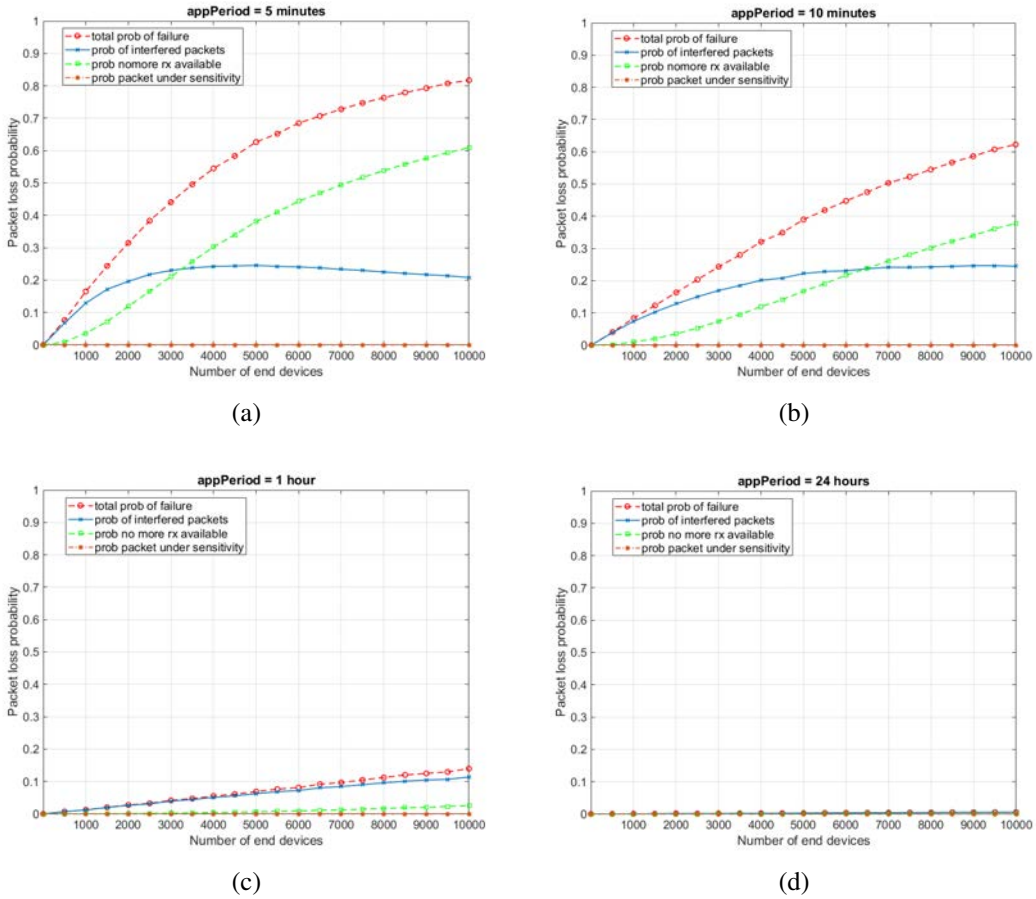


Figure 6.4: Packet loss probabilities for unconfirmed data traffic with different application periods.

source of packet losses is the lack of available receive path at the GW. This is caused by the fact that, in order for a packet to be marked as lost due to interference, it must first find an available path, and it is a reasonable assumption as if a packet survived interference but no receive path were available, it would have been lost. In particular, if we compare Figure 6.4a and Figure 6.4b, the number of devices for which the receive paths at the GW are saturated is higher when the application period is larger, as packets arrive less frequently and the GW has enough time to complete packet receptions. When the application period is larger, as in Figure 6.4c, almost the totality of packet losses are caused by interference, independently from the network size.

6.2.3 Confirmed traffic

The purpose of the second experimental campaign was to observe the behavior of the network when confirmed traffic was employed for different combinations of the variables described in section 6.1.

The first result is presented in Figure 6.5, where it is shown how the success probability at MAC layer depends on the number of EDs and on the application period. Here, a maximum number of transmissions equal to 8, no ADR procedure and zero-

