MASTER'S THESIS
TELECOMMUNICATION ENGINEERING

Improving LoRaWAN networks performance

Author
Natalia Chinchilla Romero

Supervisor
Jorge Navarro Ortiz

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, 9^{th} of July, 2021

Improving LoRaWAN networks performance

Author
Natalia Chinchilla Romero

Supervisor
Jorge Navarro Ortiz

Improving LoRaWAN networks performance

Natalia Chinchilla Romero

Keywords: IoT, LoRa, LoRaWAN, Capacity, Performance, PER.

Abstract

Connectivity within the well-known Internet of Things paradigm continues to grow by leaps and bounds. The main technologies driving IoT connections are 4G/5G cellular networks as well as Low Power Wide Area Networks (LPWANs). The latter are characterised by a massive number of devices remotely connected to one or several central stations in areas up to tens of kilometres away and powered by batteries with a lifetime of up to 10 years, contributing to low energy consumption and low infrastructure costs.

One of the main LPWAN prevalent in the market today is known as LoRaWAN, which is the focus of this project. This technology operates in the unlicensed spectrum and is driven by the LoRa Alliance (a non-profit association of companies and telecommunications operators from all over the world).

LoRaWAN has certain limitations that are currently under investigation. One of them is the limitation of network capacity. The medium access protocol used in LoRaWAN, known as ALOHA, does not allow devices to listen to the medium before transmitting. Thus, as the number of devices connected to the network increases, the number of packets that can collide also increases, as limited resources are occupied inefficiently. This results in a limitation in the number of devices connected to the network, as well as the information that can be sent simultaneously over the available radio resources, thus degrading the performance of LoRaWAN networks.

In this project, a solution is proposed in order to mitigate the capacity limitation of the LoRaWAN networks. This solution consists of the design and implementation of an efficient radio resource allocation algorithm by means of which LoRaWAN devices can make optimal use of these resources and reduce the number of collisions. In order to introduce this solution, a LoRaWAN simulator has been designed and implemented following the guidelines of the LoRaWAN standard and its performance has been validated. After implementing the new algorithm in this simulator, a series of simulations have been carried out in order to extract the main advantages and limitations of this solution. A proof of concept with commercial hardware equipment is also included.

Mejora de prestaciones en redes LoRaWAN

Natalia Chinchilla Romero

Palabras clave: LoRa, LoRaWAN, Capacidad, Rendimiento, PER.

Resumen

La conectividad perteneciente al conocido paradigma del Internet de las Cosas continúa creciendo a pasos agigantados. Las principales tecnologías impulsoras de las conexiones IoT son las redes celulares 4G/5G así cómo las conocidas como redes de área amplia y baja potencia (LPWAN). Estas últimas se caracterizan por incluir un número masivo de dispositivos remotamente conectados a una o varias estaciones centrales en áreas de hasta decenas de kilómetros de distancia y alimentados por baterías con una durabilidad de hasta 10 años, contribuyendo a un bajo coste de consumo energético y bajo coste en infraestructura.

Una de las principales LPWAN que predominan en el mercado a día de hoy es la conocida como LoRaWAN, en la cual se centra este proyecto. Esta tecnología opera en el espectro sin licencia y está impulsada por la LoRa Alliance (una asociación sin ánimo de lucro en la que participan empresas y compañías de telecomunicaciones de todo el mundo).

LoRaWAN presenta ciertas limitaciones que están actualmente en constante investigación. Una de ellas es la limitación en la capacidad de la red. El protocolo de acceso al medio utilizado en LoRaWAN, conocido como ALOHA, no permite que los dispositivos escuchen al medio antes de transmitir. Por ello, a medida que el número de dispositivos conectados a la red aumenta, el número de paquetes que puedan colisionar también aumenta, ya que se ocupan los recursos limitados de manera poco eficiente. Esto se traduce a una limitación en el número de dispositivos conectados a la red, así cómo a la información que se pueda enviar de manera simultánea a través los recursos radio disponibles, degradando así el rendimiento y prestaciones de las redes LoRaWAN.

En este proyecto, se propone una solución con el fin de mitigar la limitación de la capacidad de la red LoRaWAN, consistente en el diseño e implementación de un algoritmo de asignación de recursos radio de manera eficiente mediante el cual los dispositivos LoRaWAN puedan hacer un uso óptimo de estos y reducir el número de colisiones. Para poder introducir esta solución, previamente se ha diseñado e implementado un simulador LoRaWAN siguiendo las directrices de su estándar y se ha validado su funcionamiento. Tras introducir el nuevo algoritmo a este simulador, se han realizado una serie de simulaciones con el fin de extraer las principales ventajas y limitaciones de esta solución. Además se incluye una prueba de

concepto con equipo hardware comercial.

Yo, **Natalia Chinchilla Romero**, alumna de la titulación INGENIERÍA DE TELECOMUNICACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI [redacted], autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Master en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Natalia Chinchilla Romero

Granada a 9 de Julio de 2021.

D. **Jorge Navarro Ortiz**, Profesor del Área de Telemática del Departamento Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Improving LoRaWAN networks performance***, ha sido realizado bajo su supervisión por **Natalia Chinchilla Romero**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 9 de Julio de 2021.

El director:

Jorge Navarro Ortiz

Acknowledgments

I would like to express my sincere gratitude to some people who in some way have contributed in this project's completion.

Firstly and foremost I am very grateful to my supervisor Jorge, for his support, kindness, availability and for all his guidance along this project.

I am also very grateful to WiMuNet Lab group for their belief and letting me be a member of the research group.

I would like to thanks to my research colleagues for all their unwavering support and advice whenever I needed it.

I heartily want to thanks my parents and my brother for their affection, guidance and support, specially in this though year, that, despite of the distance I feel it always with me.

I specially want to be grateful to Lorena, my sister, for been the best sister, workmate, friend and life companion ever and also for all your emotional support.

I would also like to thank Ramón, for all his patience, advice, affection and encourage.

At last, I am deeply grateful to all my friends for giving me peace of mind and cheering me up whenever I needed it.

Thank you very much.

Contents

List of Figures

List of Tables

Acronyms

4G Fourth Generation.

5G Fifth Generation.

ABP Activation By Personalization.

ACK Acknowledgment.

ADR Adaptive Data Rate.

AES Advanced Encryption Standard.

BLE Bluetooth Low Energy.

BW bandwidth.

CARA Collision Avoidance Resource Allocation Algorithm.

CR Code Rate.

CRC Cyclic Redundancy Check.

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance.

CSS Chirp Spread Spectrum.

DR Data Rate.

eMBB enhanced Mobile Broad Band.

ERP Effective Radiated Power.

ETSI European Telecommunications Standards Institute.

FEC Forward Error Correction.

FSK Frequency Shift Keying.

ID identifier.

IoT Internet of Things.

IP Internet Protocol.

ISM Industrial, Scientific and Medical.

JCR Journal Citation Report.

LoRa Long Range.

LoRaWAN Long Range Wide Area Network.

LPWAN Low Power Wide Area Network.

LTE-M Long Term Evolution Machine Type Communication (MTC).

MAC Medium Access Control.

MIC Message Integrity Code.

mMTC massive Machine To Machine Communications.

MQTT Message Queuing Telemetry Transport.

NB-IoT Narrow Band IoT.

NTP Network Time Protocol.

OTAA Over The Air Activation.

PER Packet Error Rate.

RB Resource Block.

RF Radio Frequency.

SF Spreading Factor.

SFs Spreading Factors.

SNR Signal to Noise Ratio.

TDMA Time Division Multiple Access.

ToA Time on Air .

URLL Ultra Reliable Low Latency.

WiFi Wireless Fidelity.

Chapter 1

Introduction

According to IoT analytics [1], the global IoT connections are growing from 8 billion in 2018 up to almost 31 billion of expected IoT connections in 2025 (see Figure 1.1). IoT applications are gaining huge popularity in a broad range of private, public and industrial areas such us smart cities, smart homes, smart agriculture, industrial transportation, healthcare, etc. The latest 2020 analysis in [2] shows that most IoT projects still happen in manufacturing (industrial setting, with verticals such as transportation/-mobility), energy, retail and healthcare.

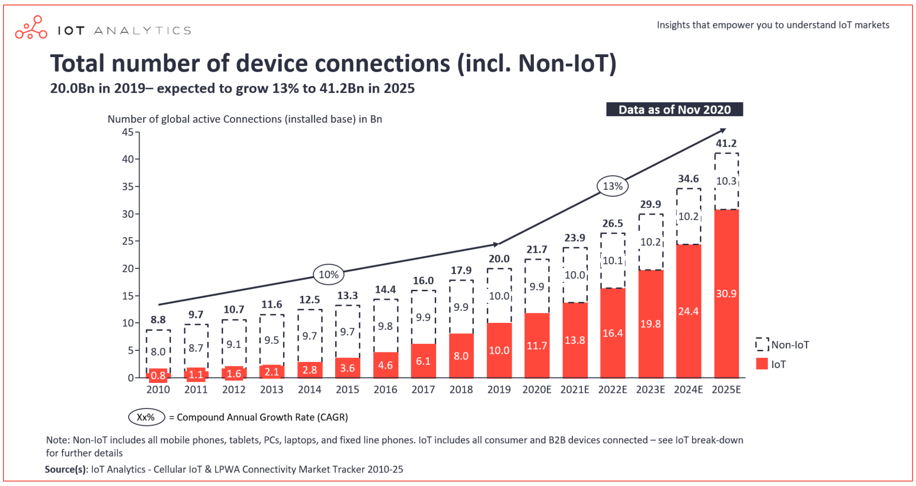


Figure 1.1: Number of global IoT connections (in red) according to IoT analytics [1].

Many IoT cases can be considered as massive Machine To Machine Communications (mMTC), one of the three key use cases considered in Fifth Generation (5G) (in addition to enhanced Mobile Broad Band (eMBB) and Ultra Reliable Low Latency (URLL) communications). Remote sensing and monitoring systems are gaining momentum in all sectors worldwide, due to the interest in monitoring large areas with minimal cost and effort. A key driver of the growth of these IoT massive connections are the Low Power Wide Area Network (LPWAN) technologies. These technologies provide low power, low data rate communications over long distances (up to several tens of kilometers with a single station) or from deep indoor environments, enabling battery-operated devices to operate for up to ten years without any human intervention.

There is a huge fragmentation of different LPWAN technologies in the market today. Nevertheless, LoRaWAN, Sigfox, Narrow Band IoT (NB-IoT), Long Term Evolution Machine Type Communication (MTC) (LTE-M) are emerging as the most popular LPWAN technologies both in terms of end-user adoption as well as ecosystem support. NB-IoT (which works in a licensed frequency band) and LoRaWAN (which works in a non-licensed frequency band) are forecasted to be the dominant technologies in public network and private network deployments respectively [2].

Among all the LPWAN technologies available in the market today, LoRaWAN is one of the most promising solutions that has attracted attention from both the industry and the academia, since it fulfills the long-range and low-power requirements, allowing bit rates between 250 bps and 11 kbps in the non-licensed Industrial, Scientific and Medical (ISM) band. This open standard defines the MAC layer and the network topology. Moreover, LoRaWAN has been deployed in 157 countries around the world and is backed by the LoRa Alliance, a non profit association of more than 500 members [3].

Although LoRaWAN has become one of the most popular LPWAN technologies due to its characteristics and ease of deployment, there are several weak points that the research community has identified. One of the main limitations in LoRaWAN is the reduction of capacity due to collisions caused by the ALOHA channel access mechanism and, consequently, this aspect leads to a degradation of the overall network performance. Moreover, as it is highlighted in [4], there is a capacity limitation due to the duty cycle regulations that limit the transmission time for each end-device in a sub-band, and, also, by the number of collisions produced by the use of low DR. This issue leads to the realization of this project, in which the coordination of the resources over the ALOHA channels is the main challenge addressed in order to achieve the maximum possible capacity.

For this reason, this project proposes some modifications of the LoRaWAN MAC layer which improves network capacity by employing a smart algorithm for resource assignment. More precisely, this project proposes an efficient resource allocation algorithm to be executes between the LoRaWAN entities in order to use in a intelligent manner the radio resources available in this kind of networks so as to increment network capacity and the overall network performance. A LoRaWAN standard network simulator is first designed and implemented in order to incorporate the new resource allocation algorithm afterwards. A proof of concept would be also carried out for testing the solution with real equipment.

1.1 Background and motivation

The LPWAN market was almost non-existent 5 years ago. The large number of IoT vertical services has made this paradigm enjoys strong momentum. This market reached 423 million devices in 2020 and by 2030 the number of LPWAN connections is expected to surpass any other IoT technology.

Currently, LoRaWAN and NB-IoT are the leading technologies, being LoRaWAN the technology that works in an unlicensed band and NB-IoT the one working in a licensed band.

The main characteristics of LoRaWAN that make it such an attractive technology are the ease and low cost of deployment, allowing long coverage distances and devices with a long battery life. This is why LoRaWAN is the technology that has been most widely adopted by the industry and the academia.

One of the main limitations of LoRaWAN is its overall capacity. Class A devices use the ALOHA protocol to access the transmission channel without any media listening procedure. This leads to a high probability of packets collision belonging to different devices. So that, it results in a performance degradation of the overall network as it implies transmission delays due to the retransmissions of the collided packets. In addition, it also affects the battery degradation of the end devices.

Inside the scientific community there are several works proposing new MAC layer proposals in order to achieve better network capacity.

Tommaso Polonelli et. al. [7] propose an approach to increase the single channel capacity using the Slotted ALOHA protocol for the communication between end-nodes and gateways. They tested this proposal in a real environment with 24 nodes accessing the same channel and demonstrated that, in a high traffic set-up, the measured improvement was up to 5.8 in comparison with the Pure-ALOHA measured throughput. This solution requires the implementation of a different medium access protocol and, therefore, it is not compatible with current LoRaWAN networks.

In order to meet the requirements of the real-time traffic management in Industrial IoT Applications, Leonardi et. al. [8] worked on a solution based on a centralized MAC scheme. In this scheme, a central node synchronized all end-nodes broadcasting a periodic beacon. After a beacon, the time to send information is divided in two periods: the first is intended for non-periodic unconfirmed data in which end-nodes use the ALOHA mechanism for channel access, and the second is intended for periodic real-time confirmed data flows in which end-nodes access the channel with a Time

Division Multiple Access (TDMA) mechanism. This solution seems to be devised for periodic transmissions in the contention-free period, guaranteeing no collisions, high reliability and bounded end-to-end delays. However, in the content-access period (with the Pure ALOHA medium access strategy), reliability is not ensured for the non-periodic transmissions.

In [9], a synchronization and scheduling solution is proposed, in which a central entity assigns the communication slots to the end-nodes based on several policies and infrequent signaling messages. In this assignment, it is necessary to know the traffic periodicity and the clock accuracy of end-nodes, so the central entity can assign the required time slots. The limitation of this work is the requirement of periodic traffic, otherwise the central entity will not know the number of time slots to assign.

