Empirical study of LoRaWan on the study case of Genova

Pablo Salinas, Javier Pardiño, Iñigo García

Public University of Navarre & University of Genova

Supervisor

Daniele Caviglia

In the bachelor's thesis for the

Degree of Telecomunications Engineering; international variant

June 28, 2022

Abstract

This thesis aims to:

- Mesure performance LoRaWan comunication standard on packed cities and with moving end nodes
- Many different scenarios were tested and analized
- The conclusions reached prove that the asumptions made were sound

Contents

List of Figures

List of Tables

Introduction

0.1 Motivations

In recent years, Internet Of Things technologies (IoT) have grown in importance and are expected to have an exponential increase in their use in the coming years. For this reason, there is a need for communication protocols with two key features, which are: - Long range communications - Low power consumption. To cover these needs LPWANs (Low Power Wide Area Networks) were created and offer big advantages when compared to high bitrate and shorter range technologies such as Wi-Fi or Bluetooth, and cellular ones like GSM or 4G that have a much higher power consumption.
The study of these technologies on the enviroment of tightly packed cities is highly important in order to advance to achive smart cities.

0.2 Context of the Study

The study is performed on the city of Genova, thightly packed with buildings, for the most part made with concrete and brick. Other cities with different architecture could perform different as the distribution of the buildings is important in the reach of the signal. One example of this is the problems with GPS localization when near skyscrapers.

0.3 Tools used in the thesis

For the storage and analysis of these messages The Things Network is being used, which is a free to use LoRa network that offers not only gateways but also data storage and various functionalities for a particular network, being able to make it private or public.

0.4 Overview of the Thesis

In this thesis we are going to take a look at LoRaWAN (Long Range Wide Area Network), a communication protocol with a low bit rate and power consumption that can reach distances of up to several kilometres, we will study the characteristics of the communication and recreate various real life scenarios to study different characteristics such as packet loss, power transmitted and the best configuration for the end- devices. With these measurements we can obtain valuable information about how to optimize the network for its different use cases, as there are big differences depending on the topology and placements of the nodes and gateways.

Chapter 1

Previous required knowledge

1.1 LPWAN

LoRa and LoRaWAN protocols are based on the LPWAN technology. LPWAN stands for "Low Power Wide Area Network" and it allows radio- equipped devices to communicate more efficiently and over longer distances. This emergent technology had the goal of finding a better data transmission protocol than other technologies such as Bluetooth, Wi-Fi or 3G/4G. The low-power consumption and the long range transmission that this technology offers, positions it as a cost-effective solution when compared with the other big technologies. As LPWAN is not concerned with transmitting big quantities of data, it offers aspects that simply others can't complete such as the coverage of long distances as it can be appreciated in the figure below.

This is why LPWAN has become one of the fastest growing spaces in the Internet of Things (IoT) ecosystem. One of the biggest and most interesting applications of this data transmission technology is on the Smart Cities field. This protocols are a perfect solution for the Smart City data networks, which requires low data rate but constant data streams and the coverage of long distances with

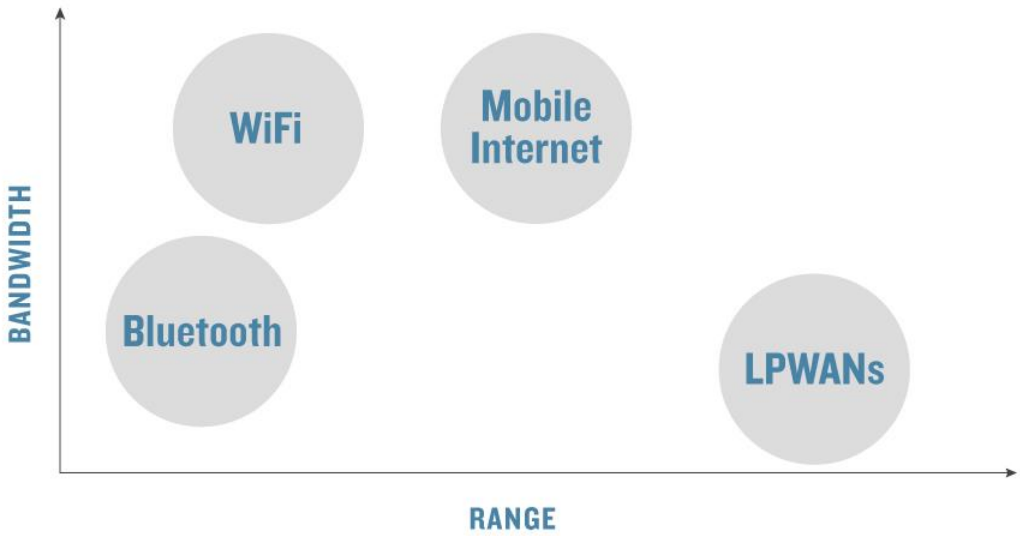

Figure 1.1: Graph comparison of range and bandwidth for different data transmission technologies.

the benefit of using significantly less power than other data network platforms. Shanghai is a perfect example of a Smart City where LPWAN is used in order to connect thousands of sensors to measure the air quality and human movement, more info here. The future of this technology is bright as IoT devices are growing a 12% annually according to a IHS Markit report. and LPWAN is one of the greatest spaces in the IoT ecosystem. LoRa and LoRaWAN are realizations of the Low Power WAN concept, and with such high demand for the technology both are having great impulse in their development.

1.2 LoRa

LoRa is the protocol which defines the physical (radio) layer of the communications in The things Network (TTN). This protocol fixes the frequency band and defines the modulation that is used to transmit data which enables the long-range

communication link.
The frequency band depends on the region in which you are, for example, in the US they use a band from 902.3 to 914.9 MHz, here in Europe the band goes from 863 to 870 MHz as it's ilustrated on the next table:

Frequency band depending on the region	
Region	Frequency (Mhz)
Asia	433
Europe	863-870
US	902-928
Australia	915-918
Canada	779-787
China	470-510

Table 1.1: frequency band of different regions

The LoRa modulation is based on the Spread Spectrum technique and it's crucial for the long distance communication. This technique spreads the signal all over the frequency band and it allows LoRa to be very robust to interferences and noise, allowing the signal to reach longer distances. More precisely it's called chirp modulation and it transmits symbols encoding them into multiple signals of increasing (upchirp) or decreasing (downchirp) radio frequencies. A LoRa transmission always starts with 8 upchirps and 2 downchirps as we can see in the figure number 1.2. In LoRa communications we are able to reduce our data rate in order to gain range by changing the Spreading Factor. The Spreading Factor

Figure 1.2: Snapshot of a LoRa transmission

means the number of raw bits that can be encoded in a symbol. ("Modelling and Performance Evaluation of LoRa Network Based on Capture ...") The Spreading Factor (SF) value goes from 7 to 12 and as we increase this value the number of raw bits in a symbol increase and this leads to a gain of range but also to a reduction of data rate, to an incrementation of the time on air and to more battery consumption.

The SF value can vary from 7 to 12 both included, trading data rate with range. The increase or decrease in data rate is proportional to the time on air, greater meaning not only more battery consumed but also greater chance of collision. The increase in range is due to the higher sensitivity of the receiver at greater values of SF. In short, increasing a level the SF means stretching the signal twofold, with its corresponding higher power consumption but better range.

LoRa has to deal with the specific regulations and norms of each region. Here in Europe we must comply the following rules:

- For uplink, the maximum transmitted power is limited to 25 mW (14dBm)
- For downlink, the maximum transmission power is limited to 0.5W (27dBm)
- There is between an 0.1% and 1% duty cycle per day. This means that the time on air of the transmission must not exceed a proportion of the total time.
- Maximum allowed antenna gain +2.15 dBi

1.3 TTN and LoRaWAN

LoRaWAN defines the communication protocol and the system architecture, it acts like a MAC layer for LoRa packets. The Things Network is a LoRaWAN network operator that use this protocol and its network topology to give free LoRaWAN connectivity.

The LoRaWAN network topology, as it's represented in figure 1.3, consist of 4 layers: end-nodes, gateway, network server and application server. Gateways are

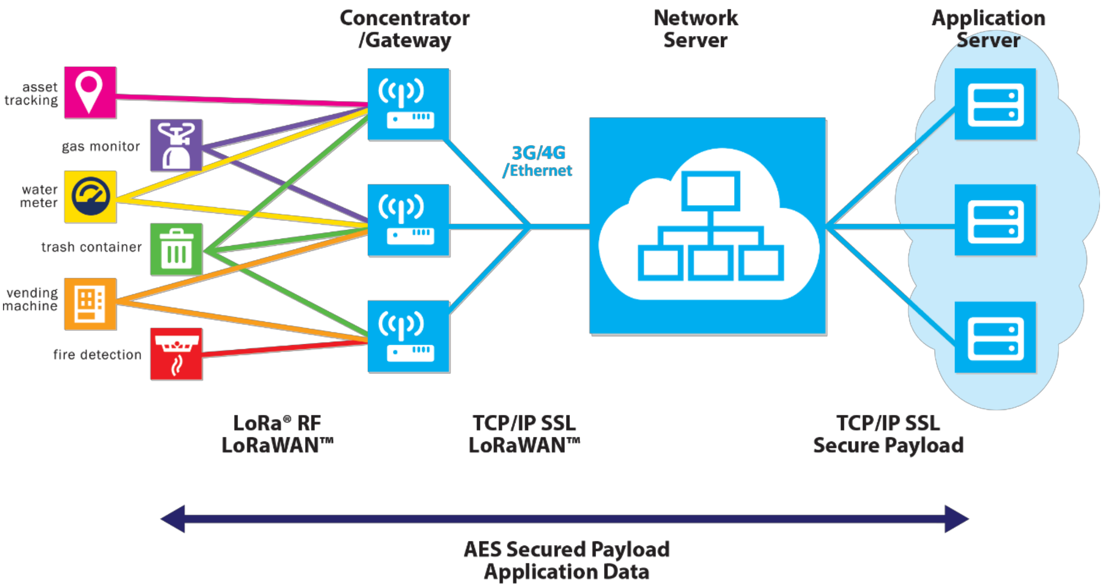

Figure 1.3: LoRaWAN network topology

able to listen from hundreds of end nodes at the same time, even if those nodes are transmitting at different frequencies or SF. The data transmitted through the gateways arrives to the network server via TCP/IP. Then the network server (NS) sends those packets to the application server (AS) and if it is needed, a different server called join server (JS), enters in the process. This server stores the sensitive parts of LoRaWAN applications such as the root keys and generates the session keys when a join procedure takes place.