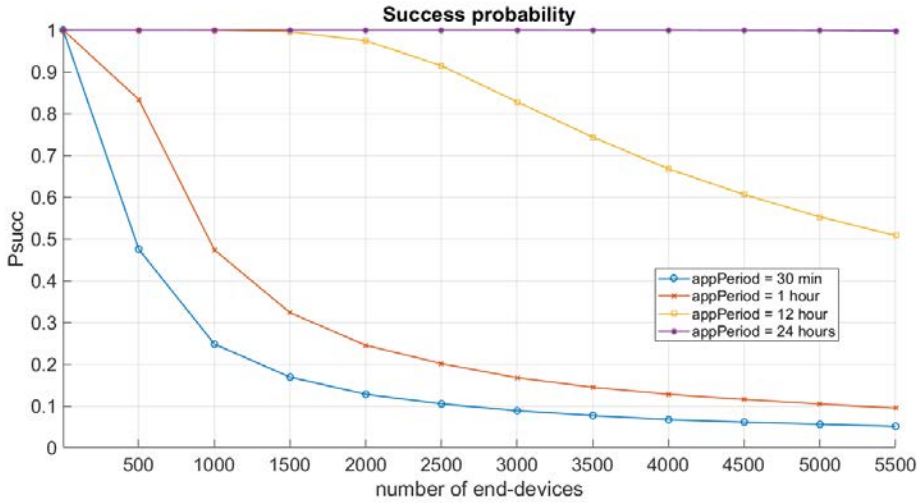
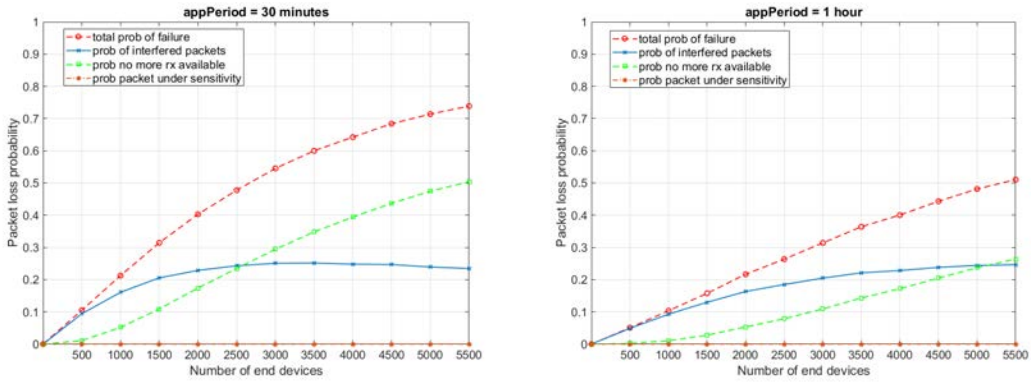


Figure 6.5: Network scalability for confirmed data traffic with different application periods.

payload ACKs were employed. Comparing the network performance with that of Figure 6.3, we see that the success probability sharply decreases when more devices are used and this effect is more pronounced when messages are sent more frequently. For example, if we consider an application period of 1 hour, the packet success probability is over 0.8 in the network with only uplink traffic for any number of devices, while when all the EDs require confirmed traffic, 1000 nodes are sufficient to make it fall under 0.5. This could seem counter-intuitive, as one could expect that retransmitting lost messages should increase the reliability of the communication. However, from Figure 6.6b we see that most of the packets were not received by the GW. In fact, with confirmed communication the traffic is increased both by ACK messages sent in RX1 and by retransmissions, and this causes an avalanche effect, as a packet loss causes retransmissions that lead to an increase in interference. Moreover, while for unconfirmed traffic unsuccessful message delivery could only be caused by a data packet loss, when confirmed traffic is employed failures can also be caused by the loss of an ACK. Finally, it can be seen that as long as the packets are sent occasionally the success probability reaches high values because interference is avoided.

Figure 6.7 shows a detail of success probability for confirmed and unconfirmed traffic when an application period of 12 hours is used. We can observe that when a small number of devices join the network, the choice of confirmed traffic yields slightly better performance, but this gain is soon lost when the number of nodes is increased.

Figure 6.8 analyzes the outcomes of PHY packets at the GW to see which are the causes of the packet loss when a different number of transmissions is allowed and ADR is adopted. Each graph, shows in blue the probability of a received packet, in red the probability that the packet was lost because of interference and in yellow the probability that the packet was dropped because of lack of available receive paths at the GW. The columns represent, from left to right, the results obtained increasing `maxNumbTx` and, on the right, the results with ADR and 4 and 8 maximum trans-



(a) (b)

Figure 6.6: Packet loss probability for confirmed data traffic with different application periods.