Another time-slotted based solution is presented in [10], where it is computed a hash algorithm with the nodes' addresses to assign them a specific time slot to transmit. The network server leverages the last slot sent in the broadcast frame in order to handle the acknowledgements and time synchronization at the expense of extra energy cost.

To achieve energy-efficiency and increase the network performance, the work in [11] designs a new LoRaWAN network architecture and a new MAC protocol based on unicast and broadcast TDMA communications. These communications are started by a sink and an intermediate entity, the cluster head, is responsible for the rely of wake-up beacons to synchronize the end-devices and avoid collisions. This solution needs the implementation of a non-standard LoRaWAN transceiver.

The different solutions proposed in EXPLoRa [12] are useful when taking into account different traffic types from different applications or in situations in which there are areas with different traffic demands requiring a balanced resource allocation. However, this solution requires a centralized entity that has to monitor and manage periodically the network behaviour.

FCA-LoRa [13] achieves a fairness resource allocation with collision avoidance. It is based on the broadcasting of periodic beacon frames sent by the gateways. This novel MAC protocol uses an alternative version of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm for listening the medium before the transmission of a packet (instead of using the ALOHA protocol used in original LoRaWAN).

The main differences between these works and our proposal are the following: 1) our proposal is compatible with current LoRaWAN networks since it works on top of the existing MAC protocol, and 2) after the initial LoRaWAN join procedure, no extra signaling or communication with the end-device is required.

1.2 Objectives

The main objective of this Master's Thesis is to propose a solution which contributes to the improvement of the overall performance of a LoRaWAN network. The solution should include a proposal that do not involve major modifications to the standard LoRaWAN MAC layer and still allow to contribute to the improvement of the network performance. In particular the specific objectives are the following:

- Software implementation of a simulator that reproduces a real LoRaWAN network following the guidelines of its standard [14]. The main purpose of this objective is to create a base scenario in order to introduce the different solutions that will be presented in the rest of the objectives of this project.
- Development of an algorithm running at the LoRaWAN MAC layer that will contribute to the enhancement of the overall network capacity and thus the overall network performance.
- Real experimentation of the proposed solution in a testbed to check its feasibility in a realistic environment.

1.3 Project structure

This section describes the general outline of this document in order to provide the reader with an overview of the content of the chapters of this project. The content of each chapter is detailed below:

- **Chapter 1: Introduction**: The first chapter of which this document is composed, tries to introduce and motivate the general objective of this project, introducing simple descriptions of the technology and the subject matter in order to facilitate the reader's reading of this document. In addition, in this chapter, previous works related to the problem to be dealt with are mentioned, as well as the specific objectives that this project intends to address. The structure followed in this document is also described in the last section of this chapter.
- **Chapter 2: Project planning and cost estimation**: In this chapter, first, a time schedule and description of the tasks involved in the realization of this project is detailed. A description of all the resources used in this project, as well as the final budget, are also included in this chapter.
- **Chapter 3: Technical overview of LoRa / LoRaWAN**: All the technical aspects of LoRa / LoRaWAN are included in this chapter. It

is provided a general overview of how this technology works, its typical network architect, types of devices ... etc. A detailed description of the LoRaWAN MAC layer is included and it is also highlighted its limitations in order to justify the importance of providing solutions to manage several weak point this technology has.

- **Chapter 4: LoRaWAN simulator implementation and validation**: This chapter is in charge of showing the first huge design and implementation block of this project. It consists on the design and implementation of a LoRaWAN simulator, following its standard guidelines. In addition, in this chapter it is included the implementation of a mathematical model of a LoRaWAN network coming from a scientific work in order to validate the functioning of the implemented simulator with this model.

- **Chapter 5: Design and implementation of an efficient radio resources allocation algorithm for LoRaWAN**: The second major design and implementation block of this project is included in this chapter, containing the proposal of an algorithm that could be included in the previously implemented simulator in order to allocate radio resources efficiently and to overcome the capacity limitation in this type of networks. The description of the different simulations that have been performed as well as a detailed discussion of the obtained results are described in this chapter.

- **Chapter 6: Proof of concept**: In order to provide a demonstration in which the solution proposed in this project can be implemented in a real LoRaWAN network infrastructure, this chapter describes the process that has been carried out for the testbed setup and tuning. A simple experiment is also included as a proof of concept.

- **Chapter 7: Conclusions and future work**: To bring to an end this document, this chapter highlights the most important conclusions of the work done as well as possible improvements and future work that would complement this work.

Chapter 2

Project planning and cost estimation

This chapter provides the estimated time planning that has been followed along this project, as well as the estimated budget which includes all the resources and costs involved in the execution of this project.

In this way, three sections has been included in this chapter. Section 2.1 includes the description and time estimation of each task that has been carry out to achieve all the project goals. The second section 2.2 describes all the indirect and direct resources that have been used for the realisation of this project. Finally, section three provides the estimated budget with the costs of this project.

2.1 Task planning and timing

The following, is the order of the tasks that has been carried out in order to achieve the objectives proposed in this project as well as a brief description of each one:

1. *Review of LoRa/LoRaWAN networks*.

   This first task includes all the study of the art about the LoRa technology and a thorough revision of the LoRaWAN standard in order to understand how this technology works at each of its levels.

2. *Study of the LoRaWAN limitations*.

   This task consists on identifying the weak points that this technology has and which are currently under investigation.

3. *Study of the LoRaWAN MAC layer*.

   The LoRaWAN MAC layer has been the main object of study in this project. A deeper study of this MAC layer in terms of entities involved, protocols and mechanisms has been carried out in this task.

4. *Design and implementation of the LoRaWAN standard network simulator*.

   A standard LoRaWAN network simulator is the main element in which the solution to alleviate the capacity issue proposed in this project will subsequently be implemented. So that, in this task the design and implementation of a LoRaWAN simulator that reproduces the standard LoRaWAN MAC layer has been conducted.

5. *Launching of simulations with the LoRaWAN standard simulator*.

   In order to ensure that the LoRaWAN standard simulator works as intended, a series of simulations must be executed in this task.

6. *Search and implementation of a LoRaWAN mathematical model*.

   Once the LoRaWAN simulator works as expected, in this task the research and implementation of a mathematical model of a LoRaWAN MAC layer is carried out in order to validate the correct functioning of the simulator with this model.

7. *Launching of math model simulations and validation*.

   Firstly, in this task, different simulations would be carried out to verify the correct implementation of the mathematical model. Then, a comparison will be made between the mathematical model and the LoRaWAN simulator in order to reliably validate its operation.

8. *Design and implementation of an efficient resource allocation algorithm for collision avoidance*.

   Once the final standard LoRaWAN simulator is available, this task is in charge of the study, the design and the implementation of an algorithm to allocate resources efficiently in order to avoid collisions and improve the capacity limitation of the LoRaWAN networks.

9. *Launching of simulations and results discussion*.

   In this task some campaign of simulations has been taken. The first couple of simulations have been carried out with ideal propagation conditions in order to compare the enhancement that this solution provides to the standard LoRaWAN MAC layer. The second couple of simulations have been performed with a real propagation model to prove that this solution can be used in real-world environments as well as some different modifications in the configuration of each simulation are carried out to characterise the effectiveness of this algorithm in different coverage areas.

10. *Design and implementation of a testbed for the proof of concept.*

    This task is in charge of the reproduction of a testbed with real hardware running the different entities of a real LoRaWAN network together with the proposed solution, in order to show a proof of concept which demonstrates that this solution can be introduced in a real physical network.

11. *Experimentation with the testbed.*

    Several tests with the installed testbed will be carried out for showing preliminary results of this solution with real equipment.

12. *Document drafting.* The final task of this project, collects all the documentation related to this master's thesis project, following the structure that has been previously explained in section 1.3.

In Table 2.1 is included the initial time planning for the tasks involved along the project that has been previously described.

Table 2.1: Time planning of the project.

Task	**Duration**
Review of LoRa/ LoRaWAN networks	10 days
Study of the LoRaWAN limitations	7 days
Study of the LoRaWAN MAC layer	7 days
Design and implementation of the LoRaWAN standard network simulator	40 days
Launching of simulations with LoRaWAN standard simulator	15 days
Search and implementation of a LoRaWAN mathematical model	7 days
Launching math model simulations and validation	7 days
Design and implementation of an efficient resource allocation algorithm for collision avoidance	45 days
Launching of simulations and results discussion	10 days
Design and implementation of a testbed for the proof of concept	35 days
Experimentation with the testbed	15 days
Document drafting	50 days

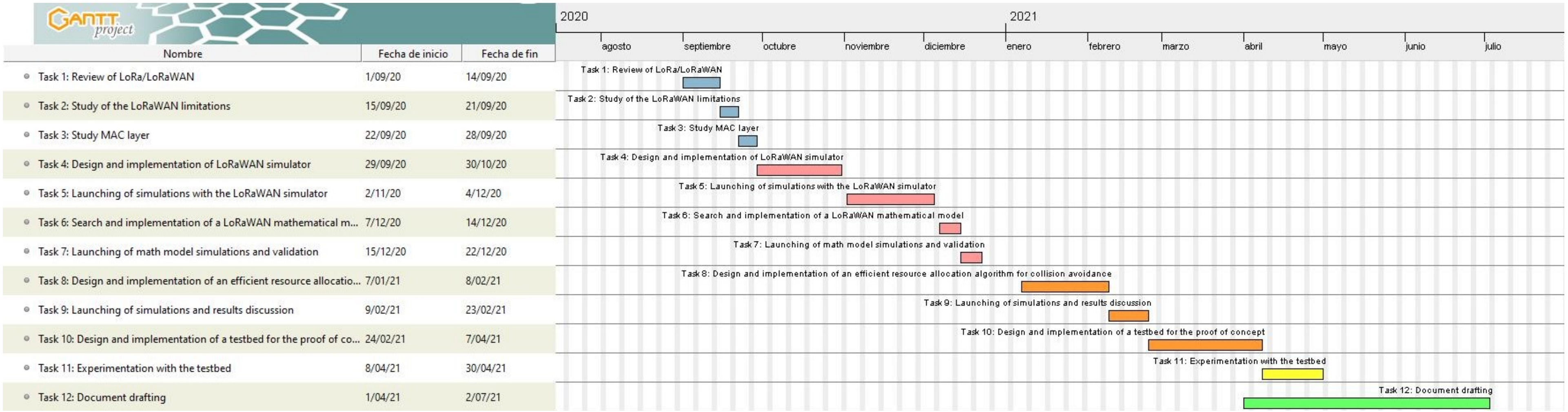


Figure 2.1: Gantt diagram with the task planning.

2.2 Resources planning

In this section the resources planning is included. With this respect, they has been classified into hardware resources, software resources and human resources.

2.2.1 Hardware resources

Regarding hardware resources it has been included a laptop in which all the simulations have been executed. Concretely, this laptop is a "Dell XPS 13 9360" with an Intel(R) Core(TM) i7-8550U CPU @1.80GHz 1.99 GHz, 16GB RAM and a 500GB SSD. All the hardware involved in the testbed for the proof of concept listed below:

- **IMST Lite Gateway 8080i [5]**: this gateway containts a Raspberry Pi connected to an IMST ic880A LoRaWAN concentrator for the ISM 868 MHz band.

- **Pycom's FiPy development board [6]**: this is the hardware employed for the end-devices. It is a multi-technology board (supporting Wi-Fi, Bluetooth Low Energy (BLE), LoRaWAN, SigFox and LTE-M/NB-IoT with an ESP32 SoC.

2.2.2 Software resources

All the programs and applications that have been used along the realization of this project and its use description are included in Table 2.2.

Table 2.2: Software resources description.

Software program/application	**Use description**
SO Windows 10 Home 64 bit	SO installed in the laptop
MATLAB R2018b	Implementation and execution of the LoRaWAN simulator and the new algorithm designed
GanttProject v.2.8.11	Design of the task planning
Overleaf online LaTex editor	Document drafting
Chirpstack platform	Open-source platform which provides components for the LoRaWAN networks
Visual Studio Code v1.55.2	Source code editor for programming the end-devices and the new server with MicroPython language and Python 3 respectively.

2.2.3 Human resources

The human resources of this project have been measured according to the number of people involved in this project and the amount of time each of them will expend working on it. The hours expended by the student have been calculated taking into account the sum of all the hours spent working on the tasks. Related to the tasks involved on the realization of the project the student will invest 6h/day approximately and for the document drafting will invest about 2h/day. The hours invested by the supervisor include all the meetings for clarifying all the doubts that will be arisen during the course of the project and for some guidance. In this respect, Table 2.3 shows the time employed in this project by the student and her supervisor:

- **Natalia Chinchilla Romero**: Student of M.Sc. Telecommunication Engineering of the University of Granada.
- **Jorge Navarro Ortiz**: Associate Professor of the Department of Signal Theory, Telematics and Communications of the University of Granada.

Table 2.3: Human resources description.

Person	**Time invested**
Natalia Chinchilla Romero (student)	1288 hours
Jorge Navarro Ortiz (supervisor)	50 hours

2.2.4 Project budget

An estimated project budget is presented in this section considering the unavailability of any of the resources prior to the realization of the project. Thus, the estimated costs of each one of the resources with the total project cost are provided in Table 2.4.

It is worth noting that it has been included the software that is not open source since it is needed a license for its use. Moreover, the student labor has been set as 25 euros per hour since it is considered as a junior engineer.

Therefore, it is concluded that the total budget that has been calculated for this Master's Thesis comes to 37.770€, thirty seven thousand seven hundred and seventy euros.

Table 2.4: Project budget.

Resource	**Units**	**Unit cost**	**Subtotal cost**
Laptop (Dell XPS 13 9360)	1 (175 days of use)	0,82 (€/day of use)	144 €
IMST Lite Gateway i880A	1	199 €	199 €
PYcom's FiPy development board	30	59,40 €	1.782 €
SO Windows 10 Home	1	145 €	145 €
Matlab R2018b (anual license)	1	800 €	800 €
Supervisor labor	50 hours	50 (€/hour)	2.500 €
Student labor	1288 hours	25 (€/hour)	32.200 €
TOTAL COST		**37.770,00 €**	

Chapter 3

Technical overview of LoRa/LoRaWAN

The scope of this chapter is to provide the reader with a condensed overview of the most important features of this technology, as well as to go into more detail on the MAC layer, in which the objectives of this project is focused.

3.1 The physical layer LoRa

In Figure 3.1 is depicted the protocol stack of this technology. It can be seen that LoRa represents the physical layer and it is behind the LoRaWAN MAC layer. The LoRa PHY layer is agnostic with higher layer implementations. So that, it allows LoRa to coexist and interoperate with existing network architectures.

The LoRa wireless technology was developed by Cycleo, a French start-up company which developed the LoRa modulation. In 2012, the company Semtech acquired Cycleo [15] and patented this technology [16].