In LoRaWAN communications we distinguish two types of transmissions depending on the sense of communication. An uplink message is sent from an end-node to a network server and in the case of a downlink message the message in sent from a network server to a specific device.

- **Uplink message:** this message uses the LoRa radio packet explicit mode, which consists of a physical header (PHDR) and a cyclic redundancy check (CRC) header (PHDR-CRC). Another CRC is required to protect the integrity of the payload; all of them are together inserted by the radio transciver in the following way:

Preamble	PHDR	PHDR-CRC	PHYPayload	CRC

Table 1.2: Uplink message format.

- **Downlink message:** The structure is very similar to the uplink message, the only difference is that in a downlink message there is no existence of CRC:

Preamble	PHDR	PHDR-CRC	PHYPayload

Table 1.3: Downlink message format.

This PHYPayload that both, the uplink and the downlink messages contains, starts with a single-octet MAC header (MHDR), followed by a MAC payload (MACPayload) and finishing with a 4-octet message integrity code (MIC):

MHDR	MACPayload	MIC

Table 1.4: PHYPayload

The MHDR defines the type of the message that is sent. Those types are join request, join accept, unconfirmed data up/down and confirmed data up/down, where confirmed means that it has to be acknowledged by the receiver, while unconfirmed does not require that.

The MACPayload contains a frame header (FHDR), which is the device address of an end-device (DevAddr), followed by an optional port field (FPort) and an optional frame payload field (FRMPayload):

FHDR (DevAdd)	FPort	FRMPayload

Table 1.5: MACPayload

Three device classes are defined in LoRaWAN:

- Class A: in this class after the uplink transmission two reception windows are set in order to receive downlink messages. A stands for All, since all devices must implement this functionality.
- Class B: in addition to class A devices, this class open extra receive windows at scheduled times. Time synchronized beacons are sent from the gateway to fix those scheduled times in the end node.
- Class C: this type of devices have almost all time open receive windows and are only closed when transmitting.

Class A is the most common class as it is the most energy efficient.
The frequency band defined by LoRa is divided into 8 channels. In the case of downlink messages there are 8 channels used by the first reception window and another one used by the second window. Due to the regulations in Europe each channel must be of 125 KHz of bandwidth. During a data transmission the channel is constantly changing in order to make the communication more robust to interferences.
An important mechanism that is used to control the uplink connection between

the end node and the gateway is the Adaptative data rate (ADR). This protocol is able to modify automatically parameters like the Spreading Factor or the transmission power according to the signal power that the gateway is receiving. As it's explained before, the SF also affects the power used by the node so this mechanism is able to find the optimal point in which the node is only using the necessary power. The Things Network allows to disable or enable it depending on your purpose. If the device is stationary, TTN recommends to leave the ADR active, but if the end node is supposed to move, the SF and power should be in charge of the programmer.
The ADR algorithm is simple, but has important ramifications on the performance if the device is not stationary. The reason for this is that the ADR algorithm was designed in order to provide the best bandwidth utilization and the best battery life to devices with an stable radio communication. From this falls that a moving end node is not the optimal target for the algorithm. Nevertheless, the effects of it are easy enough to analize. Currently The Things Stack takes the 20 most recent uplinks to compute the new parameters of the transmission. This is the current outline of the behaviour of the TTN as of now:

- If the ADR bit is not set, the TTN doesn't do anything.
- If is set, RF parameters are stored
- After 20 packets, the algorithm is executed, and if the parameters are not optimal or the ADRAckReq bit is true, a downlink is scheduled with the correct parameters. The bit is set to true usually every 64 packets without downlink, but it is up to the device.
- If the ADR failed, the packet is sent again

Figure 1.4 helps visualizing the behaviour:

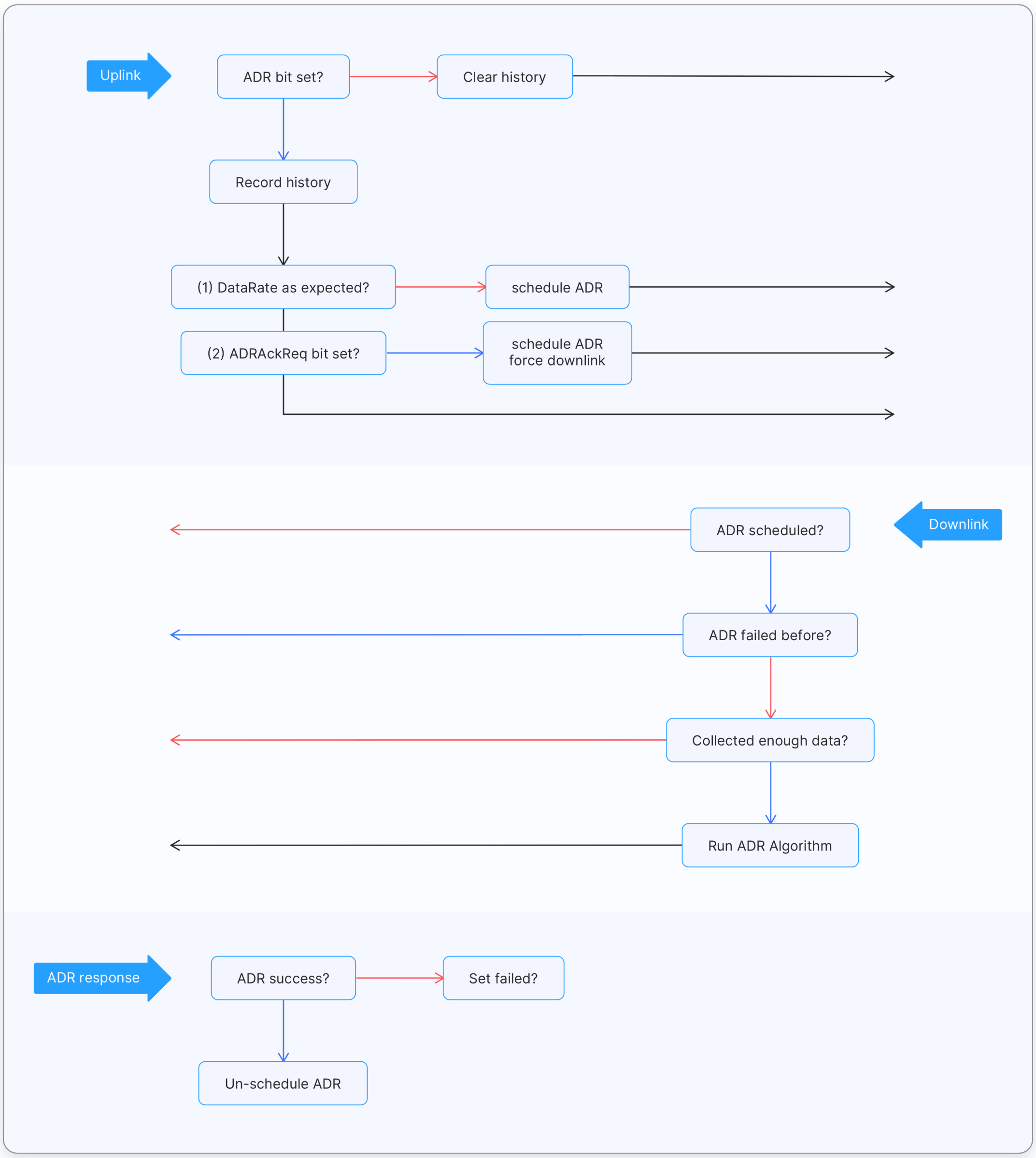

Figure 1.4: Diagram outlines of the ADR flow in The Things Stack. More here

The ADR algorithm is the following:

$$SNR_{margin} = SNR_{max} - SNR_{DR3} - margin_{dB}$$

And with the SNRmargin we compute the steps up or down on DR/SF:

$$StepsUpOrDown = \lfloor \frac{SNR_{margin}}{2.5} \rfloor$$

As we take the maximum of the SNR of the past 20 packets, the performance of this algorithm is tuned for stationary nodes. The TTN recommends the end node to have a fixed power and DR if the end node is movile. The obvious and worst case scenario where this algorithm would fail to give correct estimates is the one where the end node is moving away from the gateway. The first packet would be the maximum SNR, so the algorithm can not account for the decreasing SNR over the course of the next 19 received packets, resulting in a lower SF than what it would be needed in order to receive correctly in the future.

For any LPWAN technology it's crucial to incorporate security. LoRaWAN uses two layers of security: one for the network layer and another one for the application layer. The network security layer ensure the authentication and all the transmission is encrypted by the application layer with the AES method. Private session keys are needed and there are two main different methods to obtain them:

- **Over the air activation (OTAA):** in this method the device needs to be equipped with a DevEUI that identifies the device, with a AppEUI to identify the application and with a AppKey that is used to sign an initial join request. Once the initial request is validated, the join server generates the two session keys, and they are send to the end node. Those keys are the NwSKey and the AppSKey and are used to identify the network server and to encrypt the payload.

- **Activation by personalization (ABP):** with this method you can access the network without a join request because the device is already equipped with the AppSKey and the NwSKey.

Usually the OTAA method is preferred, this is because it can generate new session keys for each session and it allows re-key. ABP is not as flexible as OTAA as the session keys are fixed in the device.

Chapter 2

Related Work

Summary

There have been studies conducted previously that aim to analyze parameters of lora communications in various enviroments, in this section we will see how they relate to our work and measurement results.

In the paper [1] the performance of ADR in a scenario in which the end nodes aren't fixed is analized. The End devide (ED) were following three different routes inside a truck with different speed. The authors found that as the mobility increase, the performance of the ADR starts to decrease leaving space for further improvements in the LoRaWan adaptative data rate.