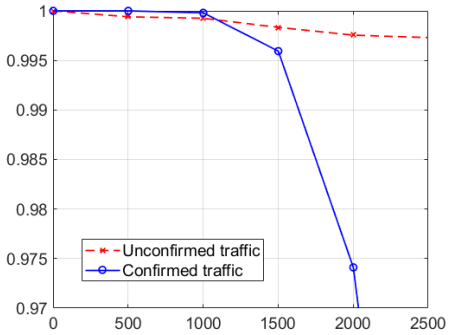
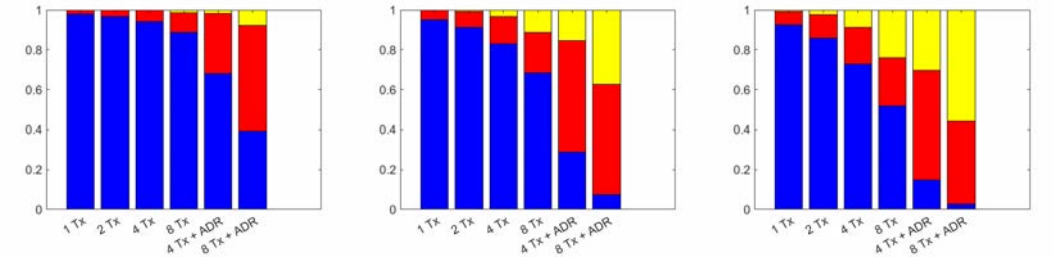


Figure 6.7: Comparison of network performance for unconfirmed and confirmed traffic with application period of 12 hours.



(a) 1000 EDs. (b) 3000 EDs. (c) 5000 EDs.

Figure 6.8: Packet outcomes for different `maxNumbTx` and ADR activation. Blue = received; red = interfered; yellow = noMoreReceivers.

missions respectively. We see that for the same configuration of `maxNumbTx` and ADR, networks hosting a lower number of devices perform better. The distribution of packet transmission outcomes has the same shape for all the considered scale: the maximum number of received packets is reached when a unique confirmed transmission is performed. The bigger the number of retransmissions are allowed, the more packets are lost because of interference.

Furthermore, we can observe that the deployment of ADR as suggested in [33] does not bring any advantage, but only increases the packet loss because of lack of available receive paths at the GW. In fact, it can be observed that this procedure easily degrades the network performance: if an ED does not receive an acknowledgment, after a few retransmissions it will transmit at very low data rates and, since the packet transmission will take longer, the collision probability will increase. Moreover, in a large network, higher traffic is generated and the collision probability increases even more: more and more devices will transmit with high spreading factors, causing an avalanche effect. Since after a few collisions the ED will transmit at the lowest data rate for the remaining simulation time, most of the devices will be transmitting at SF 12 and the network will lose the advantage of orthogonality between different SFs.

Figures 6.9 shows the impact of the size of downlink message in the network performance. In our discussion, ACKs have been considered to carry no payload to minimize their transmission time. In the graph, the network performance for downlink payload sizes of 0 and 10 bytes is shown. As expected, when downlink messages are larger, the network performance decreases, as the longer transmission times lead to a higher collision probability and "consume" duty cycle at the GW, decreasing the frequency at which it can transmit ACKs.

We also analyzed the impact of the random delay ($ACK\_TIMEOUT$) in the retransmission procedure. When a packet collision happens, the two devices will schedule a retransmission at a time imposed by duty cycle restriction. However, as collisions happen with higher probability between packets coded with the same SF it is likely that EDs will choose the same instant for retransmission, resulting in a further collision. The use of $ACK\_TIMEOUT$ make it possible to alleviate this problem. We noted that the interval in which the delay can be chosen, [1, 3] s, is small compared to the on-air-duration of the packet, that can go from about 0.04 s to 2.31 s depending on the packet length and SF: therefore, in the worst conditions of maximum packet size and lowest data rate, consecutive collisions are not always avoided. This observation was confirmed by the results of Figure 6.10, where we see that the network performance does not change when the $ACK\_TIMEOUT$ delay is used.

Finally, Figure 6.11, the blue line shows the average delay at the PHY layer between the first transmission of a packet till the moment in which the packet is received at the GW (only packet that are successfully received are included in the computation). The results of the picture are obtained using confirmed traffic with 8 allowed transmissions, no ADR and an application period of 1 hour. We see that the delay increases with the number of nodes in the network, as the success probability decreases and more transmissions are needed to successfully deliver a packet. We

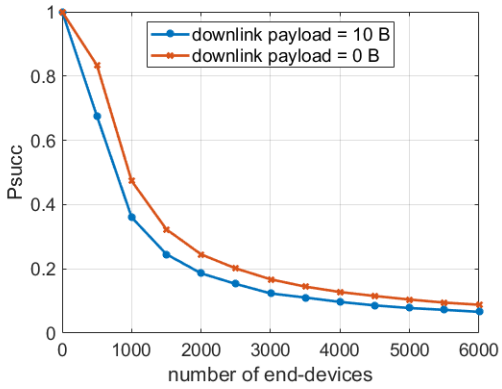


Figure 6.9: Impact of downlink messages' payload on network performance.