LoRa is a modulation derivated of the Chirp Spread Spectrum (CSS). An advantage of the LoRa modulation is the low complexity of the receiver design due to the equivalent offsets in timing and frequency between the transmitter and the receiver [17]. It implements a series of different DRs using orthogonal Spreading Factors (SFs) which allows the system designer to choose the appropiate DR or SF for range or power in order to optimize the network performance in a constant BW. The data signal is modulated onto a chirp signal that increases or decreases its frequency with time. The chirp rate define the spectral BW of a LoRa signal. For example, a chip rate of 125 kHz corresponds to a chip rate of 125.000 chips/s. Assuming a fixed BW, the data rate can change depending on the employed SF. In LoRa, the SF (which varies between 7 and 12) is the number of raw bits carried per

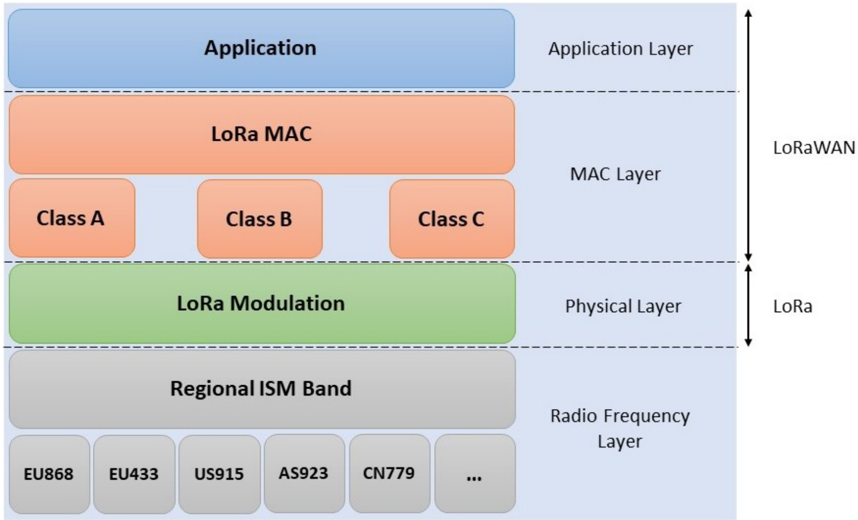


Figure 3.1: LoRa/LoRaWAN protocol stack.

symbol. Thus, the DR R_b is calculated as follows:

$$R_b = SF \times \left(\frac{BW}{2^{SF}}\right) \times \left(\frac{4}{4+CR}\right), \tag{3.1}$$

where the first term is the SF, the second term corresponds to R_s or symbol rate (symbol/s) and the third term depends on the Code Rate (CR) [1] (which varies between 1 and 4)[2]. Therefore, assuming fixed BW and CR, the data rate decreases as the SF increases.

The modulated information is then transmitted via LoRa radio following the packet format shown in Figure 3.2.

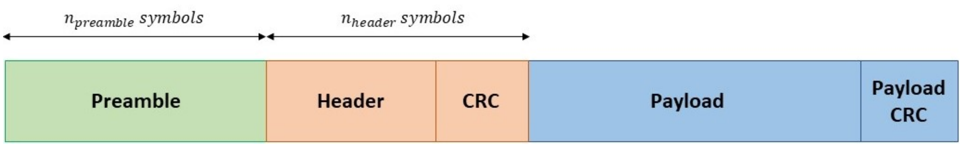


Figure 3.2: LoRa packet format.

The Time on Air (ToA) or the duration of the packet transmission is given by the sum of the preamble and payload duration as depicted in equation (3.2). The preamble is common to all modem configurations and its duration is given by expression (3.3), where $n_{preamble}$ is the number of programmed preamble symbols and T_{sym} is the symbol duration which can

[1]The LoRa modulation includes a Forward Error Correction (FEC) scheme in order to find and correct erroneous bits by adding redundancy in the code, improving the robustness of the transmitted signal [18].

[2]The default LoRa PHY configuration includes a coding rate of 4/5.

be defined as the necessary time to send 2^{SF} chips at the chip rate. T_{sym} is calculated with equation (3.4).

$$T_{packet} = ToA = T_{preamble} + T_{payload} \tag{3.2}$$

$$T_{preamble} = (n_{preamble} + 4.25) \cdot T_{sym}. \tag{3.3}$$

$$T_{sym} = \frac{2^{SF}}{BW}. \tag{3.4}$$

The payload duration is given by equation (3.5). The *payloadSymbNb* is defined as the number of symbols that form the header and the packet payload and it is calculated with equation 3.6, where PL is the payload length, H is equal to 0 when the explicit mode of the header is enabled and 1 otherwise [3] and DE corresponds with the low data rate optimization in order to increase the robustness of the LoRa link at low data rates, so that, it is enabled for $SF \geq 11$.

$$T_{payload} = payloadSymbNb \times T_{sym} \tag{3.5}$$

$$payloadSymbNb = 8 + max(ceil\left(\frac{8PL - 4SF + 28 + 16 - 20H + 1}{4(SF - 2DE)}\right), 0) \tag{3.6}$$

Table 3.1 presents a DR configuration assuming a 125 kHz BW and the default LoRaWAN code rate (4/5). As it can be seen, the bit rate increases and the ToA decreases as the SF is decremented.

Table 3.1: Data Rate configuration for a 125 kHz bandwidth in the EU868 ISM band.

DR	SF	Bit Rate (bps)	ToA for a 24 bytes packet (ms)
0	12	293	1482.75
1	11	540	823.30
2	10	980	411.65
3	9	1757	205.82
4	8	3125	113.15
5	7	5470	61.70

As it was mentioned in the Introduction (chapter 1), LoRa operates in the license exempt ISM band, i.e., the developer has not to pay any fee to

[3]If the explicit mode is enabled, it means that the LoRa physical header is included together with a Cyclic Redundancy Check (CRC) code. When explicit node is disabled, the header is removed because the properties of its fields are fixed.

used this frequency band. In Europe, the channel frequencies of the ISM band in which LoRa operates is known as the EU863-870 MHz ISM band. Its common name is called EU868 MHz which is the one that will be mentioned throughout this document. The EU868 MHz channel planning is shown in Figure 3.3. There are eight available LoRa channels with a BW of 125kHz and there is also a 250kHz LoRa channel that permits higher bit rates. The ninth available channel for uplink transmissions is a 250kHz Frequency Shift Keying (FSK) channel.

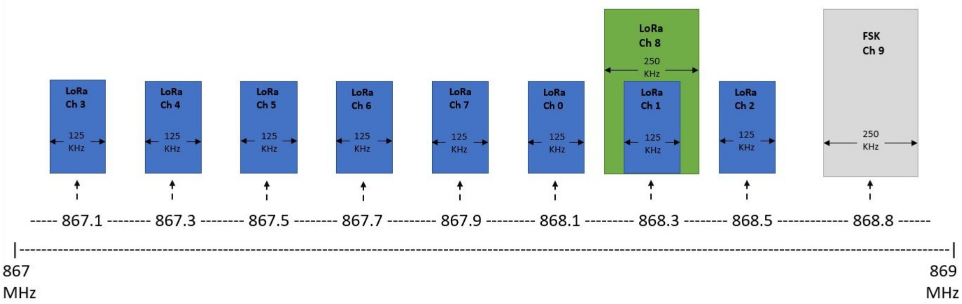


Figure 3.3: EU868 ISM band planning.

The LoRa alliance highlights that the channels can be freely attributed by the network operator, but the three following default channels shall be implemented in any end-device [19]:

Table 3.2: EU868 MHz default channels.

Modulation	**Bandwith [kHz]**	**Bit Rate [bps]**	**Duty cycle**
LoRa	125 kHz	293	< 1 %
LoRa	125 kHz	540	< 1 %
LoRa	125 kHz	980	< 1 %

When using the EU868 MHz ISM band, the users must comply with the European Telecommunications Standards Institute (ETSI) regulations [20], which are the following:

- The maximum transmission power for uplink tranmissions is limited to 25mW (14dBm). For downlink the maximum Effective Radiated Power (ERP) is limited to 0.5W (27dBm).
- There is a limitation in the maximum time a transmitter can transmit in an specific channel per hour. In particular, there is a 0.1% and 1% duty cycle per hour depending on the channel.
- The maximum allowed antenna gain should be +2.15dBi.

3.2 LoRaWAN

The LoRaWAN open standard defines the network architecture and describes the communication protocol to allow the connection between each entity of the network. It is defined by the LoRaWAN Alliance, which is a non-profit organization of more than 500 member companies and it is extended in more than 160 countries around the world. The protocol and network architecture have a big influence on determining the battery lifetime of a node, the network capacity, the security and the variety of applications than can be served by the network [21].

3.2.1 LoRaWAN architecture

A typical LoRaWAN network (see Figure 3.4) consists on a star-of-stars topology where one or more gateways (or concentrators) relay download and upload traffic between a variety of end-nodes (that may be either mobile or mounted at a fixed location) and a central network server respectively. The network server manages the traffic between each end-device and the associated application server. The communication between end-devices and gateways consists of single-hop LoRa or FSK connections, whereas gateways make use of standard and secured Internet Protocol (IP) connections via Wireless Fidelity (WiFi), Ethernet or either any cellular technology (Fourth Generation (4G), 5G, etc.) to the network server. All the communications are bi-directional, although traffic between end-devices and the central entities is predominantly uplink traffic.

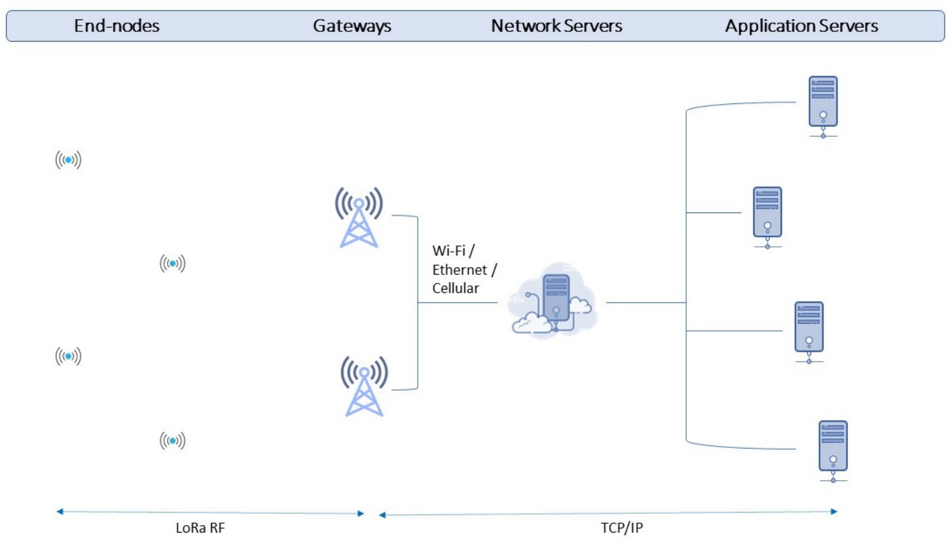


Figure 3.4: LoRaWAN network architecture.

3.2.2 LoRaWAN MAC

The LoRaWAN MAC layer describes the communication between the end-devices and the gateways and also how the end-devices access the medium in order to carry out a transmission. In the LoRaWAN specification [14] it is defined three types of LoRaWAN end-devices:

- **Class A end-devices**: This type of end-devices are also named basic LoRaWAN end-devices. This class is the most employed due to its low energy consumption. In Class A end-devices, two short receiving windows are opened after an uplink transmission in order to receive downlink transmission. The uplink transmissions in this class are scheduled using an ALOHA scheme, that is, the end-device transmits every time it has pending data without using any listen-before-talk mechanism. Class A end-devices are typically used for sensor nodes which are battery operated.

- **Class B end-devices**: These end-devices open extra receive slots at scheduled times. The end-devices receives a time synchronized beacon frames from the gateway in order to open its receive window. The time between beacons is known as the “beacon period” (which typically is 128 seconds), and the time during which the device is available to receive downlink transmissions is a “ping slot”.

- **Class C end-devices**: The end-devices of this class are continuously listening except when transmitting, i.e., the receive window is not closed. Class C is intended for mains powered actuators since continuous reception requires more power.

Figure 3.5 illustrates the transmission diagram for each class of end-device mentioned above.

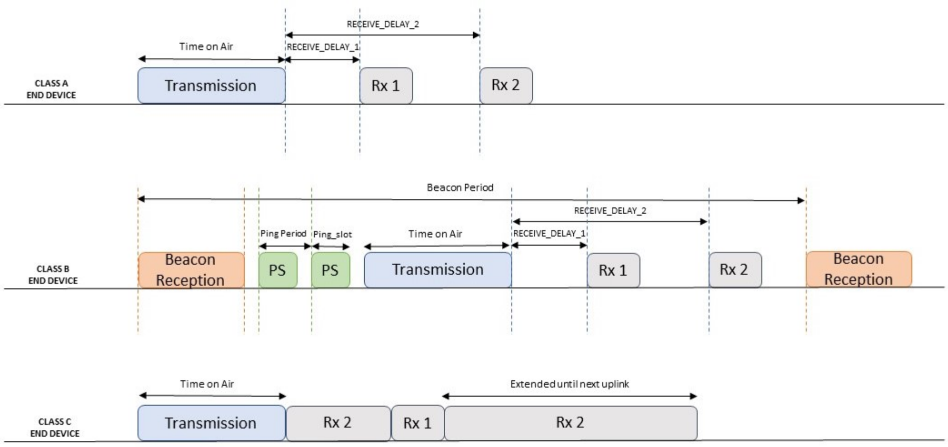


Figure 3.5: End-device classes diagram.

Each end-device which wants to join a LoRaWAN network has to perform an activation procedure in order to be personalized and activated. The activation of an end-device can be achieved in two ways, even by Over The Air Activation (OTAA) or Activation By Personalization (ABP). Before the activation each end-device is stored a global end-device identifier (ID) (*DevEUI*) that uniquely identifies the end-device and the two Advanced Encryption Standard (AES)-128 root keys *NwKey* and *AppKey* that are assigned to the end-device during its fabrication. After the activation, the following information is stored in the end-device: a device address of 32 bits which identifies the end-device within the current network (*DevAddr*), a network session key (*NwkSKey*) which is specific to each end-device and it is used by both the network server and the end-device to calculate an verify the Message Integrity Code (MIC) of all data frames to ensure data integrity and for encrypt and decrypt the payload field of the MAC data frames, and an application session key (*AppSKey*) which is also specific to each end-device and it is used by both the application server and the end-device to encrypt and decrypt the payload field of application specific data frames. The OTAA and the ABP procedures are described below:

- **OTAA**: With this type of activation, end-devices shall follow a join procedure prior to exchange data with a network server. The end-device shall initiate a new join procedure with its *DevEUI*, the *AppKey*, a *DevNonce* which is a counter that is incremented with every "Join-Request" and an application identifier *AppEUI*.

- **ABP**: This activation procedure does not require any signaling processing. That is to say that each end-device is directly stored the *DevAddr* and the two session keys *NwkSKey* and *AppSKey* instead of being derived from the *DeEUI* and *AppKey*.

Figure 3.6 shows both activation procedures mechanism that have been described above.