The idea of paper [2] is to improve the ADR system used in LoRaWan and TTN for mobile ED. The E-ADR (enhanced ADR) they propose is based on transmit messages with the shortest possible time interval between them and to assume the trajectory of the mobile end-node. The results are impressive as they achieve to reduce the ToA and the energy consumption of the ED in the scenarios that are proposed. As a consequence, the packet loss is also reduced or eliminated because the achieved Time on air (ToA) minimizes the use of the allowed time limited by the duty cycle.

The authors in [3] develop a system of geolocalization apart of gps or gsm based on the distance between the ED and the different gateways that is connected. They have reached an accuracy of 100m proving two different algorithms but it's only a firs aproach as this method can't be applied in real scenarios.

An important study made in Pamplona called "Life+Respira" uses mobile nodes in order to aquire pollution data from the city with the help of the sensors equiped on the bicycles. The LoRaWan communication protocol might be used to send the data in this cases, as it is done nowadays is usually with the cellular network. Once the infrastructure is built, many more experiments could be done without needing to invest again. This will come out cheaper in the long run, as batteries and sims are more expensive than to use a tiny lora module. The network could be used for other purposes, in the downtime between experiments. For more information about the study, please read the paper [4].

Figure 2.1: The announcement poster for the "Life+Respira" study

Chapter 3

System architecture and hardware/software used

Summary

In this chapter the design of our system is explained to give an idea of how the idea of this project is applied to the TTN architecture. In the other section we describe the devices and software used in order to achieve the objective.

3.1 Proposed system model

The next figure describes the proposed system in a graphical way to make it clear:

This LoRaWan system that is implemented starts with the laptop running the Arduino program on the adafruit board. The LoRaWan communication module of the board is capable of transmitting LoRaWan packets to the gateway. This comunnication is the one in which the measurements will be taken, so is the one that will be analized in this thesis. The gateway is connected to the Internet

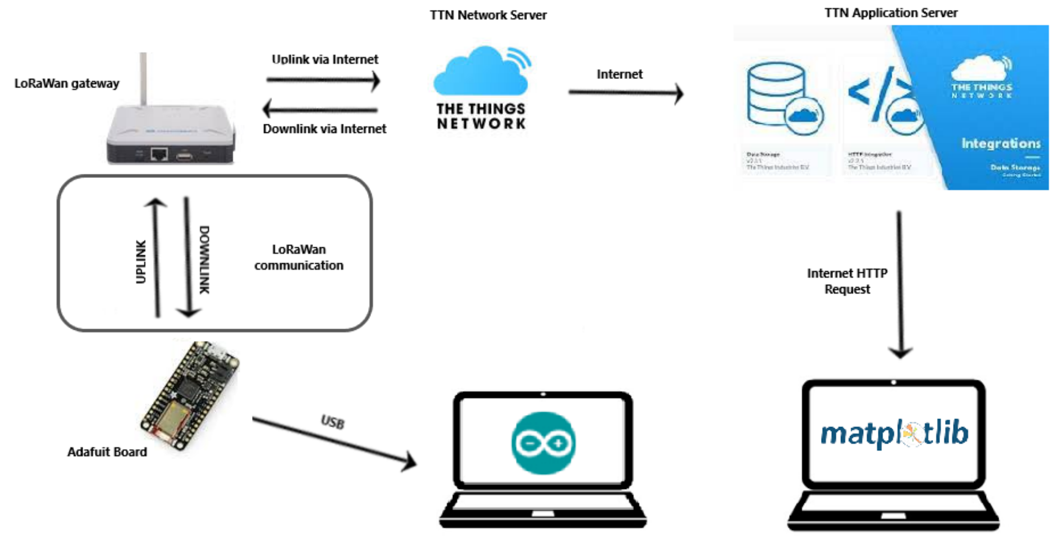

Figure 3.1: Proposed system

via the operator networkand the data sent by the gateway will arrive first to the network server of TTN and then to our respective application server. The laptop is also making HTTP requests in order to analize the data from the database of the application server using python, specifically the library matplotlib. This architecture allows to analize the LoRaWan communication, which is the goal of this project.

3.2 Devices used and their implementation

3.2.1 The gateway

The model of gateway used for this is the indoor Dragino lps-8. This device is forwarding LoRaWan packets via internet. This gateway acts like a bridge connecting the LoRaWan wireless network with the Internet. The indoor model was selected because in this case the gateway is fixed inside a building. The gateway is on the 5^{th} floor, connected to wall power through a USB-C port near a window in order to have the best possible . The access to the internet is provided through WiFi connection to the house router/modem. The configuration of the router was done as it is ilustrated in the figure 3.2:

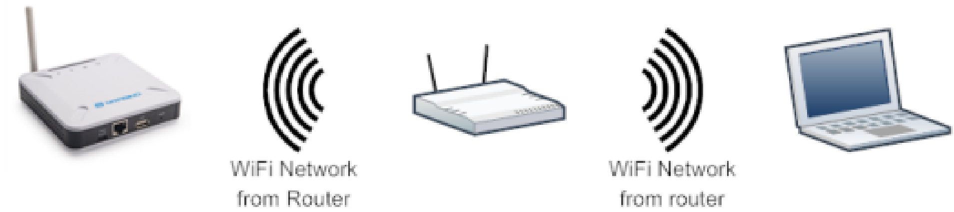

Figure 3.2: Configuration of the gateway via WiFi

On google maps, with a scale of 20 meters as shown in the image at figure 3.3 The gateaway was located in the pinpoint shown in figure 3.3

Figure 3.3: Location of the gateway on the map

3.3 The microcontroller

The microcontroller used as a node in the network is the Adafruit Feather M0 RFM95 LoRa Radio. The device is equipped with a transceptor module LoRa, with built in USB and battery charging. The processor is a ATSAMD21G18 ARM Cortex M0 at 48MHz and 3.3v (the same as the Arduino Zero) and a quarter of a megabyte of FLASH, with 32k of RAM. Thanks to the USB port, and a bootloader, the device can be programmed and debugged through the USB port, no need for an external programmer, which would have been yet another thin layer of complexity.

More info of the board and the pinout can be found here The I2S connectors were used to connect the sensors which communicated through the protocol, and DIO1 was manually wired to GPIO6 so that the radio would work. For each individual sensor a different connection may be needed, so in general, we connected the I2S to the same pins SCI and SDA and the simple voltage for temperature sensor was connected directly to the GPIO A0, setting it to an input on software to read from it.

3.3.1 Sensors

The first sensors mentioned were used to introduce us to the Arduino and IoT enviroment and the last one (the gps module) is the one we finally used in order to measure the distance between the node and the gateway:

- Temperature and humidity sensors:
  - The hdc1080 sensor is a digital humidity sensor with integrated temperature sensor that provide respective measurents using low power consumption.

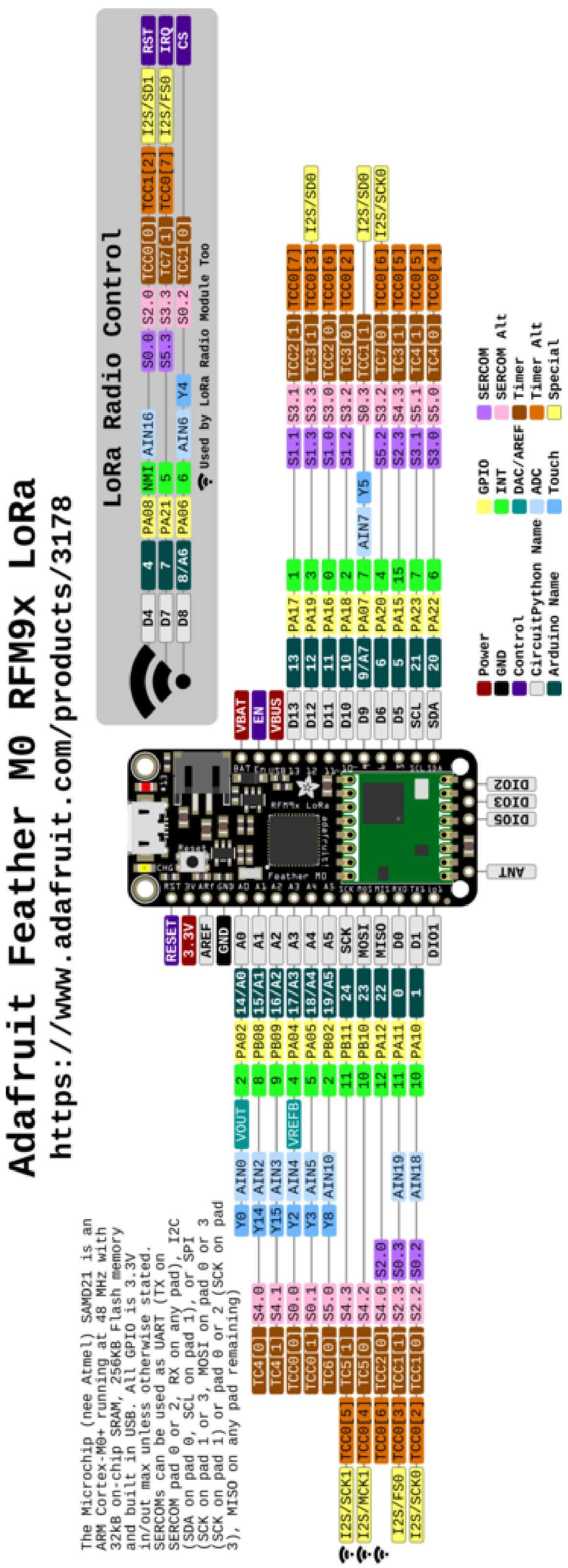

Figure 3.4: The pinout of the module

  - Another sensor with similar purpose is the bme280 sensor. in addition to the other one this sensor is capable of measuring preassure appart of temperature and humidity.

  Two of this sensors were used and the respective models are hdc1080 bme280

- Air quality sensor: sensor of the brand `SPEC SENSORS` specifically the model ULPSM-IAQ. It measures the indoor air quality with very low power consumption and the output signal in this case is analog.

This sensors are designed for the IoT enviroment as all of them use low power consumption. This characteristic is optimal for the purpose of LoRaWan networks using the less power consumption as possible.