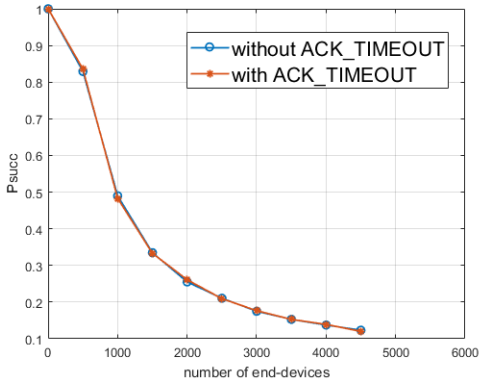


Figure 6.10: Impact of *ACK_TIMEOUT* on network performance.

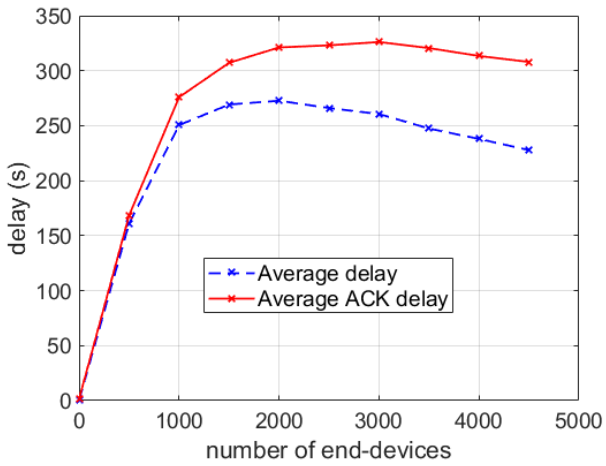


Figure 6.11: Average delay and average ACK delay for different network scales.

also see that the delay reaches a maximum around 2000 EDs: it is likely that this is due to the fact that after this number of devices the amount of interfered packets increases, they are not received at the GW and are not taken into account in the delay computation.

The average ACK delay, indicated with the red curve, accounts for the performance at MAC layer: it is the delay between the first transmission of the packet till the moment in which the corresponding ACK is received by the ED. Since it also considers the time for the ACK reception, its value is higher than the average delay. It reaches its maximum around 2000 EDs as also the success probability for the same scenario presents a less sharp decrease for this value (see Figure 6.5). A nearly constant success probability translates into the fact that the network has reached its maximum load, and thus a constant number of successfully received packets and used in the delay computation.

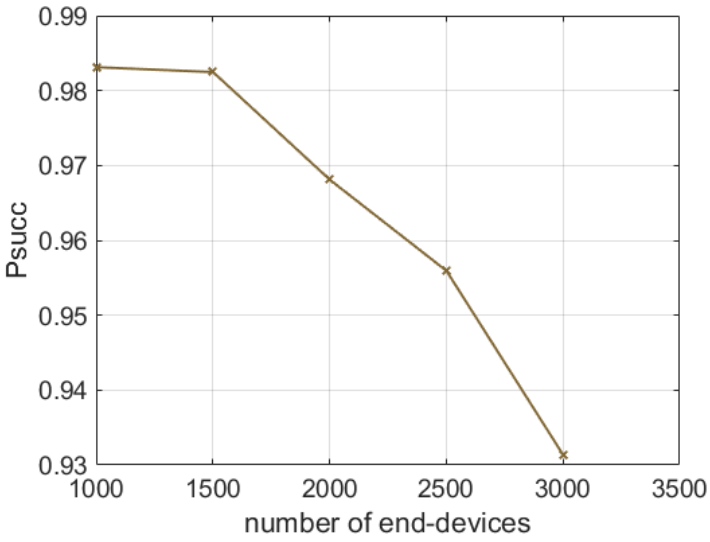


Figure 6.12: Network performance in realistic scenario.

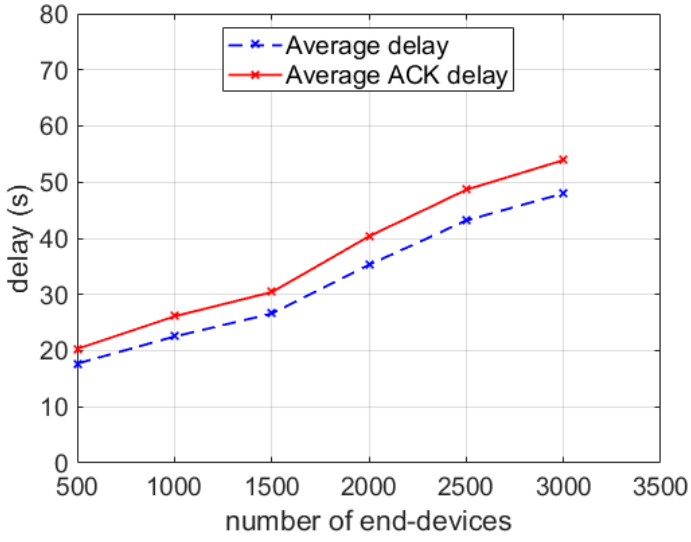


Figure 6.13: Average delays in realistic scenario.