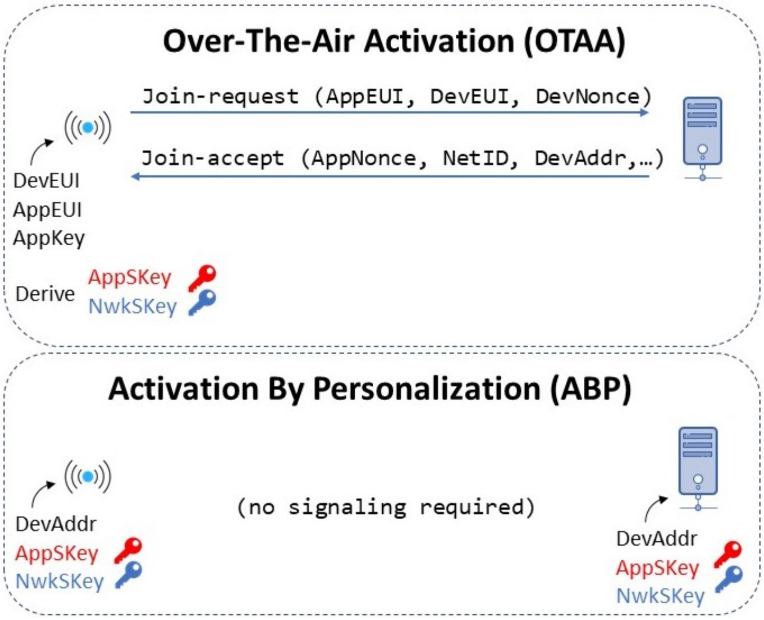


Figure 3.6: Activation procedures.

The transmissions between the end-nodes and the gateways are carried out on the available frequency sub-bands and the SFs. The selection of the SF is mainly determined by the distance between the end-device and the gateway.

The LoRaWAN specification recommends the use of an Adaptive Data Rate (ADR) scheme to manage the DR and the Radio Frequency (RF) output for each end-device in order to maximize its battery life, ensure reliable packet delivery and scale for the overall network capacity. The ADR scheme tries to tailors the node´s DR to the available link budget. The network server is the responsible of choosing the appropriate DR. Based on the IP packets with metadata about the reception time and signal strength the network server receives from the gateway, it determines what optimal DR (that is to say, the SF) should be assigned to the end-device. The DR that the node should use, is sent back to the device from the network server through the gateway with the best signal strength.

An example of a recommended ADR algorithm proposed by Semtech is described in [22]. The network server determines the proper DR based on the maximum Signal to Noise Ratio (SNR) of the last 20 frames received from the end-device, the last frame´s SNR and a margin.

The ADR algorithm is suitable for static end-devices. It is not recommended for mobile devices as the propagation radio path changes too rapidly.

Even if the ADR algorithm is used or not, the end-devices should respect the following actions:

- The end-device change the channel of each new transmission following a pseudo fashion mechanism in order to avoid interferences.

- Whenever a re-transmission is performed, the end-device must retry the next transmissions with a lowest SF (which is more robust).

The uplink messages are sent by the end-devices to the network server relayed by one or many gateways. Each downlink message is sent by the network server to a specific end-device and is relayed by a single gateway. Every first receive window (RX1) utilises the same frequency and SF as the recent uplink transmission by default, while the relationship between the uplink and RX1 DR or SF is region specific. In the EU868 MHzISM band the downlink DR could be changed following the uplink DR and a MAC parameter called *RX1DROffset* as given in Table 3.3. The RX2 receive window uses a fixed frequency and DR. The default parameters in Europe are 869.525MHz / DR0 (SF12, 125 kHz).

RX1DROffset	**0**	**1**	**2**	**3**	**4**	**5**
Uplink DR	**Downlink DR in RX1 slot**					
DR0	DR0	DR0	DR0	DR0	DR0	DR0
DR1	DR1	DR0	DR0	DR0	DR0	DR0
DR2	DR2	DR1	DR0	DR0	DR0	DR0
DR3	DR3	DR2	DR1	DR0	DR0	DR0
DR4	DR4	DR3	DR2	DR1	DR0	DR0
DR5	DR5	DR4	DR3	DR2	DR1	DR0
DR6	DR6	DR5	DR4	DR3	DR2	DR1
DR7	DR7	DR6	DR5	DR4	DR3	DR2

Table 3.3: EU868 downlink RX1 DR mapping [19]

The data messages exchanged between end-devices and the network server are used to transfer both MAC commands and application data. There area three types of data messages:

- **Confirmed-data messages**: these messages must be acknowledged by the receiver (the network server) in one of the receive windows.

- **Unconfirmed-data messages**: they do not require any acknowledgment from the receiver.

- **Proprietary messages**: this type of messages has non-standard message formats and are not interoperable with standard messages and must only be used among devices that have a commom proprietary understanding.

When and end-device carries out the transmission of a confirmed message, it expects to receive in one of the receiving windows and acknowledgement. If there is not any acknowledgment frame the end-device will try to retransmit the same data again. The retransmission must be executed on a new frequency channel (following a single hopping mechanism) and at a different DR (preferable a lower DR). For example, if a transmission is executed in the 868.3 MHz sub-band and SF7, its corresponding retransmissions must be taken on any other frequency sub-band and SF8. The standard recommends a maximum number of 8 transmissions per end-device , i.e., a non acknowledgement transmission followed by 7 retransmissions. An illustrative example of this procedure is depicted in Table 3.4.

Transmission attempt	**DR**
1	DR
2	DR
3	max(DR-1,0)
4	max(DR-1,0)
5	max(DR-2,0)
6	max(DR-2,0)
7	max(DR-3,0)
8	max(DR-3,0)

Table 3.4: Example of retransmission strategy [19].

3.2.2.1 The ALOHA channel access mechanism

The ALOHA protocol (first deployed in 1971) appears to be the first protocol used in wireless communications [24]. A simple ALOHA scheme is called "pure ALOHA". In a P-ALOHA system a node transmits whenever it has available data to transmit. Every user carries out its transmissions in an unsynchronized manner over the same channels. So as, if another node transmits at the same time or executes its transmission before the end of the transmission of another device a collision will occur and the frames that were transmitted are lost and must be retransmitted. If not, each user waits an appropriate time to receive an acknowledgment from the destination if the packet has been correctly received. Figure 3.7 shows the ALOHA mechanism and a visual example of a collision detection.

We can modeling the traffic generation of an ALOHA network as follows (based on [25]):

- The traffic source consists of an infinite number of users that forms an independent Poisson source with a mean packet generation rate of λ packets/s.

- It is assumed that each user or end-device has at least one packet requiring transmission at any time.
- Every packet have a constant length requiring an amount of time (let's call it T_{device}) for transmission.
- The input rate or the average number of packets that are generated per transmission time will be denoted as S and it is equal to $S = \lambda \cdot T_{device}$. Under stable conditions, S represents the channel throughput rate.
- Since the total traffic channel is the sum of the new generated packets plus the retransmissions of the collided ones, it increases the mean offered traffic which can be denote by G. It is fulfilled that $G \geq S$.
- The interarrival times of all packets plus retransmissions are independent and exponential.

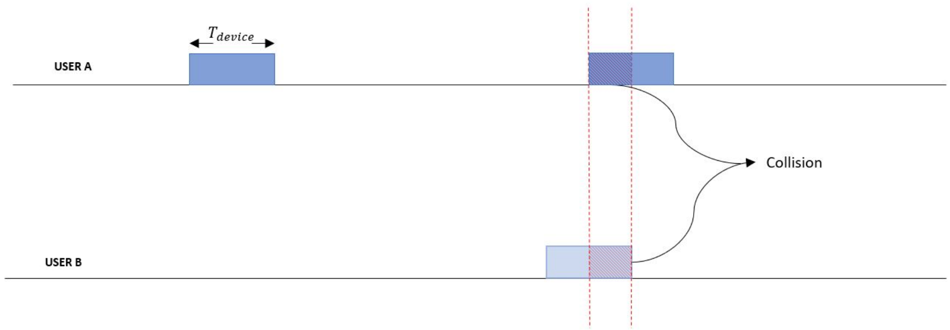


Figure 3.7: ALOHA mechanism.

3.2.2.2 Limitations of the LoRaWAN MAC layer

Class A end-devices are the most widely used in LoRaWAN networks due to their low power consumption, in addition to being the class that the standard determines that must be implemented in any LoRaWAN device. This class end-devices employs a light MAC protocol based on P-ALOHA (Pure ALOHA) for scheduling uplink transmissions from the end-devices. It is a simple multiple access protocol in which, every time the end-device has data to transmit, a packet is sent without any link coordination. There is not any listening mechanism to detect if there is any other transmission in the medium. Therefore, if two or more end-devices transmit data simultaneously, a collision will occur. The higher the number of end-devices accessing the network, the higher the number of collisions. Hence, this situation increases the PER, resulting in the degradation of the performance and the reduction of the network capacity.

Sundaram, et. al. [23] analyzes several research problems in LoRa networks such as energy consumption, communication range, multiple access, security, and so forth, and identifies many current works which try to provide appropriate solutions to mitigate these problems. One of the problems which significantly affects the performance of a LoRaWAN network is the medium access, which is related to the link coordination and the resource allocation in order to handle collisions and allocate reasonable resources to end-devices based on the deployed environment.

Adelantado et. al. [4] highlights the importance of the coordination in a single share infrastructure running many applications with different requirements (in terms of reliability, maximum latency etc.) using an ALOHA-based access.

Another remarkable limitation of the LoRaWAN MAC is the transfer rate. With LoRa modulation it can be achieved transfer rates of 5.47 kbps and 10.94 kbps with BWs of 125 kHz and 250 kHz respectively. In addition to this, the 1% duty cycle restriction in Europe, limits the time an end-device has to transmit data at a determined transfer rate (it corresponds with 36 seconds per hour in a particular frequency sub-band). This issue leads to a limitation in the transmission of delay sensitive applications such us for example image or video transfer in emergency applications.

Chapter 4

LoRaWAN simulator implementation and validation

As a first step in the design and implementation of a solution that tries to improve the LoRaWAN capacity, it is necessary to characterize a standard LoRaWAN network. So that, it has been designed a LoRaWAN simulator in MATLAB to reproduce the behavior of a real LoRaWAN network. The LoRaWAN overall system performance has been evaluated. Additionally, the validation of this simulator has been done with the implementation of a mathematical model of the LoRaWAN channel access.

4.1 Design of the simulator

As it was described in subsection 3.2.2.1, the LoRaWAN class A end-devices perform their transmissions following an ALOHA channel access mechanism. A MATLAB simulator implementating the LoRaWAN network behavior has been designed.

Some considerations have been taken into account:

- Downlink messages with acknowledgement messages had not been used due to the fact that class A devices are commonly designed for transmitting sensors data without taking into consideration the acknowledgement phase.
- It has been configured and used all the available 125 kHz LoRa frequency channels in the 868 MHz ISM band (a total of 8 channels). Moreover it has been taken into consideration the 6 available SF (from SF7 to SF12).

- Ideal propagation conditions have been assumed. Moreover, the end-devices are fixed.

- Each end-device is assigned a random SF following an uniform distribution.

- The selection of the channels is carried out following a cyclical algorithm.

- Since the duration time of each reception window is usually $T_{preamble}$ and the time to open each reception window is one second, this time can be calculated as $T_{device} = T_{frame} + 2 \cdot T_{preamble} + 2$ s (depicted in Figure 4.1). In the worst case, with SF12, this means that $T_{device} = 2.62 \cdot T_{frame}$. A collision will occur if another transmission starts in the same frequency and SF before T_{device} ends.

- A transmission matrix keeps track of all the actual transmissions time instants of each end-device and is updated every time a transmission or retransmission is carried out. In this transmissions matrix, the end-devices which are going to transmit at the current time are searched, and the vectors of transmissions on that particular channel and SF where that current transmission is being made are extracted. In Figure 4.2 is shown a visual scheme of the operation of this transmissions matrix and the extraction of the transmission vector in a specific channel and SF. In that example, the matrix saves the last transmissions of 5 end-devices. If, for example, the end-device 'dv2' starts its transmission, the vector corresponding to the channel and SF in which it is transmitting (called $Vector\_CHi\_SFj$) is updated with all the transmissions of the end-device that are transmitting or will transmit in CH1 and SF7. So, in this case, transmissions of the end-devices 'dv2' and 'dv4' may or may not collide.

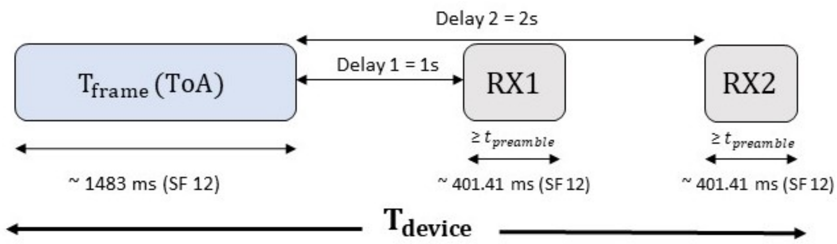


Figure 4.1: T_{device}.

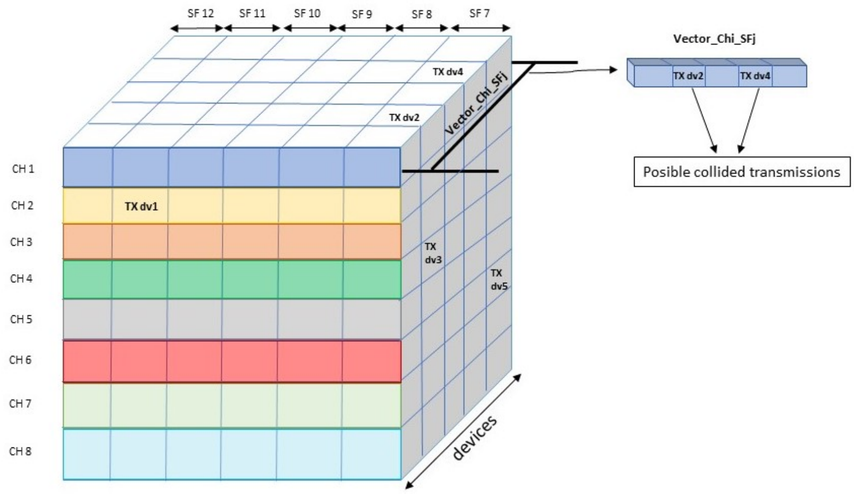


Figure 4.2: $Matrix_{tx}$ scheme.

Figure 4.3 represents the flow diagram of the operation of the LoRaWAN simulator. It consists on a time based simulator in which the status of the LoRaWAN network is checked every millisecond. Once a traffic loading point G is simulated for an hour, the simulator increments this parameter until it reaches a maximum value passed as an input parameter in the simulator setup. All the different scripts and functions have been implemented on MATLAB software. Moreover, a pseudo-code of the main script that has been implemented is described in Algorithm 1.