3.3.2 GPS module and antenna

The module u-block LEA-6S-0-001 is able to obtain the specific coordinates in which the module is placed with a precission of 2.5m and it's also designed for low power consumption.

This module is designed for the use of passive and active antenna. The antenna selected for this gps module is an active one, which means that the received signal is amplified. The available freaquency bands are the following ones:

In our case we will work with 1575 MHz as it is the band that detects our antenna.

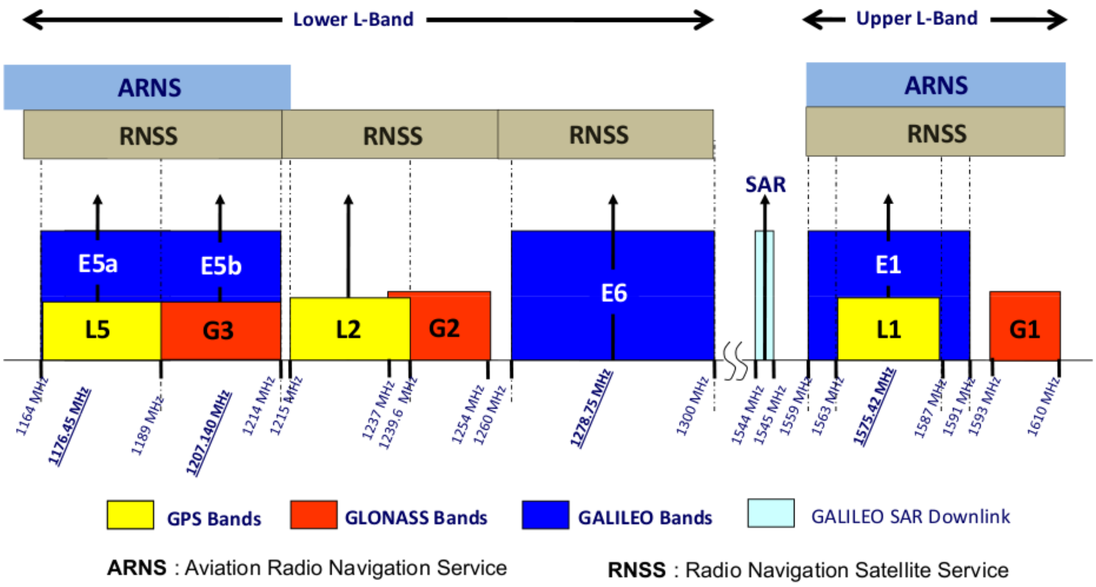

Figure 3.5: Center frecuency and BW for the three different GPS bands

3.4 Software used

3.4.1 MCCI LMIC Arduino library

The `MCCI_LoRaWAN_LMIC` library was the most complete library for our purpose. The recommended library by Adafruit has the flaw of not being compliant with the TTN standard anymore as it cannot receive downlink messages, and the RadioHead library is a LoRa library, requiring us to implement the whole LoRaWAN engine for ourselves. `MCCI_LoRaWAN_LMIC` library is already done so there was no need to waste our efforts. For more information about the library, the documentation, code and repository are here. The library is a pretty big and complex piece of C code, so a brief explanation will be given. Using the 4.1.0 version, so it will be the model explained: (GPS Click - Breakout Board for u-blox LEA-6s, 2022)

In figure 3.6 is pictured the whole finite state machine. The only part a user should be concerned about is the frame data set up, the actions to be taken

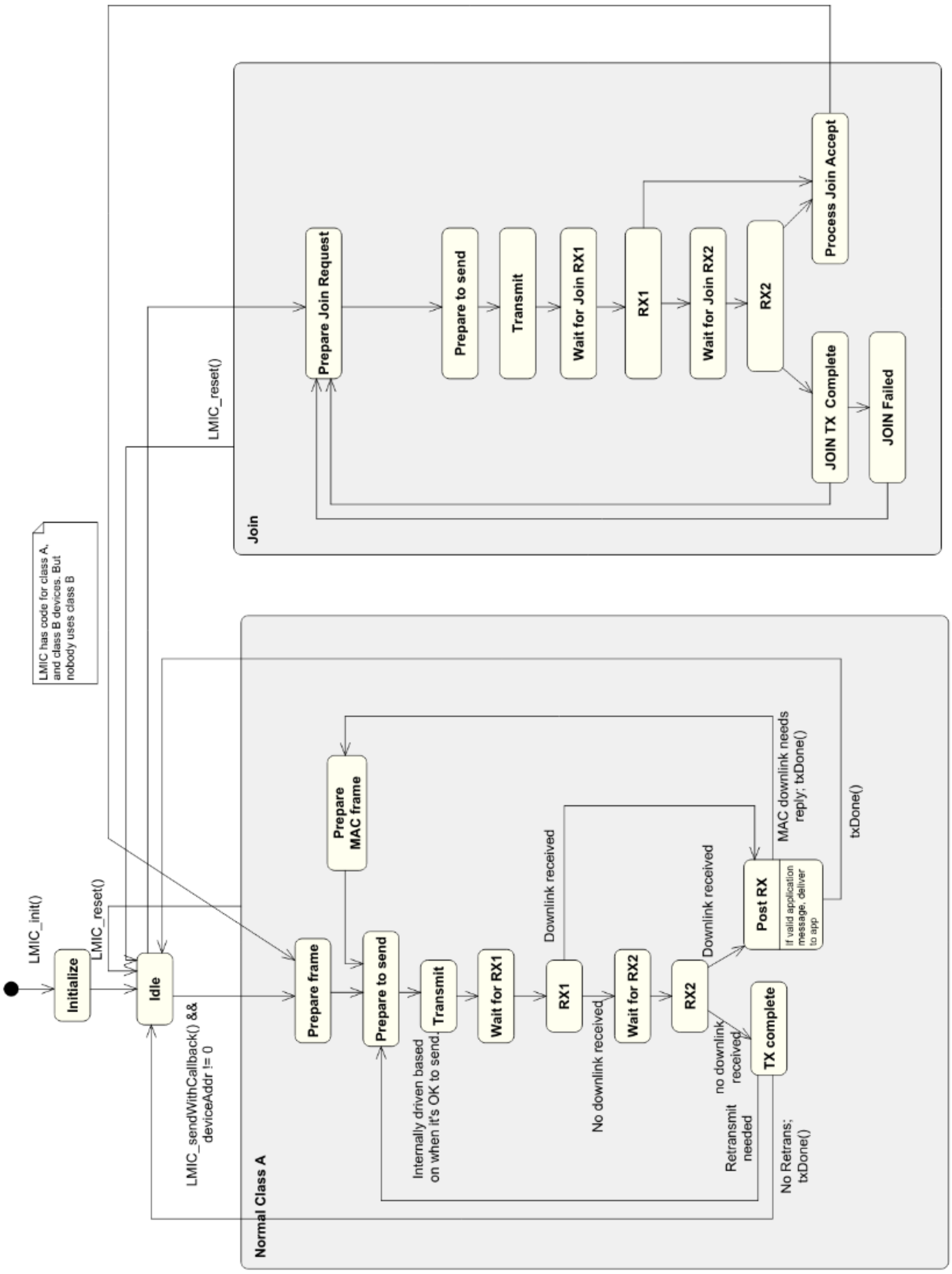

Figure 3.6: Finite state machine of LMIC library

for every event desired to react to with the `onEvent()` call-back function or `LMIC_registerEventCb()` API function. Both are currently supported, and although the `onEvent()` application call-back is currently deprecated, it still works on version 4 and it is simple to implement and debug. The other option is the recommended one, but is harder to comprehend. We ended with a function that, when the package is sent, a timed call- back is issued thanks to the convenience function `os_setTimedCallback()` to schedule a job in the next `TX_INTERVAL`, that job in charge of queuing up another packet of data to be sent, witch will be collected by the engine and, when sent, the cycle starts once again:
first job → sent → wait TX → job → sent → wait TX → job ...
This is "interrupted" by the compliance engine, that ensures that the law is not broken by keeping track of the air time so that the device can't send more than the percentage set by the region configuration in the library (currently on EU 868). The I2C library was used to communicate with the sensors, as a dependency of the particular libraries for each one of them. `ClosedCube_HDC1080` library was used for the more straightforward and the `Adafruit_BME280` was used for the other sensor.

3.4.2 TinyGPSPlus

This library is design to parse all type of NMEA data that the GPS module provides. In our case this library is used in order to extract the exact latitude and longitude of the ED. More info about this library can be found in the github repository

3.4.3 U-center

The software developed by the company u-block is a monitoring system for the gps module mentioned before. U-center provides info about the signal that gps is detecting from the gps-satellites. Once the gps is calibrated the u-center software provides all data the gps is collecting including the longtude and latitude.

3.4.4 Arduino

Arduino was used to program the Adafruit Feather M0 and is one of the recommended ways to program the device, as adviced by Adafruit. The programming interface, the wide support for libraries and the abstraction layer made it a great fit for the goal of this thesis.

3.4.5 Python

Python was used to parse the incoming data and create the grapthics with the popular suite of libraries, like matplotlib, numpy, etc.

Chapter 4

Experimental measurements and metodology

Summary

In this chapter the metodology followed in experiments and the different measuremns that are observed during the data transmission are explained in detail.

4.1 Measurements

The following indicators describe the quality of our wireless LoRaWan communication in the mentiened experiments:

4.1.1 RSSI

Received signal stenght indicator (RSSI) is one indicator that provides the information about how strong the received signal is. It is masured in dBm and its value is always below zero, this means that the closer to zero the RRSI value, the stronger the received signal is.

This indicator is calculated by the sum of different paramenters like the path loss, the transmission power of the antenna, the cable losses and the antenna gain.