Variable	Value
radius	1000 m
Application period	mixed (as defined in [51])
Type of traffic	30% confirmed, 70% unconfirmed
Packet size	random (Pareto distribution)
`maxNumbTx`	8
ADR	disabled
Propagation model	Log Distance
Shadowing	enabled
Buildings	enabled

Table 6.1: Parameters configuration for realistic simulation.

6.2.4 Realistic network scenario

A realistic scenario has been simulated with the configuration described in Table 6.1. The radius has been decreased with respect ot the previous simulations because, as observed in Section 6.2.1, when shadowing and buildings are introduced, it is more difficult for the devices to gain connectivity to the GW, and many EDs result out of range. For this reason, the value of the radius has been set to 1000 m, which gives about 5% of unreachable devices. The packet payload size is obtained with a Pareto distribution with a minimum value equal to 10 bytes and shape parameter equal to 2.5.

Figure 6.12 shows the success probability for the considered scenario. Note that the packets transmitted by out of range nodes were however considered for the success probability. Since these packets were not received by the GW, they cause the P_{succ} to be smaller than 1 even with small number of EDs in the network. The performance decreases with a higher number of EDs joining the network, but it is generally much better than the network performance when all the devices employed confirmed traffic for communication. Figure 6.13 shows that in the realistic scenario the delays are always less then 60 ms and, even if they increase with the number of devices, no evident increase due to ACK deployment. Note that the different slope starting at 1500 EDs is due to the decrease in the success probability that can be observed in Figure 6.12 for the same number of devices, causing additional retransmissions.

6.3 Model validity

The validity of the model has been tested in an open air scenario using Okumura-Hata model to describe propagation losses. In Figure 6.14 we see that the SF distribution is similar to that obtained employing only Log-Distance propagation model, but the maximum achieved range is now 19200 m. Therefore, in the simulations, the radius of the circle in which EDs can be located is set to that value.

Figure 6.15 compares the results of the models and the one obtained with the simulator. The model proposed in [49] is represented with the black curve, indicated as “Original model”, while the model considering the improvements that we

presented in Section 4.3 is represented by the red curve. The simulations employed Okumura-Hata path loss model, confirmed data with one transmission attempt, and an application period large enough to avoid duty cycle restriction at the EDs. The change in the network load was obtained by taking $N = 4000$ EDs and varying the application period. We see that even if our model does not perfectly catch all the effects of the simulator, its performance prediction are closer to the simulations that the ones of the model proposed in [49]. In fact, the model found in literature overestimates the real performance by not taking into account the receive path at the GW and the duty cycle constraints that limit the ACK transmission frequency.

Studying the correct formulation of the model we can formulate the following observations:

- When a collision between packets happens, two things should be considered:

  1. The model assumes perfect orthogonality, while the simulator was developed to consider the fact the ortoghonality is not perfect: the performance of the model should be better than the one of the simulator;
  2. If the collision overlap is small, the simulator considers two packets colliding with the same SF and similar power as successful, because it bases the interference computation on the energy of the packets during the overlap time. Therefore, the model performance should be worse then the ones of the simulator.

  We ran simulations using two channel models, one considering imperfect SF orthogonality and interference computation based on the overlap, as described in Chapter 4, and one that considered SFs perfectly orthogonal but for which a minimum overlap meant a sure collision. We observed that the simulator outcome in the two cases was the same and, thus, the two effects compensate each other.

- We removed the constraints at the GW for which, at the moment of sending an ACK, all the underway receptions were dropped. We have seen (Figure 6.16) that this behavior has a limited effect on the network performance.

- When λ increases, the network perfomance is strongly limited by the duty cycle at the GW before than collisions.

- We considered a situation in which the choice of the receive window in which the ACK is transmitted is switched: the GW transmit in RX2 as long as it can, then when this choice is not possible, it selects RX1. This is motivated by the fact that when ACK message is sent in RX2, the ED always receives it correctly, as the channel is only used for downlink transmission. Moreover, the fact that the GW does not to add interference in the uplink channels is relevant. Results of this enhancement are showed in Figure 6.17. There is a little improvement for small λ, the performance mostly depend on the GW duty cycle.