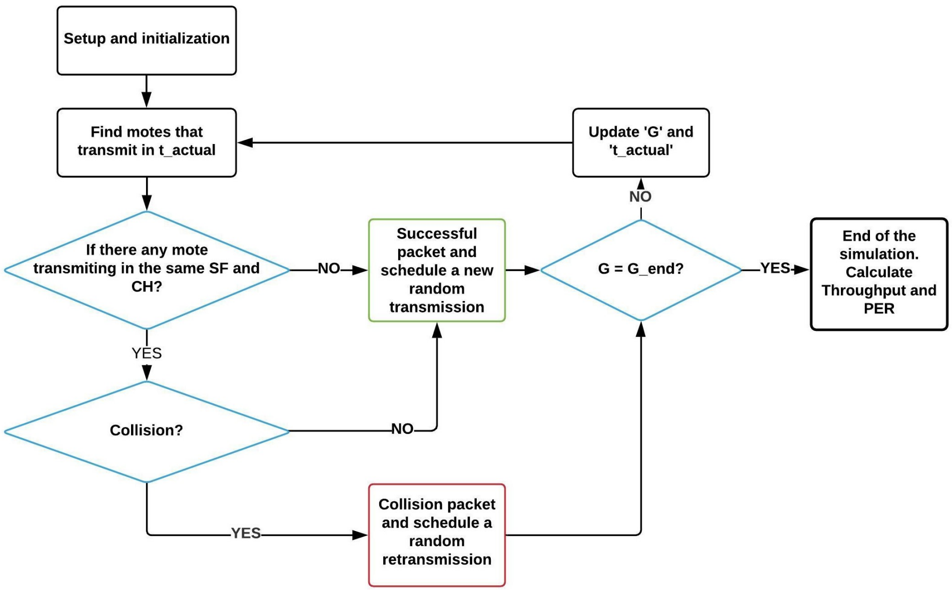


Figure 4.3: LoRaWAN simulator flow diagram.

Algorithm 1: *main.m* of the LoRaWAN simulator.

```
initialization (t = 0, G = 0);
setup;
for G_actual = G_start : G_step : G_end do
    while (t_actual < t_simulation) do
        1. Search id_devices transmitting in t_actual;
        2. Find if there is any other device which will transmit in
         the same CH and SF;
        if (There is not any other device transmitting in same
         CH − SF) then
            success + 1;
            change device channel and program another transmission;
        else
            Check if there is collision between transmission packets in
             the same CH-SF;
            if (There is not any collided packet) then
                success + 1;
                change device channel and program another
                 transmission;
            else
                collision + 1;
                change device channel and program a retransmission;
            end
        end
    end
    Calculate PER and throughput for G_actual;
end
```

4.2 Validation

In order to verify the correct operation of the implemented LoRaWAN simulator, a mathematical model has also been included in the implementation. In particular, the mathematical model has been chosen from [26]. This model estimates the dependence between the packet error rate and the offered load, so as, it can evaluate the maximal system throughput. The model has been implemented in MATLAB. Some parameters has been modified from the model in order to adapt them to the parameters set in the simulator. For example, the probabilities of the DRs have been chosen with an uniform distribution in contrast to [26]. The principal parameters that has been configured to carry out the validation are described in Table 4.1.

Table 4.1: Parameters setting.

SFs	7-8-9-10-11-12
Frequency Channels	8 CHs of 125 kHz
Num. of end-devices	50000 (simulator) Infinite (math-model)
Environment	Ideal propagation conditions

Figure 4.4 represents the PER resulting from the simulations of the designed LoRaWAN simulator (which corresponds to the green line) as well as the mathematical model (which correspond to the red line). In light of the results obtained it can be concluded that, albeit there is a little difference owing to the mathematical model considering an infinite number of nodes. The maximum relative error is 3.98% for a network load of 150 packets/s, which validates the correct behaviour of the designed LoRaWAN simulator.

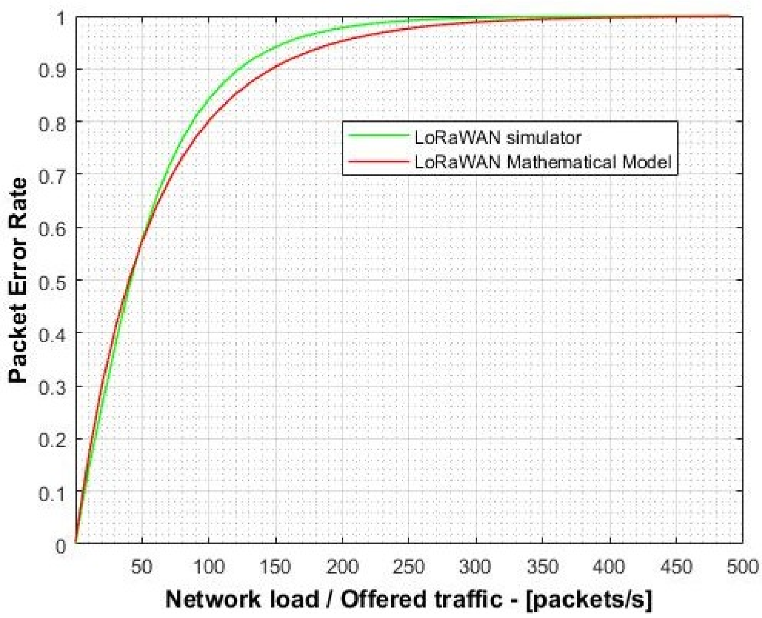


Figure 4.4: PER validation of the LoRaWAN simulator.

Chapter 5

Design and implementation of an efficient radio resources allocation algorithm for LoRaWAN

This chapter focuses on the design and implementation of an algorithm based on the efficient assignment of the LoRaWAN radio resources. In turn, this algorithm must comply with the standard requirements as must as possible while at the same time not requiring high computational complexity.

In the first section, a detail description of the design of the final algorithm that has been implemented will be given. The second section explains how this algorithm has been introduced in the LoRaWAN simulator as well as all the modifications that have been introduced in the standard LoRaWAN simulator for the correct operation of the final simulator. Finally, in the last section of this chapter, the most relevant results extracted from the simulations will be presented with a comparison with respect to the standard LoRaWAN mechanism following different configurations.

5.1 Design of the proposed algorithm

As it was mention in section 3.1, in Europe, we can find eight available carrier frequencies of 125 kHz to transmit with LoRa. Moreover, each carrier frequency is spread out into 6 SFs between SF7 (the faster and less robust one) and SF12 (the slowest and more robust one). In total we could say there are 8 frequency channels $\times$ 6 SFs = 48 available resources blocks. So it is proposed a resource allocation algorithm with the aim of used efficiently those radio resources and to avoid collisions in order to improve the network capacity.

The Pure-ALOHA medium access protocol is still maintained as detailed in the LoRaWAN specification. The end-devices are gonna used this protocol to access the medium, therefore, every time each of them has a pending packet to transmit, it will wait for a random amount of time, and, after this period of time, it will send the packet with a certain channel and SF without previously listening to the medium.

Taking advantage of the orthogonality of the SFs, the frequency channels $\times$ SFs space is divided so, each pair of values represents an independent “resource block”. So as, different end-nodes can simultaneously transmit in different resource blocks without colliding. Concretely, 48 end-devices will be able to transmit without any collision. Each resource block will be identified like RB_j, where $j \in \{1, 48\}$. Without loss of generality, it is assumed that the first 6th resource blocks utilize the first channel and SFs from 7 to 12, and so on.

Each node will divide the simulation time in windows with a fix duration of T_{window}. If node i transmits during the k-th window, that is $t \in \{t_0 + k \times T_{window}, t_0 + (k+1) \times T_{window}\}$, it will utilize a specific resource block $RB_{i,k}$. For example, if an end-device with identifier $i = 3$ transmits at $k = 5$ it means that it will utilize the resource block $RB_{3,5}$.

In order to compute which resource block (i.e., frequency channel and SF) has to be used for each transmission, that is, the $RB_{i,k}$, the gateway associated to a set of motes will be responsible for assigning to node i:

1. A set of eligible SFs, $\overline{SF_i}$.
2. The initial resource block RB_{j_i} for a reference time t_0.

$\overline{SF_i} = \{s_{i,7}, ..., s_{i,j}, ...s_{i,12}\}^T$ is a binary vector which represents the SFs available to a node i, where $s_{i,j} \in \{0, 1\}$ indicates whether this particular SF can be used or not. With these parameters, the resource block selected by node i in the k-th window (starting from t_0) is given by Equation (5.1).

$$RB_{i,k} = RB_{\widehat{j}} \mid \widehat{j} = (j_i + k) \bmod \left(8 \times \Sigma_{l=7}^{12} s_{i,l}\right), \quad (5.1)$$

where $\widehat{j} \in \{1, 8 \times \Sigma_{l=7}^{12} s_{i,l}\}$, that is, only the available SFs ($s_{i,\widehat{j}} = 1$) are considered.

The procedure for assigning resource blocks to end-devices is depicted in Figure 5.1 and is described below:

- When an end-device wants to join a LoRaWAN network, it will send a “*Join-Request*” message to the gateway.
- For each new end-device that joins the network, the gateway reserves a row of the resource block matrix (depicted in Figure 5.2). The value of this matrix in the i-th row and the k-th column is $RB_{i,k}$, that is,

the resource block that node i will use in window k, that is to say $RB_{matrix} = \begin{pmatrix} RB_{1,1} & ... & RB_{1,k} & ... & RB_{1,K} \\ & & \vdots & & \\ RB_{i,1} & ... & RB_{i,k} & ... & RB_{i,K} \\ & & \vdots & & \\ RB_{N,1} & ... & RB_{N,k} & ... & RB_{N,K} \end{pmatrix}$. The reserved row can be used by the end-device to compute the corresponding resource block for a specific window. To generate the row, the end-device only needs the initial resource block $RB_{j,i}$ and the usable SFs ($\overline{SF_i}$).

- Both parameters are calculated by the gateway in order to uniformly distribute the end-devices among the different resource blocks, and are sent in the "*Join-Accept*" message. If required, for example, due to changing the radio conditions or network load, these parameters may be sent at any time for class B and class C end-devices or in Acknowledgment (ACK) messages for class A end-devices (i.e., after transmitting a data frame). In Figure 5.1 it can be seen each of the RBs rows each end-device is assigned by the gateway. In particular, the RBs to be used by the end-devices in order to schedule their transmissions are marked in a darker colour.

- In order to avoid collisions between devices, it is used a cyclical shift between each row. As a result, two devices a and b with $RB_{j_a} \neq RB_{j_b}$ (in the case of both devices using all SFs). These mechanism can be easily seen in Figure 5.2, in which two rows with different initial values (column 1), will never have the same value in the same column. This is the reason why these operation highly reduces the number of collisions, being null if the number of devices is lower than the number of resource blocks, i.e., 48.

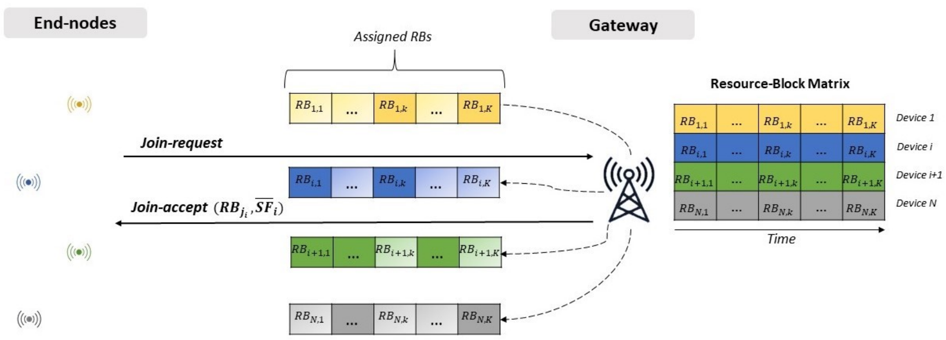


Figure 5.1: Join scheme with the assignment of resource blocks.

A flowchart which describes the computation process of the resources blocks from the end-device side is included in Figure 5.3.

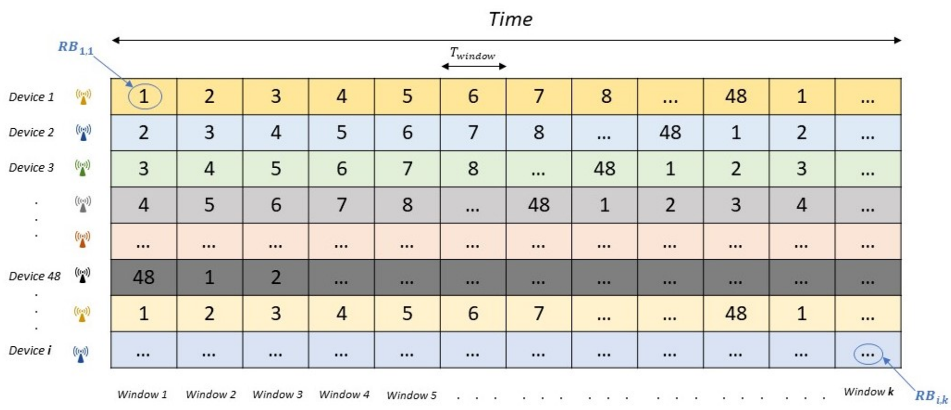


Figure 5.2: Resource block matrix RB_{matrix}.

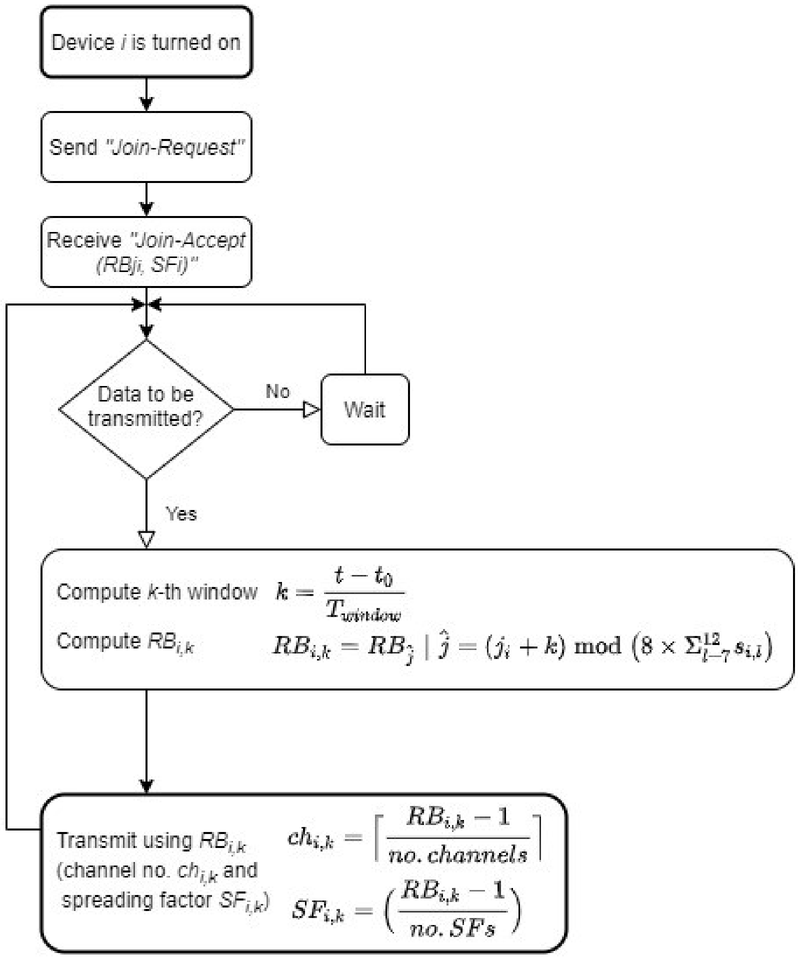


Figure 5.3: Flowchart of the resource blocks computation process from the end-device side.