4.1.2 SNR

This indicator of quality of signal received takes into account the noise floor signal power in the comunnication. When we transmit in wireless comunnication, the presence of the noise can surpass the power of signal received and not being able to get any information from the signal. The Signal to noise ratio (SNR) is also measured in dBm and it's the subtraction between the power received and the power of the noise floor during the comunnication.
However, LoRa due to its demodulation explained in chapter 2, is able to demodulate and extract information from the signal for negative SNR values. The limit of the minimun value of SNR for the receiver depends on the SF as the figure below shows:

SF	chips / symbol	SNR limit
SF7	128	-7.5 dB
SF8	256	-10 dB
SF9	512	-12.5 dB
SF10	1024	-15 dB
SF11	2048	-17.5 dB
SF12	4084	-20 dB

Table 4.1: Limit SNR and chips per symbol for different SF

4.1.3 Packet Loss

The Packet Loss is simply the ratio between the packets that the gateway is receiving and the packets the end node is sending. This measure is observed thanks to the frame counter that the LoRa packets implent. The frame counter contains the number of packets that the end node is sending.
In order to calculate this ratio the frame counter and the number of packets that appear in our TTN server are considered.

4.2 Metodology

4.2.1 First Experiment

The ADR as it's introduced in 1.3 can modify parameters of communication such as the SF, the transmission power or the BW to get a better SNR or to reduce the excesive power consumption of the end node.

The so-called "Margin" is calculated after a number of transmissions (uplink). The maximun SNR of those transmissions minus the required SNR to demodulate a message given the data rate are considered. This margin is used to determine how much we can increase the data rate or decrease the transmit power.

In the things stack for modifying the SF to increase the SNR and get a better communication, 64 packets without a downlink message of the ADR are needed. In that case the node will increase the SF or the transmission power in order to start receiving downlink packets again.

In the case that the margin is high enaugh, which means that the node is consuming excessive power, the ADR after 20 packets of uplink sends a dowlink message to the node in order to adapt the parameters of comunnication and consume less power.

As it can be supossed from the last information about the ADR, its performance is usefull when the end node is fixed. But if the end node is moving with the ADR activated, what happens to the comunnication? Is better for the data transission to fix one SF and the transmission power in the end node?

Those questions were formulated in order to perform our first experiment. In this first experiment two types of configuration between the end node and the TTN servers are selected. In one, the ADR is activated from both sides of the comunnication as it follows:

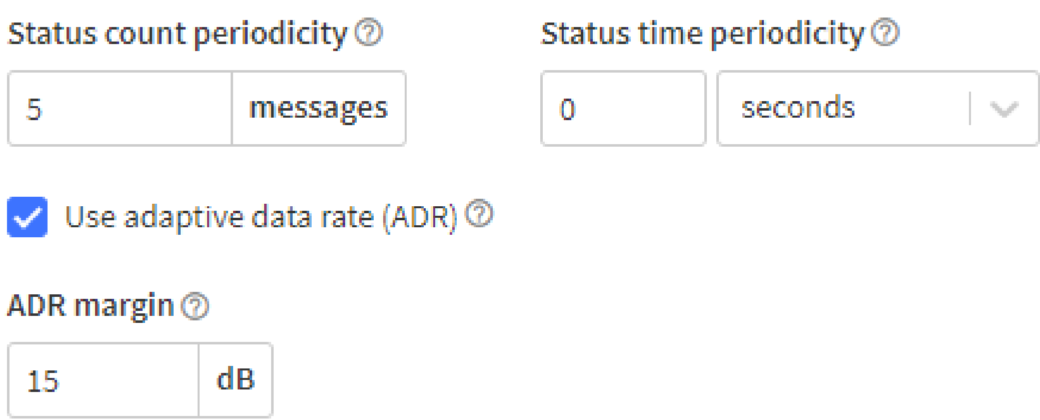

Figure 4.1: ADR configuration in TTN console

For the other one a medium SF is selected, SF10, and the transmission power is fixed to the default one (14 dBm).

The idea of this experiment is to follow the same route in a urban area for both configurations and analize which is performing better by measuring the packet loss, RSSI and SNR in both cases. The route planned for this experiment is inside the city of Genova, in the Porto Antico area as it is ilustrated in 4.2

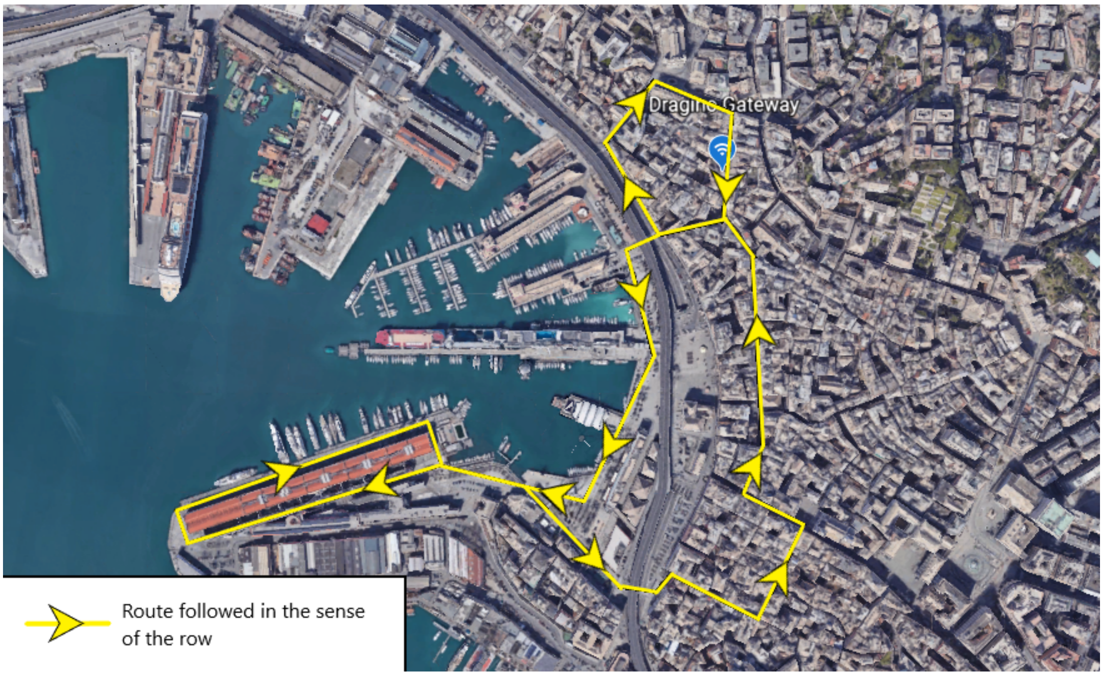

Figure 4.2: Area of Genova where the experiment is performed

Thanks to a battery connected to the adafruit board the mobile end node is builded. This battery was attached to the bottom of the board and packed inside a tupper as it is ilustarated in 4.3 allowing us to move confortably following the planned route.

4.2.2 Second experiment

The ADR shows how important is to choose the correct Transmission power used (Tx power) and SF in a LoRaWan data transmission. The margin calculated by the network server depends basically on how far from the gateway the ED is located and on the area/landscape the network is placed. We know that different scenarios can cause a huge difference in terms of interference for our comunni-

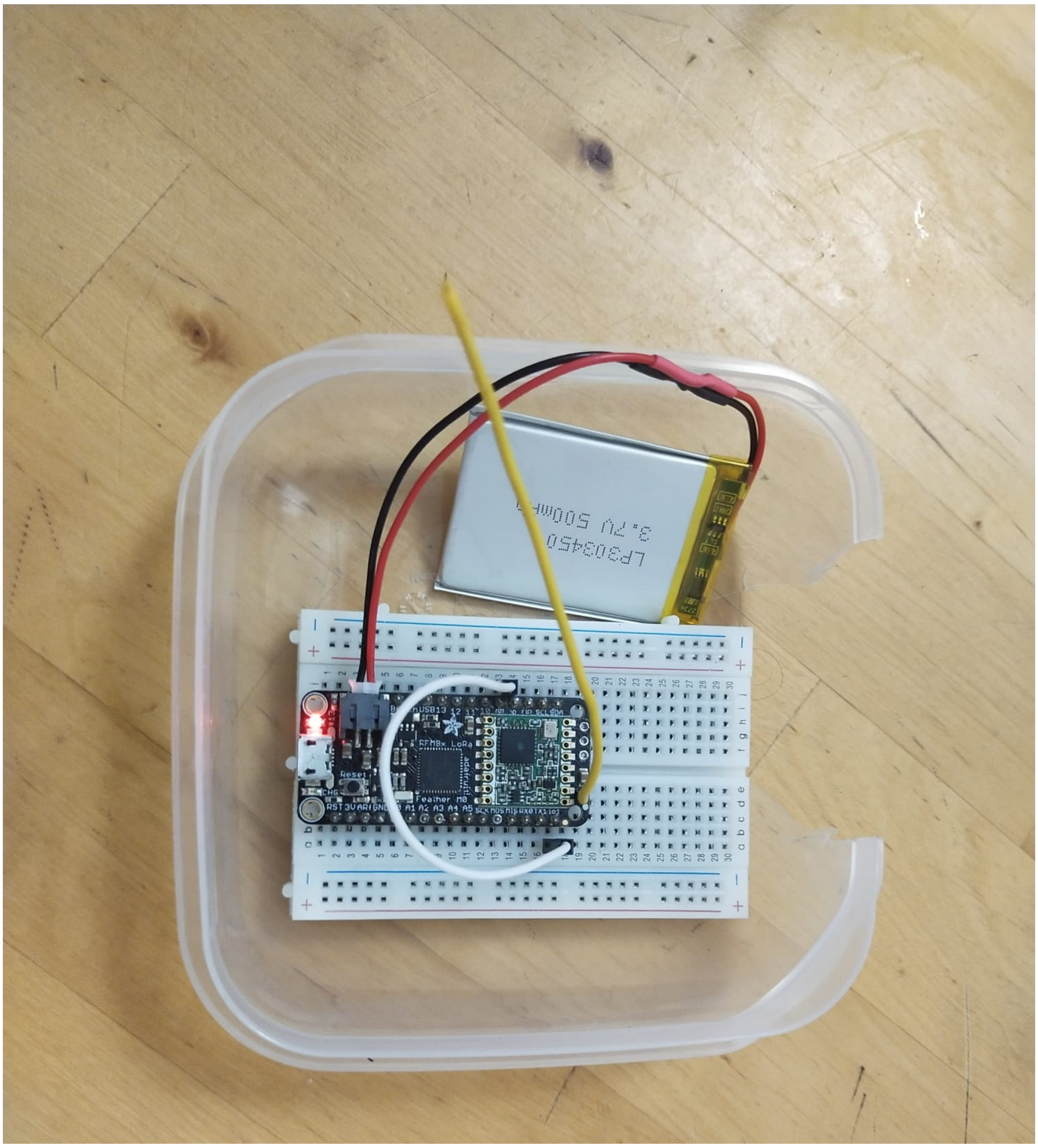

Figure 4.3: LoRaWan mobile end-node used

cation, e.g, in an open space LoRaWan protocol is able to perform much better than in urban areas where buildings and other causes of interference are present. This idea is corroborated by the thesis (link).