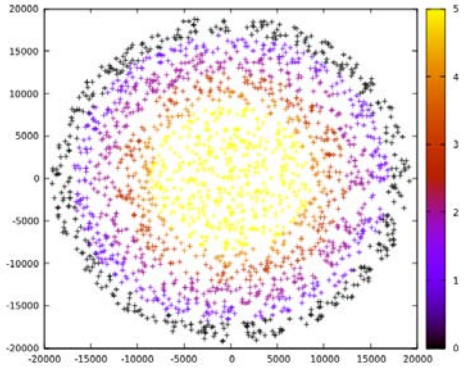


Figure 6.14: SF distribution with only Okumura-Hata propagation model

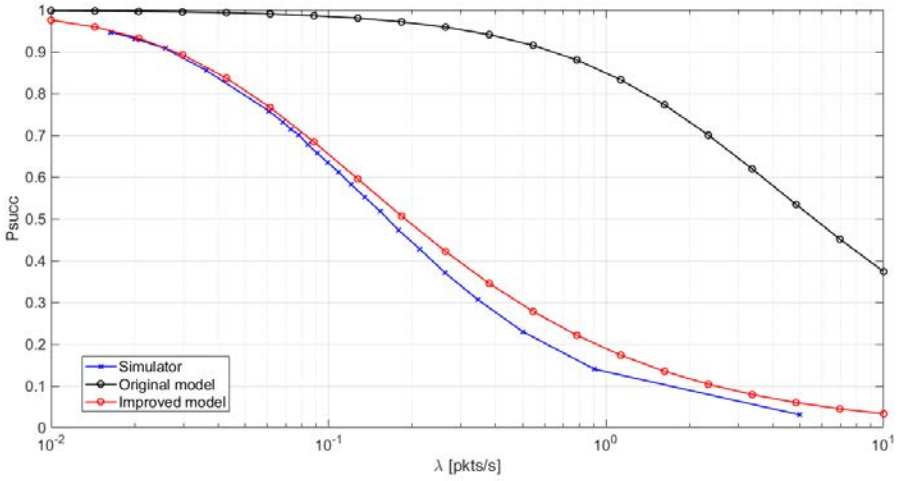


Figure 6.15: Comparison between simulation and model results.

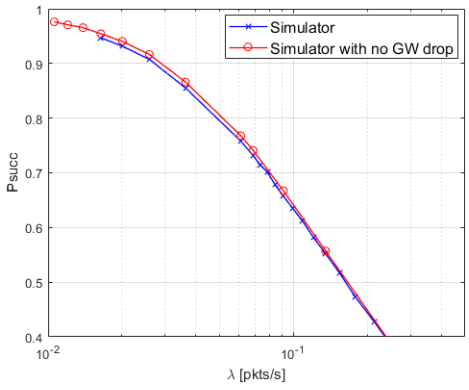


Figure 6.16: Simulator with simultaneous reception and transmission capabilities at the GW.

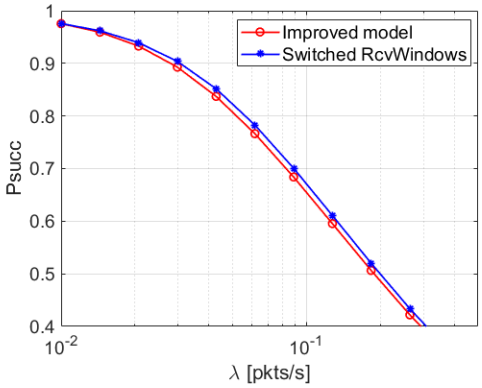


Figure 6.17: Simulator with switched receive windows.

Chapter 7

Conclusion and future work

The purpose of this thesis was to analyze the performance of a LoRaWAN network employing confirmed traffic to increase communication reliability.

After a brief introduction on the advantages and requirements of the Internet of Things and an overview on the most popular solutions, the LoRaWAN technology was introduced. The discussion covered the LoRa physical modulation and the LoRaWAN specification that standardizes the MAC layer features, network topology, classes of devices and retransmission procedure, together with regional parameters. Then, a review of the most significant works available in the literature was presented, focusing on the aspect of pseudo-ortoghonality of SFs and on the ADR mechanism, which is possible only when downlink messages are employed. Moreover, an overview of the state of the art of models and simulators for a LoRaWAN network is given.

Chapter 4 described the `lorawan` module implementation for the network simulator ns-3, specifying models and assumptions on which the implementation is based. We also proposed a mathematical model to describe the performance of a LoRaWAN network taking realistic features into account, like the duty cycle constraint, the multiple receive paths at the gateway and the capture effect, that is one of the advantages of the LoRa modulation.

Then, extensive simulations compared the performace of networks using unconfirmed or confirmed traffic. For a network employing confirmed messages, various parameters were analyzed. Results show that the fact of retransmitting a message is not always a guarantee of reliability, but instead worse network performance when a high number of devices is involved. Moreover, we remarked the importance of the time period at which messages are generated by EDs for the packet success probability. Average delivery delays and causes of losses were also analyzed, and it was noted that, in large networks, the limited number of receive paths in the GW the most caused delivery failures. It was shown that, as expected, the payload size of downlink messages impacts the network performance, while the presence of a random delay between consecutive retransmission as defined in the standard does not have a significant effect.