Another issue has been handled in the design phase. Even if two devices transmit in different resource blocks in the same window, it may occur that one of the transmissions finishes during the next time window and produce a collision. For example, in Figure 5.4 the end-device 2 (i.e., the yellow one) may start transmitting in the first window (i.e., $RB_{2,1} = 2$) but, in this case, it will finish the transmission in the second window, which may cause a collision if the end-device 1 starts transmitting in the second window ($RB_{1,2}$). To avoid this issue the end-device, first, checks if its transmission will not exceed the duration of the transmission window. If it does, it postpones the transmission to a later random time to avoid unnecessary collisions.

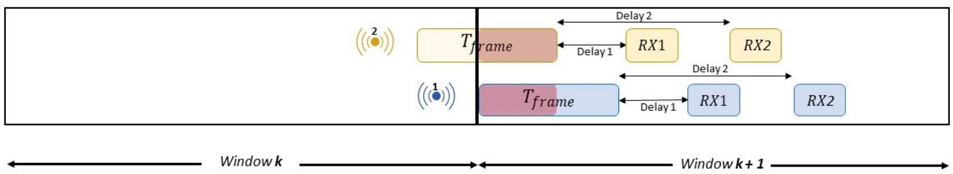


Figure 5.4: Example of border effect between time windows.

5.2 Algorithm implementation in LoRaWAN simulator

Having defined in the previous section the algorithm for the efficient allocation of resources, we now proceed to include this mechanism in the LoRaWAN simulator. In order to include this algorithm, several modifications had to be made in the simulator code in order to include the border-effect treatment and also to include the calculation of the RB matrix.

In Algorithm 2 it is briefly described a pseudocode of the new algorithm operation that has been added in the LoRaWAN simulator.

Moreover, some functions that has been included are described below:

- *calcRBmatrix.m*: This function performs the computation of the $RBmatrix$ following a cyclical algorithm. The input data of this function includes the number of end-devices, the duration of each time window, and the number of available frequency channels and SFs.

- *getCHSF.m*: This function extracts the channel and SF of the RB on which the device is currently transmitting in order to calculate, for example, the transmission time and to keeping track of other relevant data which must be saved to compute the final results. The input data of this function includes the assigned RB of an specific end-device in a given time.

- *getRBassigned.m*: In this function it is calculated the RB to be assigned to an end-device for its next transmission. The input data of this function includes the end-device ID, its instant of transmission, the RB matrix, the duration of the time window and the simulation time.

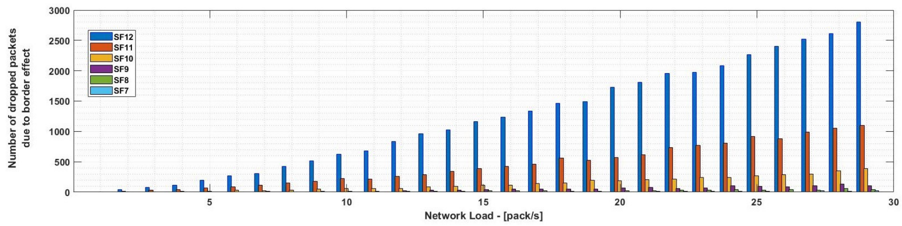
Algorithm 2: *main.m* of the LoRaWAN simulator with the designed algorithm.

```
initialization (t = 0, G = 0);
setup;
for G_actual = G_start : G_step : G_end do
    - Each device schedules its first transmission and it is assigned
      its first RB;
    while (t_actual < t_simulation) do
        - Search id_devices transmitting in t_actual;
        - Keep track of the actual used window and the next window;
        for (each "i" end-device which is gonna transmit in t_actual
          and in the current window) do
            - Keep track of the actual used RBs and the used ones in
              the next window;
            if (There is/are any end-device which will transmit in the
              same RB that end-device "i") then
                - Checks if it will be any collision with others
                  end-devices;
                if (NO collision) then
                    - Check border effect;
                    if (There is border effect) then
                        - Schedule a later random transmission;
                    else
                        - success + 1;
                        - Schedule another transmission;
                    end
                else
                    - collision + 1;
                    - Schedule another transmission;
                end
            else
                - Check border effect;
                if (There is border effect) then
                    - Schedule a later random transmission.
                else
                    - success + 1;
                    - Schedule another transmission;
                end
            end
        end
    end
    - Calculate PER and throughput for G_actual;
end
```

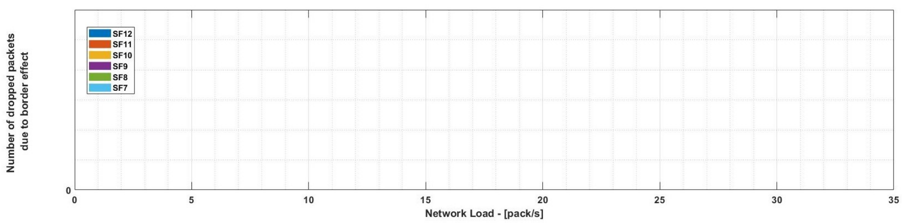
5.3 Simulations and results

This section details the different simulations that have been carried out for the evaluation of the implemented algorithm. First, in order to evaluate the effectiveness of the proposed algorithm in comparison with standard LoRaWAN, this algorithm has been executed under ideal propagation conditions. The other parameters configured in this simulation are similar to those used in the initial evaluation of the LoRaWAN simulator (see Table 4.1). Therefore, all 125kHz frequency channels and SFs available in the EU868MHz band have been used (in total 8 channels with 6 SFs each). This means that the simulation of the algorithm has been performed with all the 48 available RBs.

As a preliminary step, we first check if the design of the algorithm for avoiding border-effect collisions works. In Figure 5.5(**a**) it is depicted the number of collisions for each SF with different network loads. These collisions mainly occurs in SF12 since the higher ToA of the transmissions increases the probability of collisions and the probability of exceeding the time window of the transmission. For example, in a simulation with a network load of 30 packets/s, 51222 packets were sent and only 2366 collided (resulting a PER = 4.6%). The number of packets that were transmitted with SF12 was 1485. The simulator was included the proposed algorithm for avoiding border-effect collisions, so as, in Figure 5.5(**b**) there are not collisions at all.



(**a**)

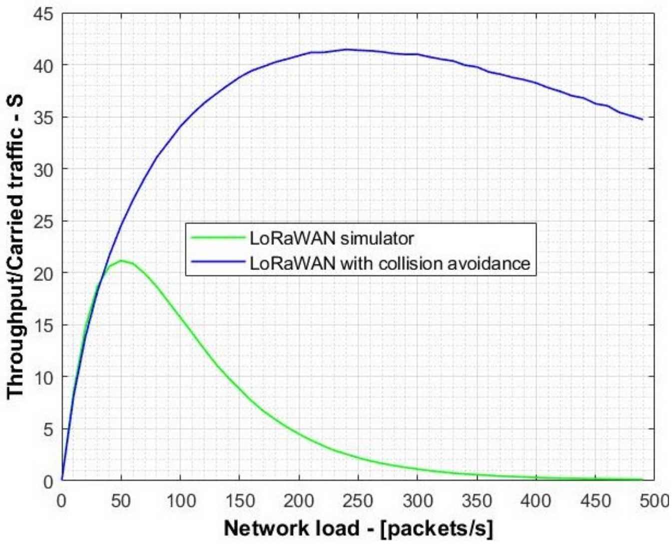


(**b**)

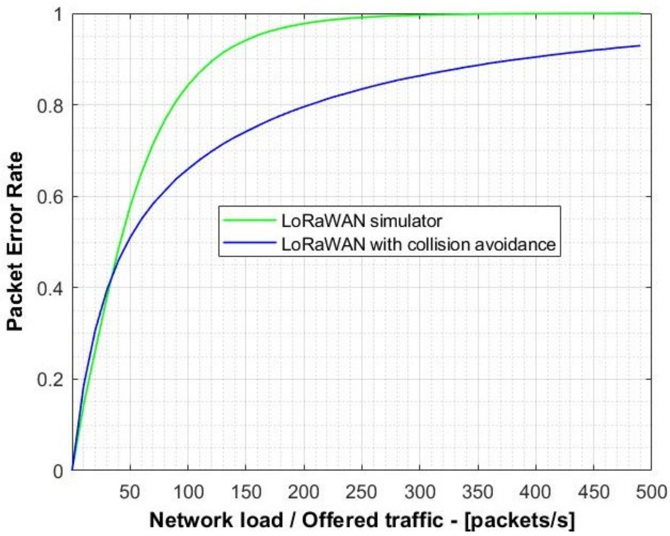
Figure 5.5: Collisions (a) with no border effect check and (b) avoiding the border effect.

The simulation result of the implemented algorithm under the above mentioned conditions, in terms of throughput and PER is shown in Figures 5.8(**a**) 5.8(**b**). In addition, the result of the simulation with standard LoRaWAN is shown in order to visualize a comparison between the two graphs.

As shown in the graphs, the proposed algorithm outperforms the standard LoRaWAN achieving a maximum PER gain of 30.1% at a network load of 150 packets/s (being a PER of 71.4% in the proposed algorithm and 92.9% in standard LoRaWAN). Since this first set of simulations assumes ideal radio conditions, the packet errors are due to the collisions meaning that this solution significantly reduces the collision probability. This PER reduction allows the system to increase the maximum capacity (system throughput) from 21 packets/s, producing a relative gain of 95.2%.



(**a**) Throughput comparison.



(**b**) PER comparison.

Figure 5.6: Throughput and PER comparison between standard LoRaWAN and the new LoRaWAN MAC with collision avoidance.

Secondly, we carried out another simulations but now taking into account realistic radio conditions. For that purpose, we apply this novel RB allocation scheme in a urban realistic scenario. For the implementation of the propagation model we have taken into accounting the "Macro Cell Propagation Model" based on [27].

The signal power that the gateway receives from each end-device is calculated with expression 5.2, where:

- S_{TX} is the signal power that and end-device transmits (usually 14 dBm for LoRaWAN end-devices).
- G_{TX} and G_{RX} refer to the transmitter and receiver antenna gains, respectively.
- L_{prop} ($L_{propagation}$) represents the urban environment path loss assuming that all the building heights are similar and it is calculated according to expression 5.3. Dhb and R are the gateway height in meters, measured from the average rooftop level and the separation between the gateway and the end-device in kilometers, respectively. f is the carrier frequency in MHz.
- X_σ refers to a log-normally distributed shadowing with a standard deviation of 10dB.

$$S_{RX}(dBm) = S_{TX}(dBm) + G_{TX}(dBi) + G_{RX}(dBi) - L_{prop}(dB) - X_\sigma(dB) \tag{5.2}$$

$$L_{prop} = 40 \cdot (1 - 4 \cdot 10^{-3} \cdot Dhb) \cdot log_{10}(R) - 18 \cdot log_{10}(Dhb) + 21 \cdot log_{10}(f) + 80dB \tag{5.3}$$

The parameters setting for the simulation with the proposed propagation model has been chosen taking into consideration the work in [28]. f will be the carrier frequency 868MHz. Dhb is set as 15 meters and R is calculated for each pair of gateway – end-device. The signal power of the end-device is set as 14dBm and both transmitter and receiver antenna gains are set as 2dBi.

The considered scenario, depicted in Figure 5.7 consists of a gateway located in the origin of coordinates(red point). End-devices (blue points) have been positioned in a randomly manner following an uniform distribution around the aforementioned gateway and they are also fixed.

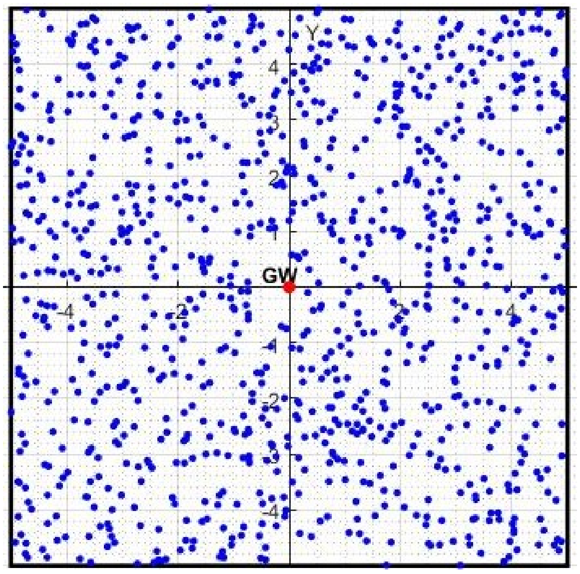


Figure 5.7: Realistic scenario simulation.

The allocation of the RBs is carried out taking into consideration the information in Table 5.1. According to the strength of the signal received by the gateway, it will assign a more robust SF such as SF10, SF11 or SF12, when the signal is weak due to remoteness of the end-device or maybe due to the shadowing caused e.g. by the buildings. If the signal strength that the gateway receives from the end-devices is strong, it will be assigned a less robust SF (which are the faster ones for transmission time), such as SF7, SF8 or SF9.

Table 5.1: Gateway Sensitivity for each SF with a 125kHz BW [29].

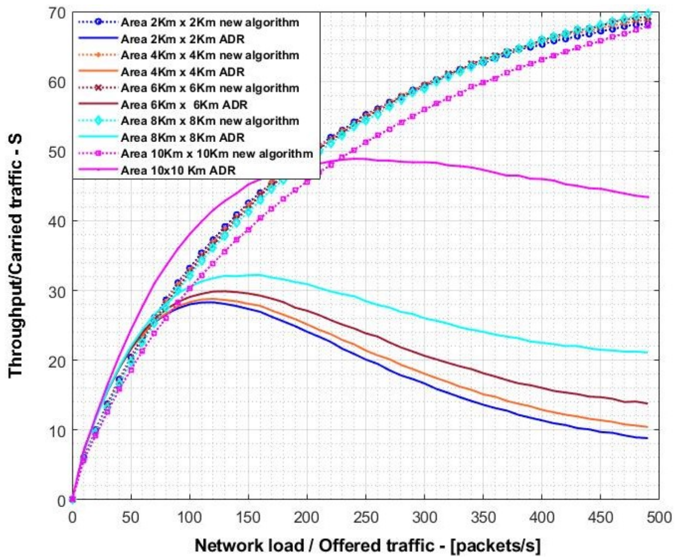
SF	**Gateway Sensitivity (dBm) [29].**
12	-142.5
11	-140
10	-137.5
9	-135
8	-132.5
7	-130

In order to carry out several simulations with the configured scenario, it has been decided to deploy a network of 1000 end-devices. Moreover, the coverage area has been changed from smaller to longer areas in order to understand the effect of the coverage area on the performance and SF usage distribution. Concretely, two campaigns of simulations have been performed. In the first campaign, the SF assignment have been carried out according to the ADR algorithm proposed in the LoRaWAN standard which was explained in Subsection 3.2.2. Since it has been assumed fixed end-devices, the SF assigned to each end-device depends on the received sensitivity. In the second campaign, end-devices use the SFs in which they can transmit. For example, if a device can transmit in SF9, it will hop between SF9, SF10, SF11 and SF12. It has been differentiate results for "fast end-devices" (those whose lower eligible SF is SF7, SF8 or SF9) and "slow end-devices" (those whose lower eligible SF is SF10, SF11 or SF12). Table 5.2 shows the simulated areas (from 2 Km x 2 Km up to 10 Km x 10 Km) and the distribution of "fast devices" and "slow devices" according to the propagation model and the gateway sensitivity (see Table 5.1). As expected, a smaller area has a higher percentage of "fast devices" since they are closer to the gateway and therefore the radio conditions are better. On the contrary, a larger area means that the percentage of "slow devices" increases due to the same reasoning.