We wanted to know how different transmission parameters can affect a LoRaWan communication without the use of ADR and to see the infuence of different landscapes inside a urban area for the indicators cited in 4.1.

The idea for this second experiment the objective is to further expand and analyze the results obtained previously designing a program that will be running in the adafruit board that is constantly changing the values of SF and Tx power.

Once the device is set up we modified our test code to send the coordinates of its current position which will allow us to calculate te distance from our Gateway to the end node and it also allows us to trace the path and determine in which part of the city we are sending our packages. With this information we can analyze how different parameters behave in different enviroments.

The area of testing will be the same as in the last esperiment with the exception that in this experiment we won't follow a fixed route.

Chapter 5

Data analysis for first and second experiments

In this section the graphs are displayed, whereas in 6 this will be explained. For the second experiment the parsed json structure was used to simplify the representation of the data with python.

5.1 Results of the first experiment

The packet structure for the first experiment is as follows:


```
{
    "powRet": 17,
    "f_port": 42,
    "f_cnt": 51,
    "rssi": -90,
    "rssi_ch": -90,
    "snr": 4.8,
    "bw": 125000,
```

```
    "SF": 7,
    "f(MHz)": 867.5,
    "time": "YYY-MM-DDTHH:mm:ss.ssssss+HH:mm",
}
```

The tables for losses on the first experiment are included, but for the second, the tables are on chapter 6

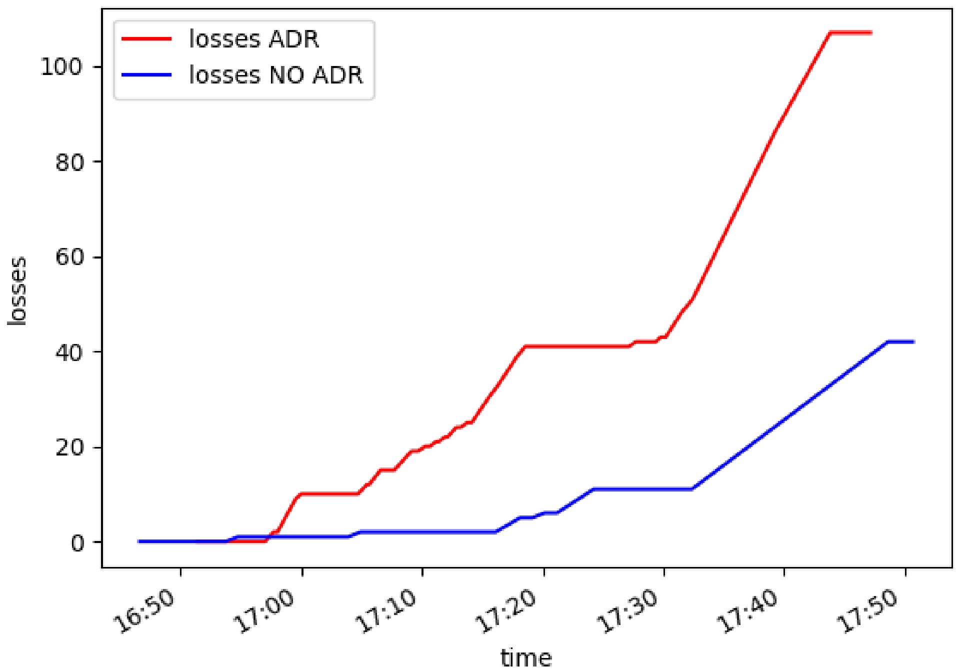

Figure 5.1: Acummulative packet loss depending on the time the packets arrived

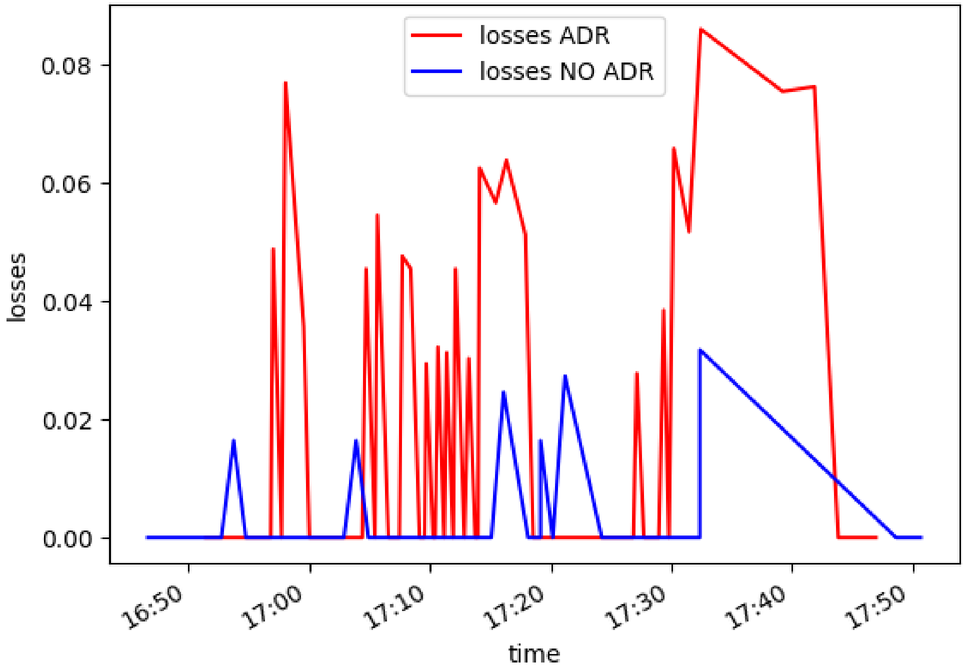

Figure 5.2: The discrete derivative of 5.1

ED configuration	Packets sent	Packets received	Packet loss
ADR activated	263	156	40.68%
SF10 and 14 dBm	112	70	37.5%

Table 5.1: Table comparing the packet loss of the two cases

ED configuration	Medium SNR (dB)	Medium RSSI (dBm)
ADR activated	2.07	-103.68
SF10 and 14 dBm	-2.11	-111.85

Table 5.2: Table comparing the medium RSSI and SNR in both cases

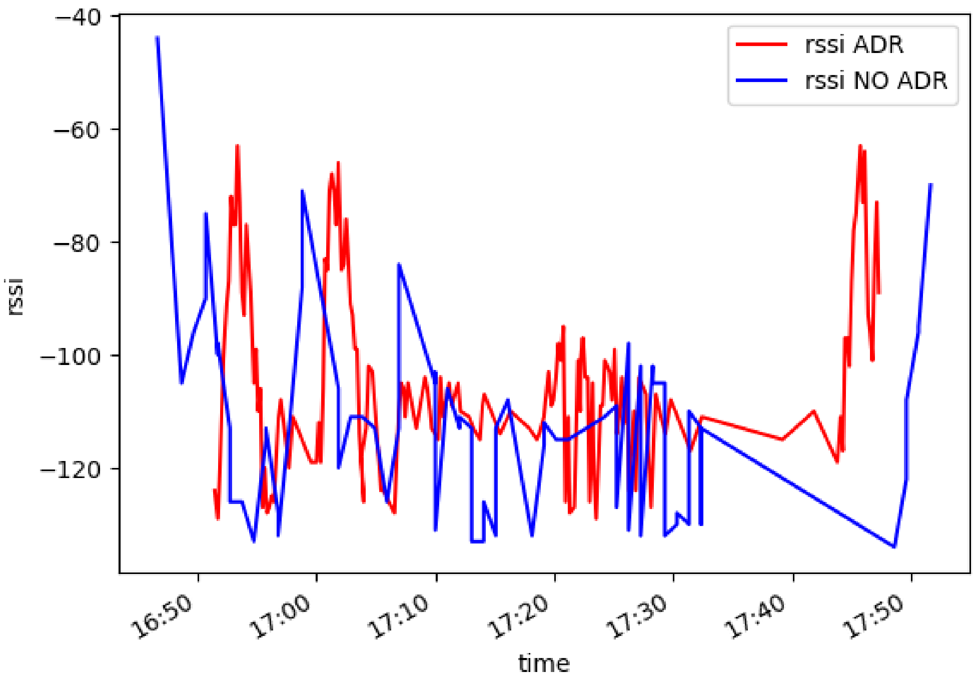

Figure 5.3: RSSI depending on the time

5.2 Results of the second experiment

Our parsed package structure is the following:


```
{
    "powRet": 17,
    "f_port": 42,
    "f_cnt": 51,
    "rssi": -112,
    "rssi_ch": -112,
    "snr": 4.8,
    "bw": 125000,
    "SF": 7,
    "f(MHz)": 867.5,
    "time": "YYY-MM-DDTHH:mm:ss.ssssss+HH:mm",
    "lat": 44.4134407043457,
    "lng": 8.928983688354492
}
```


This was downloaded and parsed from the storage integration The Things Network provides. It is suitable for us, since we do not need instant feedback, for that we would have to set up a webhook on an external server. The date is on ISO format. From here we parse the graphs depicted on chapter 6.