The last simulation campaign analyzed the scalability of a realistic LoRaWAN network, where only 30% of EDs required confirmation messages. A urban scenario, with the presence of shadowing and buildings in addition to the path loss was

considered. The simulation showed that the network achieved acceptable results even when a large numbers of EDs were considered.

Finally, the results of our mathematical model were compared to the ones obtained from the simulator. It has come to light that our model is more realistic than that proposed in [48], as it considers realistic elements that were not taken into account so far. The validity of the model is confirmed by results of the simulator.

A number of points that could be improved as part of future work are discussed here.

Further investigations could be conducted on the optimal SF allocation, so as to maximize the advantage brought by pseudo-ortoghonality and to minimize the packet transmission time. Simulations considering multiple GWs could be conducted, to make the analysis even more realistic, as it is expected that multiple LoRaWAN cells will be employed in the same area to improve the reliability of packet delivery. As discussed in the related work survey, the presence of multiple gateways will increase the network performance, as it is more likely that EDs will use lower SFs. Moreover, the downlink traffic will not increase with the number of GWs, as downlink messages are sent and controlled by the NS, that only transmits the message once and always chooses the best GW, minimizing the transmission time and, therefore, interference in the network. Implementation of ADR algorithms could allow evaluation of their performance.

The model proposed in this thesis can also be extended taking multiple retransmissions into account. In particular, the traffic caused by retransmissions should be added to the traffic generated by EDs that has been considered until now. Moreover, further investigations should be conducted on the missing elements or assumptions that make the performance of the simulator and of the model differ.

Bibliography

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: A survey," *arXiv preprint arXiv:1606.07360*, 2016. version 1.

[2] K. Mikhaylov, J. Petajajarvi, and J. Janhunen, "On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-Network Interference," *arXiv preprint arXiv:1704.04257*, 2017.

[3] Semtech corporation, "SX1272 datsheet," March 2015.

[4] M. Knight and B. Seeber, "Decoding LoRa: Realizing a Modern LPWAN with SDR," in *Proceedings of the GNU Radio Conference*, vol. 1, 2016.

[5] LoRa Alliance, "LoRaWAN^{TM} 1.1 Specification," *LoRa Alliance*, October 2017.

[6] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[7] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *Communications (ICC), 2017 IEEE International Conference on*, pp. 1–6, May 2017.

[8] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive Configuration of LoRa Networks for Dense IoT Deployments,"

[9] Semtech corporation, "SX1301 datsheet," June 2014.

[10] LoRa Alliance, "LoRaWAN^{TM} 1.1 Regional Parameters," *LoRa Alliance*, 2017.

[11] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: A survey," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 585–598, 2015.

[12] J. Morgan, "A Simple Explanation Of 'The Internet Of Things'," *Forbes*, 2014. [Online; accessed 30-12-2017].

[13] "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016," 2017. [Online] Available: https://www.gartner.com/newsroom/id/3598917.

[14] L. Columbus, "2017 Roundup Of Internet Of Things Forecasts," *Forbes*, 2017. [Online; accessed 30-12-2017].

[15] "GrowthEnabler, Market Pulse Report, Internet of Things (IoT)," tech. rep., April 2017. [Online] Available: https://growthenabler.com/flipbook/pdf/IOT%20Report.pdf.

[16] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[17] S. M. Darroudi and C. Gomez, "Bluetooth Low Energy Mesh Networks: A Survey," *Sensors*, vol. 17, no. 7, p. 1467, 2017.

[18] B. Mihajlov and M. Bogdanoski, "Overview and analysis of the performances of ZigBee-based wireless sensor networks," *International Journal of Computer Applications*, vol. 29, no. 12, pp. 28–35, 2011.

[19] R. Challoo, A. Oladeinde, N. Yilmazer, S. Ozcelik, and L. Challoo, "An overview and assessment of wireless technologies and co-existence of ZigBee, Bluetooth and Wi-Fi devices," *Procedia Computer Science*, vol. 12, pp. 386–391, 2012.

[20] JFR, "Z-Wave Protocol Overview," tech. rep., Zensys, April 2006.

[21] D. Flore, "GPP Standards for the Internet-of-Things," February 2016.

[22] Ericsson, "Cellular Networks for Massive IoT," tech. rep., January 2016. [Online] Available: https://www.ericsson.com/assets/local/publications/white-papers/wp_iot.pdf.