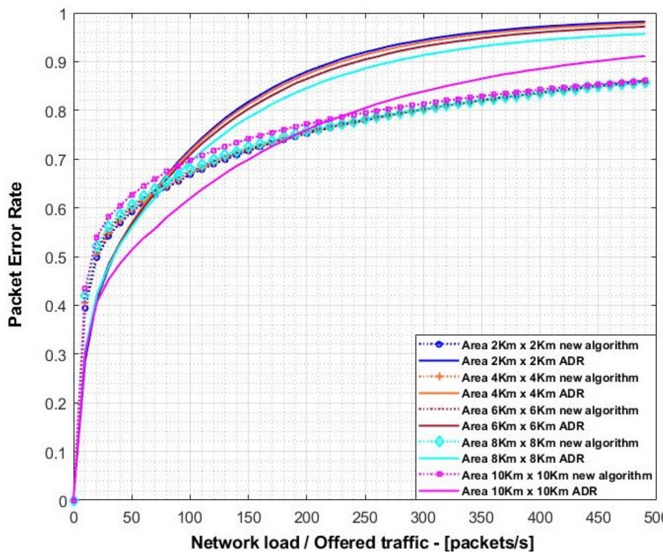
Table 5.2: Number of devices transmitting in each group.

Area	**Number of "fast" devices**	**Number of "slow" devices**
2 x 2 Km^2	658	342
4 x 4 Km^2	646	354
6 x 6 Km^2	632	368
8 x 8 Km^2	623	377
10 x 10 Km^2	562	438

Figures 5.8(**a**) and 5.8(**b**) represent the mean system throughput and PER for each round of simulations using coverage areas between 2 x 2 Km^2 to 10 x 10 Km^2. It can be seen that, for areas between 2 x 2 Km^2 and 8 x 8 Km^2, the collision avoidance algorithm in this realistic scenario decrements the PER, achieving a maximum throughput up to almost 40% higher that the maximum achieved with ADR. However, with an area of 10 x 10 Km^2 it is observed a reduction of the average throughput for medium/low network loads using the collision avoidance algorithm. As the coverage area grows, the number of "slow devices" also increases. This is common for both the collision avoidance algorithm and standard LoRaWAN. However, using the collision avoidance algorithm, "fast devices" also utilize slow SFs part of the time. This produces a concentration of network load being carried over slow SFs, causing a higher number of collisions. This effect is more noticeable with larger areas due to the SFs distribution. On the contrary, when the coverage area is smaller, there are more end-devices using cyclically all SFs. In this case, the SF distribution is more uniform and, thus, this effect is negligible and the improvement for using the collision avoidance algorithm is higher.



(**a**) Throughput comparison.



(**b**) PER comparison.

Figure 5.8: Throughput and PER for 1000 devices in the case of non groups of SF with different area dimensions. (**a**) Mean throughput of the system for ADR and collision avoidance algorithm cases. (**b**) Packet Error Rate of the system in ADR and collision avoidance algorithm cases.

In light of the obtained results some conclusions can be highlighted:

- With the collision avoidance algorithm it is obtained a great overall network throughput up to an area of 8 x 8 Km^2 with 1000 end-devices deployed on it.

- For longer areas, due to the less proximity of the end-devices to the gateway, the slowest and more robust data rates are more frequently used and the probability of collisions increases. On the contrary, with the same number of end-devices but smaller areas, the collision avoidance algorithm contributes to the increment of throughput and presents a great improvement compared to the throughput obtained with the ADR algorithm.

- Although this solution slightly increases the overall transmission time and may lead to a little increase in energy consumption compared to the ADR algorithm proposed in the LoRaWAN specification, it achieves a fairer resource sharing. For example, in a coverage area of 2 x 2 Km^2, the ADR algorithm assigns shortest SFs due to the proximity of the end-devices to the gateway, thus allowing shortest transmission time. However, this lead to more collisions due to the saturation of the fastest RBs. With the collision avoidance algorithm proposed here, owing to the set of SFs each end-device is able to use, it leads to a more uniform RBs utilization, causing less collisions and therefore improving the overall network throughput.

Chapter 6

Proof of concept

This chapter describes the process carried out for the design and implementation of a new server included in a general LoRaWAN architecture that will be in charge of handling the necessary signaling messages between the different entities of the network, as well as the logic necessary to implement the new designed resource allocation algorithm in the LoRaWAN nodes. Section 6.1 details the design of the testbed architecture to be carried out as well as its implementation. Section 6.2 will show the actual testbed implemented and the tests performed as a proof of concept.

6.1 Design and implementation of the testbed

The testbed architecture that has been designed and implemented is shown in Figure 6.1.

The rectangular area surrounded by green corresponds to the LoRaWAN radio access network. It consists on several end-devices connected to a LoRaWAN gateway. The end-devices are based on the Pycom's FiPy development board [6], which is a multi-technology board with a ESP32 SoC that supports Wi-Fi, BLE, LoRaWAN, Sigfox and NB-IoT/ LTE-M. The LoRaWAN gateway correspond to a IMST Lite-Gateway [5], which contains a Raspberry Pi connected to an IMST ic880A LoRa concentrator for the ISM 868MHz band.

The blue area contains all the components involved in a LoRaWAN core network. This entire block of LoRaWAN entities is implemented with the Chirpstack platform [30], an open source project which provides open-source components for the gateway bridge, network server, application server among others for the LoRaWAN networks.

On the top of the both mentioned LoRaWAN entities it has been included a new server called CARA server (red area), which will be responsible for executing the designed resource allocation algorithm for the new end-devices that join the network.

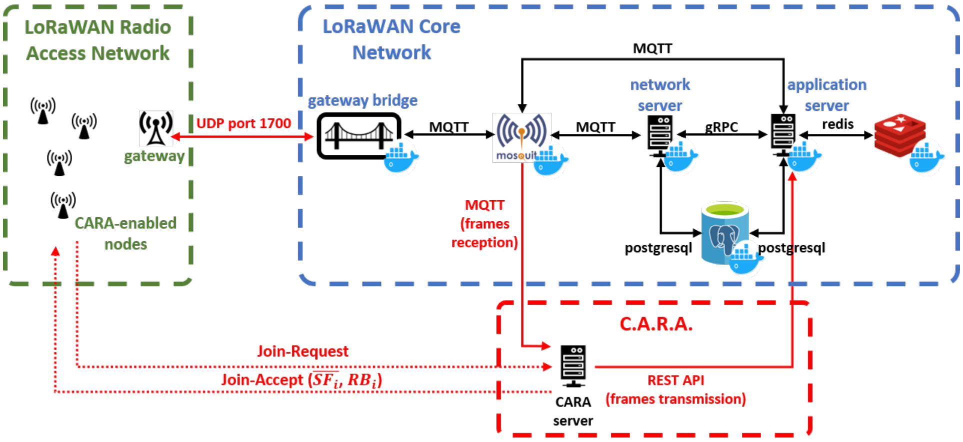


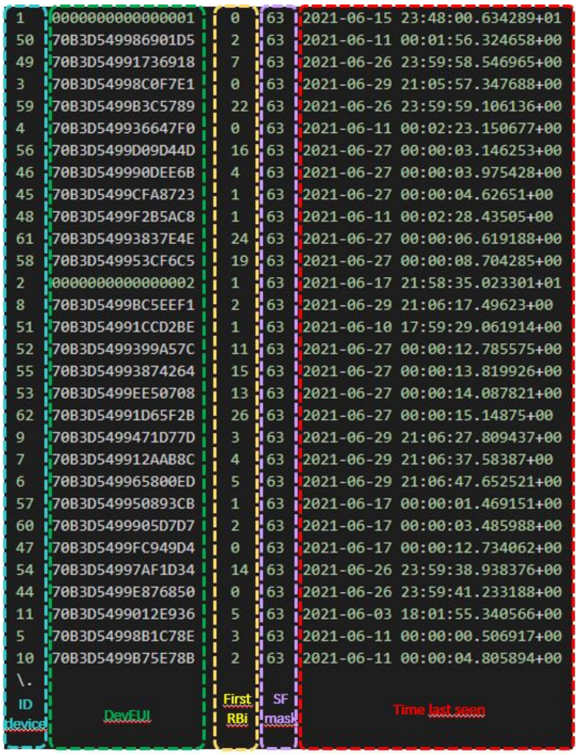
Figure 6.1: Testbed architecture designed and implemented for the proof of concept.

Regarding the CARA server, it is programmed using Pyhton 3 language, and implements the following:

- A **Message Queuing Telemetry Transport (MQTT) listener (subscriber)** for messages from the application server in order to allow the CARA server to receive signaling messages from nodes, i.e. the "Join Request" messages.
- A **signaling handler** which is in charge of generating "Join Accept" messages as a response for the received "Join Request" frames. For that purpose, it uses the Chirpstack network server RESTful API [31]. This handler communicates with the resource allocation logic to obtain the available spreading factors ($\overline{SF_i}$) and the initial resource block (RB_i) that each end-device can use. Both obtained parameters are included in the "Join Accept" response.
- A **database client** to store and retrieve the current status of the collision avoidance algorithm from a PostgreSQL database. In this database it is stored, for each node, its identifier ($DevEUI$), the initial resource block RB_i, the SFs that each end-device has available (in the form of a 6-bit binary SF mask, in which '1' means that the SF can be used and '0' otherwise), and the time when the end-device was last seen (in order to remove the end-device from the algorithm calculations after a configurable time without receiving data from that node). An example of how this information looks like in the database is shown in Figure 6.2.
- The **logic for the resource allocation algorithm**, that checks which is the resource block with lowest assignments to be employed for the next end-device, considering which SFs are available.

Concerning to the end-devices, these are programmable using MicroPython with the Pycom's LoRa API [32]. The end-devices' firmware has been also modified in order to include the signaling handling and the resource block selection for each transmission. Since we do not have access to the source code of the MAC layer in the end-devices, we implemented the required signaling at the application layer. In particular, the functionalities that have been included and used are the following:

- In order to include *DeviceTime* MAC commands for end-devices synchronization, since there is not access to the source code of the MAC layer, this synchronization has been achieved using the Network Time Protocol (NTP) protocol through a Wi-Fi connection.
- The **experiment parameters** are also sent over Wi-Fi to customize the testbed end-devices.
- End-devices join the network using OTAA activation. ABP has also been successfully tested.
- The new addition signaling for the resource allocation has been implemented on the end-device's side. As soon as the end-device joins the network, it send a "Join Request" message to the application server. After that, it waits the reception of a "Join Accept" message with the initial RB_i and the available SFs. Both messages are sent as application frames. If a "Join Accept" message is not received after a configurable time, the "Join Request" is retransmitted.
- After the initial signaling explained above, it is assumed a random time between packets for the traffic generation. This time is composed of a fixed part (to avoid sending messages too fast) and a random part (to avoid correlation between transmissions from different end-devices).
- For transmission, a **border effect check** has been implemented based on the time to initiate transmission and the ToA calculation. If the transmission ended on the next time window, it would be postponed until the beginning of that window.



ID device	DevEUI	First RBi	SF mask	Time last seen
1	0000000000000001	0	63	2021-06-15 23:48:00.634289+01
50	70B3D549986901D5	2	63	2021-06-11 00:01:56.324658+00
49	70B3D54991736918	7	63	2021-06-26 23:59:58.546965+00
3	70B3D54998C0F7E1	0	63	2021-06-29 21:05:57.347688+00
59	70B3D5499B3C5789	22	63	2021-06-26 23:59:59.106136+00
4	70B3D549936647F0	0	63	2021-06-11 00:02:23.150677+00
56	70B3D5499D09D44D	16	63	2021-06-27 00:00:03.146253+00
46	70B3D549990DEE6B	4	63	2021-06-27 00:00:03.975428+00
45	70B3D5499CFA8723	1	63	2021-06-27 00:00:04.62651+00
48	70B3D5499F2B5AC8	1	63	2021-06-11 00:02:28.43505+00
61	70B3D54993837E4E	24	63	2021-06-27 00:00:06.619188+00
58	70B3D549953CF6C5	19	63	2021-06-27 00:00:08.704285+00
2	0000000000000002	1	63	2021-06-17 21:58:35.023301+01
8	70B3D5499BC5EEF1	2	63	2021-06-29 21:06:17.49623+00
51	70B3D54991CCD2BE	1	63	2021-06-10 17:59:29.061914+00
52	70B3D5499399A57C	11	63	2021-06-27 00:00:12.785575+00
55	70B3D54993874264	15	63	2021-06-27 00:00:13.819926+00
53	70B3D5499EE50708	13	63	2021-06-27 00:00:14.087821+00
62	70B3D54991D65F2B	26	63	2021-06-27 00:00:15.14875+00
9	70B3D5499471D77D	3	63	2021-06-29 21:06:27.809437+00
7	70B3D549912AAB8C	4	63	2021-06-29 21:06:37.58387+00
6	70B3D549965800ED	5	63	2021-06-29 21:06:47.652521+00
57	70B3D549950893CB	1	63	2021-06-17 00:00:01.469151+00
60	70B3D5499905D7D7	2	63	2021-06-17 00:00:03.485988+00
47	70B3D5499FC949D4	0	63	2021-06-17 00:00:12.734062+00
54	70B3D54997AF1D34	14	63	2021-06-26 23:59:38.938376+00
44	70B3D5499E876850	0	63	2021-06-26 23:59:41.233188+00
11	70B3D5499012E936	5	63	2021-06-03 18:01:55.340566+00
5	70B3D54998B1C78E	3	63	2021-06-11 00:00:00.506917+00
10	70B3D5499B75E78B	2	63	2021-06-11 00:00:04.805894+00
\.				

Figure 6.2: End-device information stored in the new server database.

Figure 6.3 depicts a flow diagram which describes the exchanged messages between the main blocks involved in the testbed network architecture in order to understand what type of messages and parameters each entity sends to each other for the initial resource allocation.