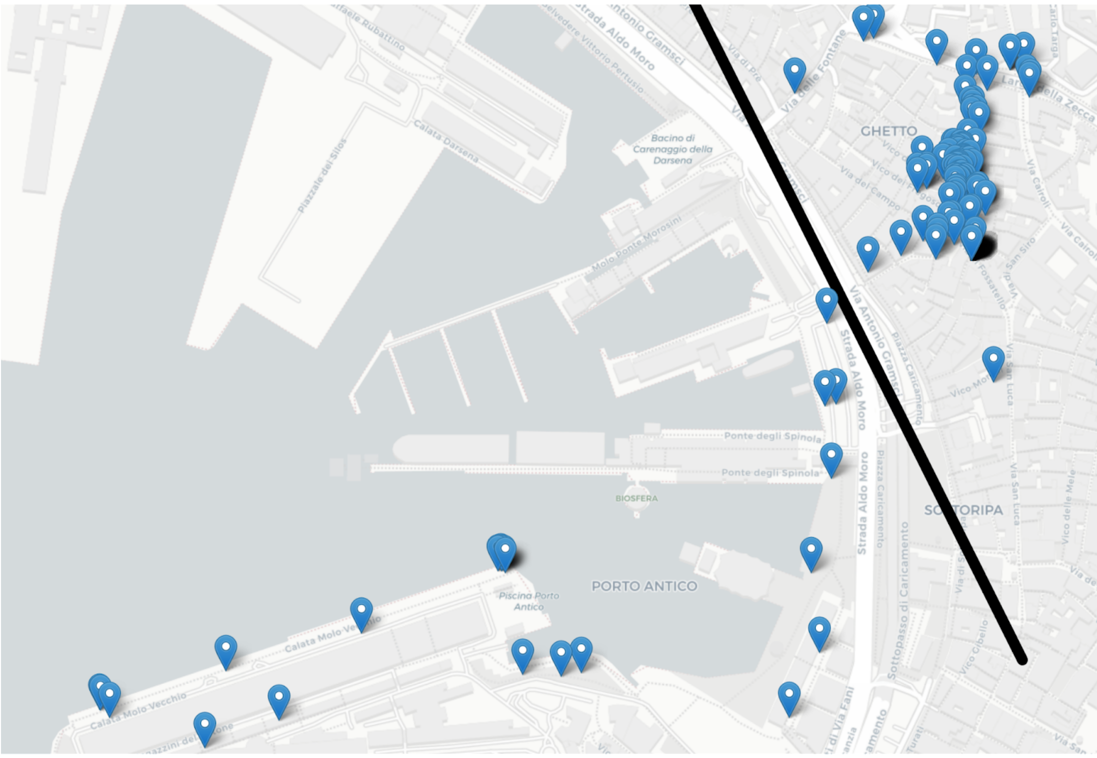

Figure 5.4: Separation line used

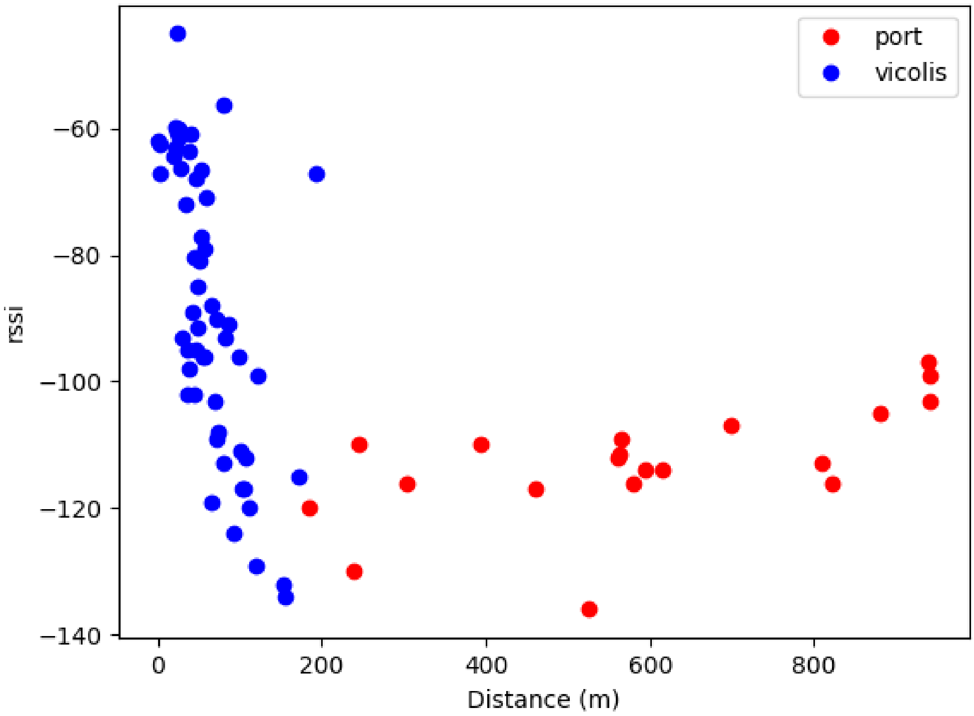

Figure 5.5: Separation line used

Figure 5.6: Path taken colored by the losses

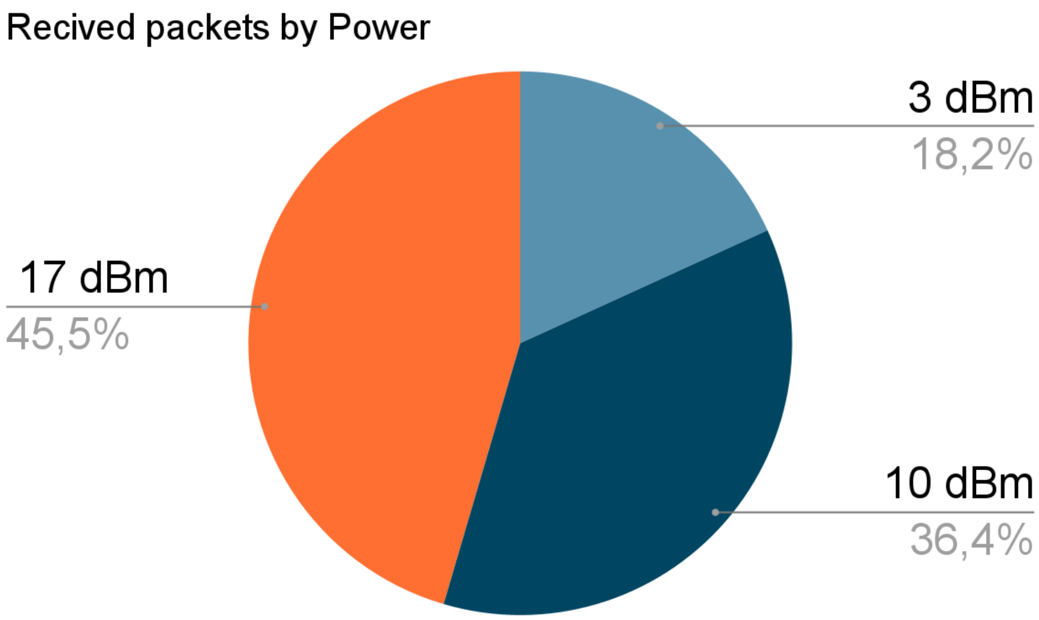

Figure 5.7: Losses grouped by power

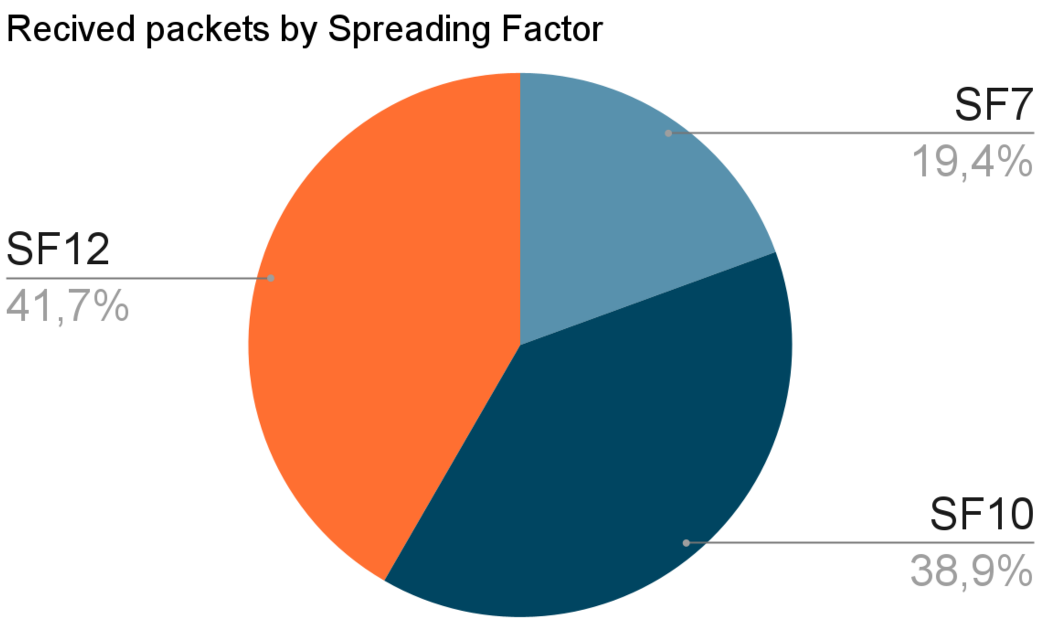

Figure 5.8: Losses gropued by spreading factor

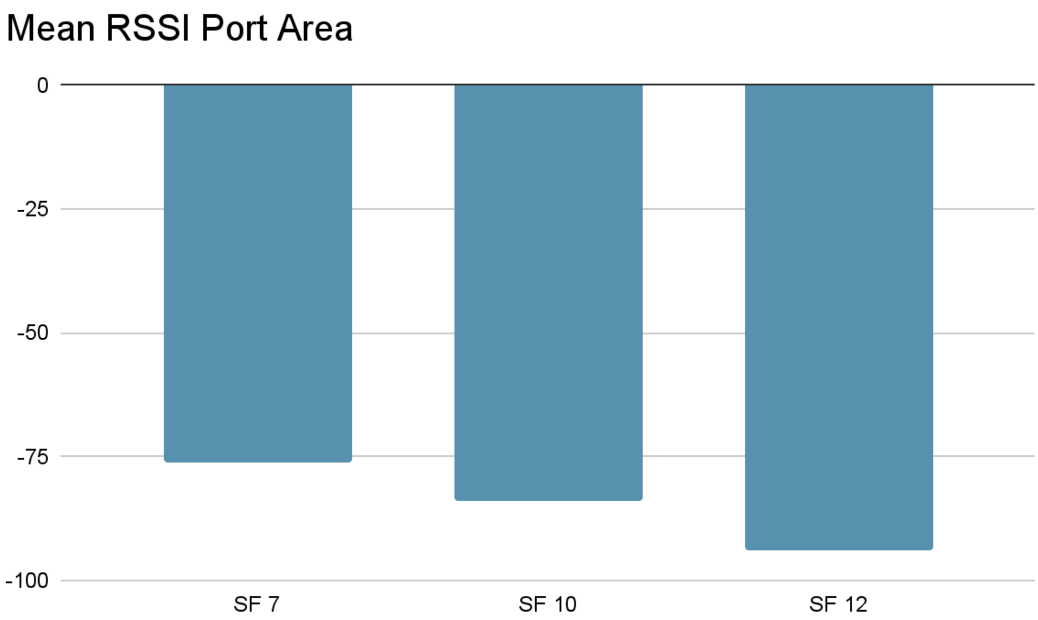

Figure 5.9: Mean RSSI grouped by spreading factor

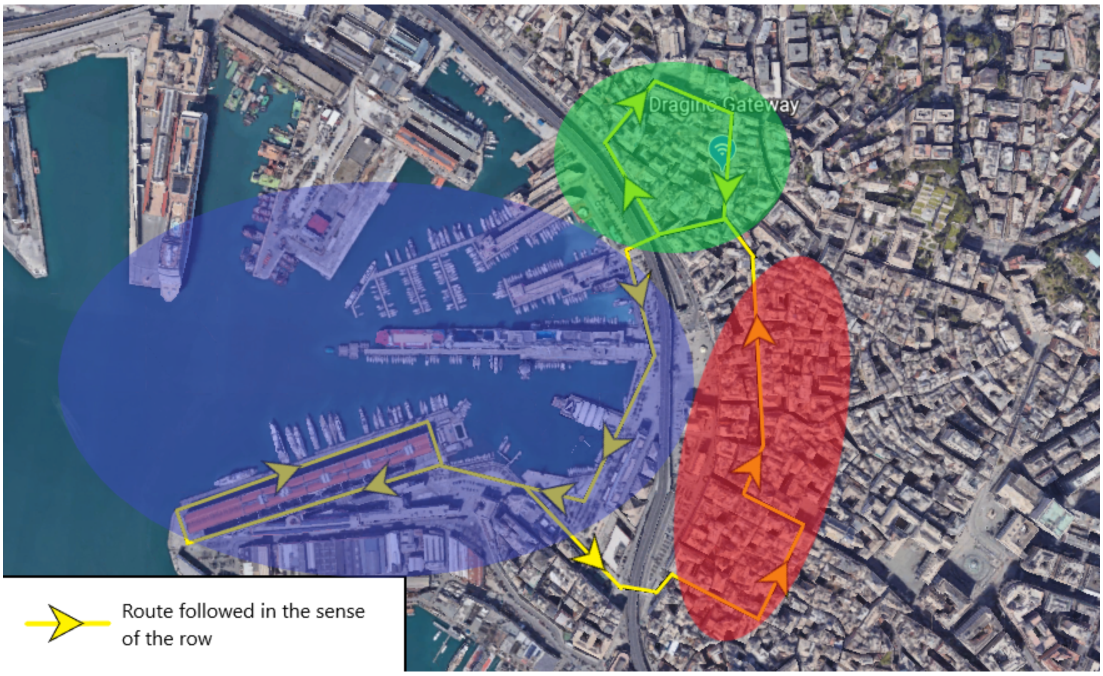

Figure 5.10: Areas, separated by color: green for an area where some packets arrived, red for the far vicoli area and blue for the port area, which aproximates to open space

Chapter 6

Conclusions and future work

6.1 First experiment conclusion

In the configuration with ADR the ED did not change from SF7. This is problably because we were moving and we didn't get 20 uplinks a low maximun SNR to increase the SF. We din't get either 64 uplinks without a confirmation as our maximun packet loss between two uplinks is 35. (All this is expalined in 4.2.1). In this case the ADR configuration was pointless as the end node wasn't able to change the parameters like the SF or the power to obtain the optimal data transmission. As we got always an spreading factor of 7, the performance during the experiment with this configuration is worse compared with the one we selected with SF10 and a fixed Tx power of 14dBm as you can see in 5.1. The reason why in 5.2 the values of SNR and RSSI are lower in the case of ADR is because the other configuration is able to detect signal in worse situations, leading to lower RSSI values.

Thanks to the performance of this experiment we were able to extract valuable information releted with the signal received depending on the landscape of the city. We distinguish two zones, the area of "vicolis" and the port area: "Vicoli" is an italian word used in the city of Genova for narrow streets. The historic center of Genova is plenty of them making the data transmission almost impossible unless the ED is close to the gateway (around 100m but it depends if you have direct vision). Next to this zone is the area of "porto-antico" which is a much more open-space area where the comunnications reach much longer distances. The results and graphics shows that the port area showed in the figure (link) is much more suitable for a LoRa comunnication rather than the area of the "vicolis". In the figure 5.10 on page 44 the different zones of the route that we detected are drawn with different colors. The green circle is the area near the gateaway and despite it was inside the most dense area, almost all packets were able to arrive. Two more areas were defined as we were moving away from the gateway, the blue area where the connection was stable (some packets were lost but not most of them) and the red zone in which the gateway was not collecting any info from the end-node.

6.2 Second experiment conclusion

For this second experiment the objective was to further expand on the results that we obtained on the previous one, where we concluded that the packet loss increased exponentially when the device was on a highly building dense area. After receiving the packets with their locations, we separated, as in figure 5.4 on page 39the map into two regions:

- The lower area, where the conditions resembled the ones in free space
- the upper area where dense buildings can interfere with the signal

This information also allowed us to extract the path followed and know if there had been any losses between the points received. With that information the map on figure 5.6 on page 41 map was created:

In the map 5.6 on page 41 the points are connected by lines. A red line between two points indicates that some packets were lost between them. The width of the line indicates the amount of packets lost, wider lines indicating bigger losses. A green line indicates the opposite, with wider lines indicating the streak of packages received.

Once the points were separated, we analysed the Received signal strength of the points in both groups obtaining the figure 5.5 on page 40 comparing both.

We can clearly see that the signal strength in very dense urban areas decreases very rapidly and fades near the 200 meter distance whereas on an open area it maintains stable levels for much longer distances.