[23] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range IoT technologies: The dawn of LoRa^{TM}," in *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pp. 51–58, Springer, 2015.

[24] C. Goursaud and J.-M. Gorce, "Dedicated networks for IoT: PHY/MAC state of the art and challenges," *EAI endorsed transactions on Internet of Things*, 2015.

[25] Semtech, AN1200.22, "LoRa^{TM} Modulation Basics, Application Note," 2015.

[26] M. Knight, Bastille Research, "gr-lora," [Online] Available: https://github.com/BastilleResearch/gr-lora.

[27] M. Knight, Bastille Research, "Reversing LoRa," 2016. [Online] Available: https://static1.squarespace.com/static/54cecce7e4b054df1848b5f9/t/57489e6e07eaa0105215dc6c/1464376943218/Reversing-Lora-Knight.pdf.

[28] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[29] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa Scalability: A Simulation Model Based on Interference Measurements," *Sensors*, vol. 17, no. 6, p. 1193, 2017.

[30] L. Vangelista, "Frequency shift chirp modulation: The lora modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, 2017.

[31] D. Croce, M. Gucciardo, I. Tinnirello, D. Garlisi, and S. Mangione, "Impact of spreading factor imperfect orthogonality in lora communications," in *International Tyrrhenian Workshop on Digital Communication*, pp. 165–179, Springer, 2017.

[32] M. Bor, U. Roedig, T. Voigt, and J. Alonso, "Do lora low-power wide-area networks scale?," in *The 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2016.

[33] LoRa Alliance, "LoRaWAN^{TM} 1.0.2 Specification," *LoRa Alliance*, July 2016.

[34] D. Magrin, "Network level performances of a LoRa system," 2016.

[35] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of lpwans: range evaluation and channel attenuation model for lora technology," in *ITS Telecommunications (ITST), 2015 14th International Conference on*, pp. 55–59, IEEE, 2015.

[36] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, "Evaluation of lora and lorawan for wireless sensor networks," in *SENSORS, 2016 IEEE*, pp. 1–3, IEEE, 2016.

[37] N. Vatcharatiansakul, P. Tuwanut, and C. Pornavalai, "Experimental performance evaluation of lorawan: A case study in bangkok," in *Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on*, pp. 1–4, IEEE, 2017.

[38] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in lorawan based lpwa networks?," *arXiv preprint arXiv:1704.04174*, 2017.

[39] K. Mikhaylov, J. Petajajarvi, and J. Janhunen, "On lorawan scalability: Empirical evaluation of susceptibility to inter-network interference," *arXiv preprint arXiv:1704.04257*, 2017.

[40] P. Ferrari, A. Flammini, M. Rizzi, E. Sisinni, and M. Gidlund, "On the evaluation of LoRaWAN virtual channels orthogonality for dense distributed systems," in *Measurement and Networking (M&N), 2017 IEEE International Workshop on*, pp. 1–6, IEEE, 2017.

[41] V. Hauser and T. Hégr, "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure," in *Future Internet of Things and Cloud (FiCloud), 2017 IEEE 5th International Conference on*, pp. 85–90, IEEE, 2017.

[42] "The Things Network," [Online] Available: https://www.thethingsnetwork.org.

[43] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of latency and mac-layer performance for class a lorawan," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, 2017.

[44] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "A Fair Adaptive Data Rate Algorithm for LoRaWAN," *arXiv preprint arXiv:1801.00522*, 2018.

[45] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "Lora scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, 2017.

[46] Z. Li, S. Zozor, J.-M. Drossier, N. Varsier, and Q. Lampin, "2D Time-frequency interference modelling using stochastic geometry for performance evaluation in Low-Power Wide-Area Networks," in *Communications (ICC), 2017 IEEE International Conference on*, pp. 1–7, IEEE, 2017.

[47] G. Ferré, "Collision and packet loss analysis in a lorawan network," in *Signal Processing Conference (EUSIPCO), 2017 25th European*, pp. 2586–2590, IEEE, 2017.

[48] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of lorawan channel access," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*, pp. 1–3, IEEE, 2017.

[49] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of lorawan channel access with capture effect," in *International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2017)*.

[50] F. V. d. Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *arXiv preprint arXiv:1705.05899*, 2017.

[51] 3GPP, "Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT)," Tech. Rep. 45.820 V13.1.0, November 2015.

[52] J. Lee and F. Baccelli, "On the effect of shadowing correlation on wireless network performance," *arXiv preprint arXiv:1712.00900*, 2017.

[53] "The ns-3 network simulator," 2016. [Online] Available: https://www.nsnam.org/.

[54] P. L'Ecuyer, "Random number generation," in *Handbook of Computational Statistics*, pp. 35–71, Springer, 2012.