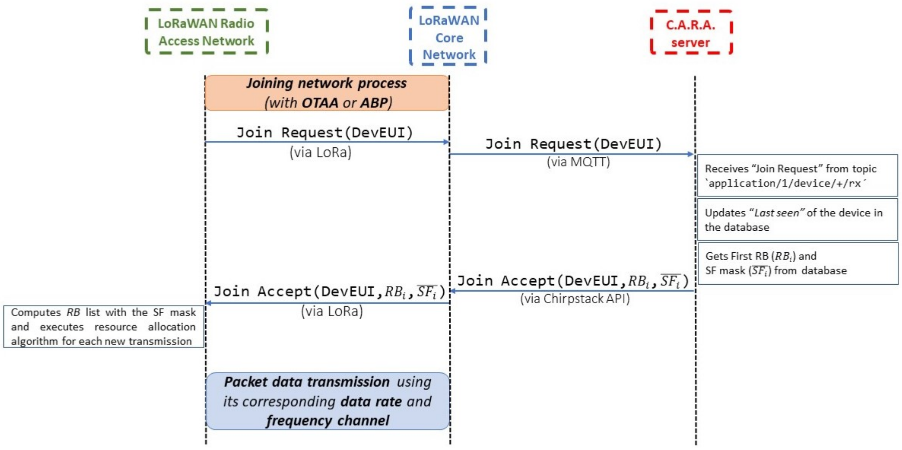


Figure 6.3: Flow diagram of the messages exchanged between the network entities for the resource allocation process.

6.2 Testing and preliminary results

The real testbed that has been set up for the proof of concept is shown in Figure 6.4. The LoRaWAN radio access network is composed of 30 end-devices and one gateway. Then, the LoRaWAN core network (coming from the Chirpstack platform) and the CARA network are implemented on the same server (in a virtual machine of the PC from the Figure 6.4). In the PC it is also launched the Chirpstack platform web interface for monitoring all the network functioning and also a pycom console for programming and uploading the firmware of the end-devices.

Behind the real testbed architecture it is depicted a log from the CARA server. It includes and example of the messages exchanged between one end-device and the CARA server. Firstly, the CARA server receives one "*#JOINREQ#*" message ("*Join Request*"), computes the initial resource block to be assigned to that end-device, and sends back this value in the "*#JOINACC#*" (*Join Accept*) message. In addition, an SF mask is also sent (in this case the SF mask is 63), which indicates which SFs are available to the end-device (one bit per SF, so 63 = '111111', means that all the six SFs are available).

In this experiment it has been assumed that all the SFs are available to all end-devices. Data transmissions from each end-device are sent after $10 + rand(10)$ s, where the $rand(x)$ function returns a random value between 0 and x. This produces a network load of 6000 packets/s. As in the simulations explained in chapters 4 and 5, data frames contain 24 bytes. It has been verified that, since the number of devices is smaller than the number of resource blocks, each end-device is granted a different initial resource block and therefore no collisions occur. Figure 6.5 shows an example of RBs allocation to the end-devices in this testbed. Only 5 end-devices are shown for the sake of clarity. As expected, the end-devices do not collide (residual collision probability lower than 0.5% due to other LoRaWAN devices in the surroundings) and the RBs are allocated sequentially to each node, following a ramp pattern. In addition to the CARA server log, it has been also checked the correct behaviour of the solution analyzing the traces from the node (in the Pycom console) and the frames received by the Chirpstack application server.

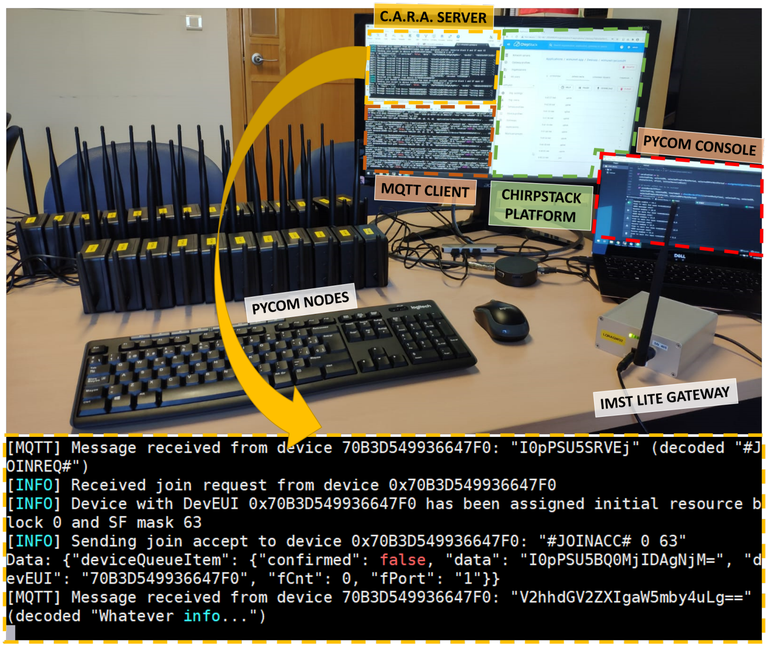


Figure 6.4: Testbed with a CARA server log.

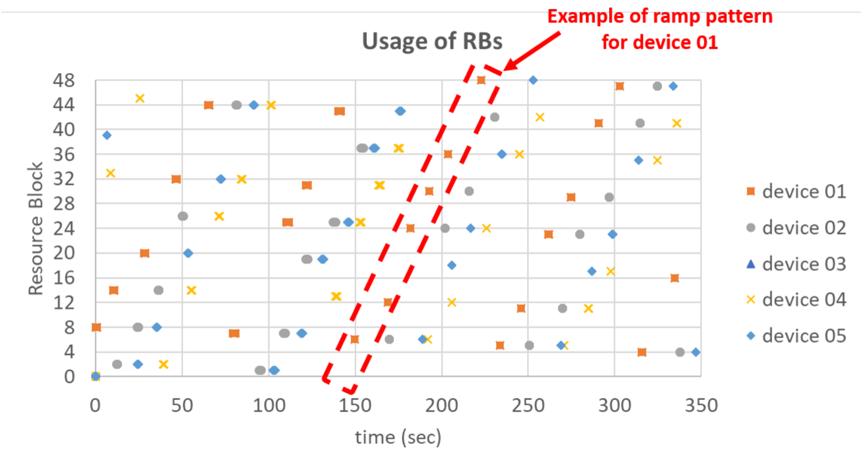


Figure 6.5: RB usage in the proof of concept.

Chapter 7

Conclusions and future work

The last chapter of this master´s thesis document details the main conclusions and future work of this project. Section 7.1 describes the main findings as well as the possible limitations of the proposed solution. Finally, section 7.2 remarks some improvements and future work for this project enhancements.

7.1 Conclusions

In this project, a standard LoRaWAN simulator has been designed and developed. Moreover, a collision avoidance resource allocation algorithm that runs on top of this simulator has been proposed for increasing the capacity of LoRaWAN networks.

This solution proposes the division of the wireless medium capacity into independent resources blocks which are defined by one frequency channel and one SF. Thanks to the SFs orthogonality feature in LoRa, transmissions in different resource blocks will not cause collisions. In addition, the algorithm leverages the existing joining procedure for parameters exchange and synchronization, not requiring communication between the end-devices and the network afterwards. The algorithm has been implemented in a MATLAB-based dynamic simulator, using a cyclical assignment which ensures that the end-devices with a different assigned initial resource block will not collide.

The simulation results show that this solutions outperforms the standard LoRaWAN networks achieving a 95.2% of capacity gain with ideal radio conditions. In realistic propagation environments, this solution increases the capacity by 40%. Finally, a proof-of-concept has been performed using commercial equipment, in order to check the feasibility and the proper operation of the proposed solution.

The contribution extracted from this project has been published in a Journal Citation Report (JCR) journal. More precisely, the publication is titled **"Collision Avoidance Resource Allocation for LoRaWAN"** [33].

Some remarkable findings and limitations are highlighted as follows:

- This solution fits with fixed end-devices scenarios. When mobility is included, the radio conditions change very fast and it is necessary a more sophisticated algorithm to take into account other dynamic parameters of the environment.

- The simulation's results show that this algorithm matches in urban scenarios up to areas of 10 Km x 10 Km. In biggest areas this algorithm does not contribute in an enhancement of the overall network capacity.

- A noteworthy drawback is the simulation time of the LoRaWAN simulator. In the simulations in which 50000 devices where deployed in the network, the simulation took more than 3 days to finish. This may have happened because the simulator was designed to work in a temporal scale of milliseconds and with a dense scenario of end-devices the simulation time is slowed down.

- Due to some hardware limitations, the proof of concept has been performed in an indoor scenario and it has not been possible to test this solution with real equipment in an outdoor environment.

- In addition, this solution did not require many modifications in the firmware of the end-devices as well as in the software of the core network. This means that it is a computational cost-efficient solution.

7.2 Future work

In this work we have design and developed a collision avoidance resource allocation algorithm for the LoRaWAN MAC layer, which runs on a LoRaWAN simulator also designed and developed in this project. Nonetheless, several challenges lie ahead:

- The study of others propagation models is an interesting challenge in order to assess the feasibility of this solution in different environments. For example, for rural areas, it is expected an improvement of the capacity due to fewer obstacles, which traduce in less collisions even in areas of more than 10 Km x 10 Km.

- The insertion of a LoRaWAN mobility scenario could be an ambitious future work in order to study resource allocation, energy and security aspects for this kind of networks.

- As commented in the conclusions, the simulation time may get too high. Another aspect to be addressed as future work could be the reduction of the simulator complexity.

- Future work will be also focused on the energy consumption study in order to determine whether, despite the fact that this solution can increase energy consumption very insignificantly, it can be labelled as an energy aware solution.

Bibliography

[1] IoT Analytics, *State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time*. Available online: `https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/` (last access 01/11/2020).

[2] IoT Analytics, *Top IoT Applications in 2020*. Available online: `https://iot-analytics.com/top-10-iot-applications-in-2020/` (last access 01/11/2020).

[3] LoRa Alliance. Available online: `https://lora-alliance.org/` (last access 01/11/2020).

[4] Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melia-Segui, J.; Watteyne, T. "Understanding the limits of LoRaWAN". *IEEE Commun. Mag.*, 2017, 55, 34–40.

[5] IMST, *Lite Gateway - Demonstration Platform for LoRa® Technology*. Available online: https://shop.imst.de/wireless-modules/lora-products/36/lite-gateway-demonstration-platform-for-lora- technology (last access 24/02/2021).

[6] Pycom, *Fipy development board*. Available online: https://pycom.io/product/fipy/ (last access 24/02/2021).

[7] Polonelli, T., Brunnelli, D., Marzocchi, A., & Benini, L. "Slotted aloha on lorawan-design, analysis, and deployment". *Sensors*, 2019, 19(4), 838.

[8] Leonardi, L., Battaglia, F., Patti, G., & Bello, L. L. (2018, October). "Industrial LoRa: A novel medium access strategy for LoRa in industry 4.0 applications". *In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, October, 4141-4146.

[9] Haxhibeqiri, J., Moerman, I., & Hoebeke, J. "Low overhead scheduling of lora transmissions for improved scalability". *IEEE Internet of Things Journal*, 2018, 6(2), 3097-3109.

[10] Zorbas, D., Abdelfadeel, K., Kotzanikolaou, P., & Pesch, D. "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things". *Computer Communications*, 2020, 153, 1-10.

[11] Piyare, R., Murphy, A. L., Magno, M., & Benini, L. "On-demand LoRa: Asynchronous TDMA for energy efficient and low latency communication in IoT". *Sensors*, 2018, 18(11), 3718.

[12] F. Cuomo et al., "Towards traffic-oriented spreading factor allocations in LoRaWAN systems". In*17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, IEEE, Capri, 2018, 1-8.

[13] Triantafyllou, A., Sarigiannidis, P., Lagkas, T., Moscholios, I. D., & Sarigiannidis, A. "Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance". *Computer Networks*, 2020, 107735.

[14] LoRa Alliance, LoRaWAN® Specification v1.1. Available online: `https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/` (last access 01/11/2020).

[15] Laurens Slats, A Brief History of LoRa: Three inventors share their personal story at thing conference, 8 Jan 2020. Available online: `https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference` (last access 15/11/2020).

[16] Seller, O.B. & Sornin, N. "Low Power Long Range Transmitter". US Patent 9,252,834, 2 February 2016.

[17] Semtech. LoRa™ Modulation Basics. Application Note AN1200.22, Tech. Rep., 2015.

[18] Semtech. SX1272/3/6/7/8: LoRa Modem Design Guide. Application Note AN1200.13, 2013; pp.5–6.

[19] LoRa Alliance, RP002-1.0.1 LoRaWAN® Regional Parameters. Available online: `https://lora-alliance.org/resource_hub/rp2-102-lorawan-regional-parameters/` (last access 15/11/2020).

[20] ETSI, ETSI EN 300 220-2, "Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment", V3.2.1, April 2018.

[21] Technical Marketing Workgroup 1.0. LoRaWAN^{TM}, What is it? A technical overview of LoRa and LoRaWAN. *LoRa Alliance*, 2015, 11. Available online: `https://lora-alliance.org/resource-hub/what-lorawanr` (last access 15/11/2020).

[22] Semtech. LoRaWAN – simple rate adaptation recommended algorithm. Common to Class A/B/C specification. Preliminary, 2016.

[23] Sundaram, J.P.S.; Du, W.; Zhao, Z. A "Survey on LoRa Networking: Research Problems, Current Solutions and Open Issues". *IEEE Commun. Surv. Tutorials*, 2019, 22, 371–388.

[24] Abramson, N. (1970, November). "The ALOHA system: Another alternative for computer communications". In *Proceedings of the November 17-19, 1970, fall joint computer conference* (pp. 281-285).

[25] Kleinrock, L., & Tobagi, F. (1975). "Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics". *IEEE transactions on Communications*, 23(12), 1400-1416.

[26] Bankov, D.; Khorov, E.; Lyakhov, A. "Mathematical model of LoRaWAN channel access". In *Proceedings of the IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Macau, China, 12–15 June 2017; pp. 1–3.

[27] ETSI, TR. 136 942,"LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios (3GPP TR 36.942 version 16.0.0 Release 16)". Technical Report Version 16.0. 0, 2020.

[28] Farhad, A., Kim, D. H., & Pyun, J. Y. "Scalability of LoRaWAN in an Urban Environment: A Simulation Study". *In 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, IEEE, 677-681.

[29] Semtech Corportation. Understanding the LoRa® Adaptative Data Rate. Technical Paper, 2019.

[30] Chirpstack, *Open-source LoRaWAN® Network Server stack*. Available online: `https://www.chirpstack.io/` (last access 01/04/2021).

[31] ChirpStack, *Open-source LoRaWAN® Network Server API interface.* Available online: `https://www.chirpstack.io/application-server/api/` (last access 01/04/2021).

[32] Pycom, *LoRa API documentation.* Available online: `https://docs.pycom.io/firmwareapi/pycom/network/lora/` (last access 01/04/2021).

[33] Chinchilla-Romero, N., Navarro-Ortiz, J., Muñoz, P., Ameigeiras, P. "Collision Avoidance Resource Allocation for LoRaWAN". *Sensors*, 2021, 21(4), 1218.