Another conclusion can be extracted from this data. If we look at the percentage of packets received when dividing by spreading factor and power transmitted, on open areas the results closely correlate with theory. On the other hand, the data in the urban dense area highlights that these parameters do not have noticeable impact due to the fact that the landscape plays a much more important role.

6.3 Final conclusions and future work

The results of the experiments lead to the conclusion that the most important factor when performing a LoRaWan project inside a complex urban area for mobile ED, is the study of the landscape for the different areas. The less obstacles between the ED and the gateway, the better for our communication.
Transmission parameters as the SF or the Tx power are useless when the signal is not received by the gateway because of interference caused by buildings and other obstacles for the signal inside a city. This parameters are relevant for a more open space area, the port area in our case, in which they can make the

difference regarding the range for our communication.
The ADR for mobile end nodes where packets are not sent with a high frequency (one per minute) and where the mobile end node is moving with not much speed (walking case) is pointless.
Regarding the future, the limitations due to the duty cycle didn't let us collect the data as we wanted. The transmission interval between packets was controlled by the library used and it was impossible to collect a huge amount of data in a short period of time. Being able to control the time between packets would suppose a great improvement for the experimentation. This improvement would lead to a better analysis of data regarding the SF and the TX power as more packets would be taken into account and also to build new hypothesis such as how this interval can affect the ADR. In order to measure the packet loss depending on distance, a device that collects all the messages the ED is trying to send is needed. By adding this device to the end node, the distance between the ED and the gateway for the packets that doesn't reach the gateway can be calculated. This would suppose a new interesting measure that can be compared depending on the SF and Tx pow in a open space area.

Chapter 7

LMIC Modifications

In order to make LMIC change some parameters, we modified a little of the source code.
On lmic/lmic.h:

```
extern int ENERGIA
```

On lmic/radio.c :

```
static void configPower () {
    // our input paramter -- might be different than LMIC.txpow!

    s1_t const req_pw = (s1_t)ENERGIA;
    ...
}
```

References

[1] K. Kousias, G. Caso, Ö. Alay, and F. Lemic, "Empirical analysis of lorawan adaptive data rate for mobile internet of things applications," in *Proceedings of the 2019 on Wireless of the Students, by the Students, and for the Students Workshop*, pp. 9–11, 2019. 14

[2] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, "Enhanced adr for lorawan networks with mobility," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1–6, IEEE, 2019. 14

[3] B. C. Fargas and M. N. Petersen, "Gps-free geolocation using lora in low-power wans," in *2017 global internet of things summit (Giots)*, pp. 1–6, IEEE, 2017. 15

[4] C. S. Elola and J. M. S. Ulecia, "Life+ respira: reducción de la exposición de ciclistas a contaminantes urbanos," *Boletín del Grupo Español del Carbón*, no. 48, pp. 40–44, 2018. 15

[5] A. Carlsson, I. Kuzminykh, R. Franksson, and A. Liljegren, "Measuring a lora network: Performance, possibilities and limitations," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 116–128, Springer, 2018.

[6] A. Mohamed, F. Wang, I. Butun, J. Qadir, R. Lagerström, P. Gastaldo, and D. D. Caviglia, "Enhancing cyber security of lorawan gateways under adversarial attacks," *Sensors*, vol. 22, no. 9, p. 3498, 2022.

[7] A. Akhbari, M. Allegretti, and L. Matekovits, "Lora applications and validations in complex urban environment,"

[8] "The things network lorawan learn web site." https://www.thethingsnetwork.org/docs/lorawan/.

[9] "Lora alliance web site." https://lora-alliance.org/about-lorawan/.

[10] "Lora alliance web site." https://lora.readthedocs.io/en/latest/.