ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A LoRaWAN Full IoT Pipeline Implementation

by

Facundo David Farall, 1900092175

Gonzalo Joaquín Davidov, 1900092260

Rafael Nicolás Trozzo, 1900092496

Professors:

Luciano Bononi

Marco Di Felice

Internet of Things - Distributed Systems

Alma Mater Studiorum University of Bologna

Bologna, Italy

Saturday 5th June, 2021

Contents

1 Introduction

The current report deals with the implementation of a full IoT pipeline using the LoRaWAN protocol stack as the main protocol for the communication in a radio channel. The system involves a wide variety of IoT technologies, belonging to different parts of the pipeline (edge, fog and cloud) to complete the path that goes from a GPS sensor that acts as a bike in a real world, to its visualisation through a web-page, adding physical and application-like meaning to the data transmitted.

A proof of concept will be provided, with results taken from a real-world implementation, as well as measurements to characterise the performance of the link.

2 System Overview

The application goal is to perform bikes' location tracking based on low-cost technology like LoRaWAN, Arduino and Raspberry Pi. All the locations are stored and are afterwards visualised and analysed, enabling a potential user to take decisions based on this information. Another goal is to analyse the parked bikes to know if they are inside a 'safe zone', which is Bologna's historical centre. If they are not, a notification is sent by email so that this situation can be taken care of.

The cost of the whole implementation is shown in Table 1. Since the components used are in the affordable tier, there are limitations regarding the functionalities available for the LoRaWAN network.

Item	Price
Arduino Uno	20 €
LoRa Shield	35 €
GPS Sensor	10 €
LoRa GPS Hat	60 €
Raspberry Pi 3	50 €
Total	**175 €**

Table 1: Implementation price

3 LoRaWAN Stack

The LoRaWAN protocol stack is an IoT solution that involves different layers of the protocol stack using various approaches. Technically speaking, the name LoRa refers to the Physical Layer, and it is a proprietary solution developed by Semtech which uses the 900 MHz ISM band (no license needed). The exact frequency used will change from region to region, for instance in the EU corresponds to 868 MHz while in the USA belongs in the 902-928 MHz.

On the other hand, LoRaWAN is the MAC, Network and up to the Application Layer. It is open and it is defined by the LoRa Alliance.

The following sections of this report will briefly explain and explore our specific use and approach to each of the layers composing the LoRaWAN protocol stack.

3.1 LoRa Modulation: Physical Layer

LoRa uses a Chirp [1] Spread Spectrum modulation technique to efficiently resist interference, multi-path fading and Doppler effect. It consists of frequency sweep between f_{min} and f_{max} (therefore having a bandwidth $BW = f_{max} - f_{min}$) wich can be an *up chirp* (f_{min} to f_{max}) or *down chirp* (f_{max} to f_{min}).

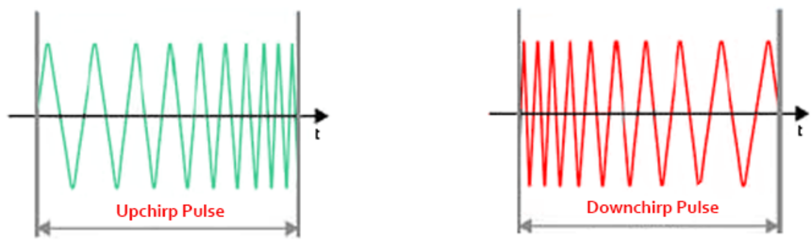

Figure 1: Up chirp vs down chirp

One Chirp is the container for a symbol, meaning that one frequency sweep is the time it takes for a LoRa transmission to send a symbol. The detection of which symbol is being received, is done by looking at the frequency it starts with, at the beginning of the sweep. Figure 2 better illustrates these different alternatives for the Chips (symbol contained in the Chirp), but also elaborates on how many different symbols you can have, by introducing the concept of the Spreading Factor (SF).

[1]CHIRP: Compressed High Intensity Radar Pulse

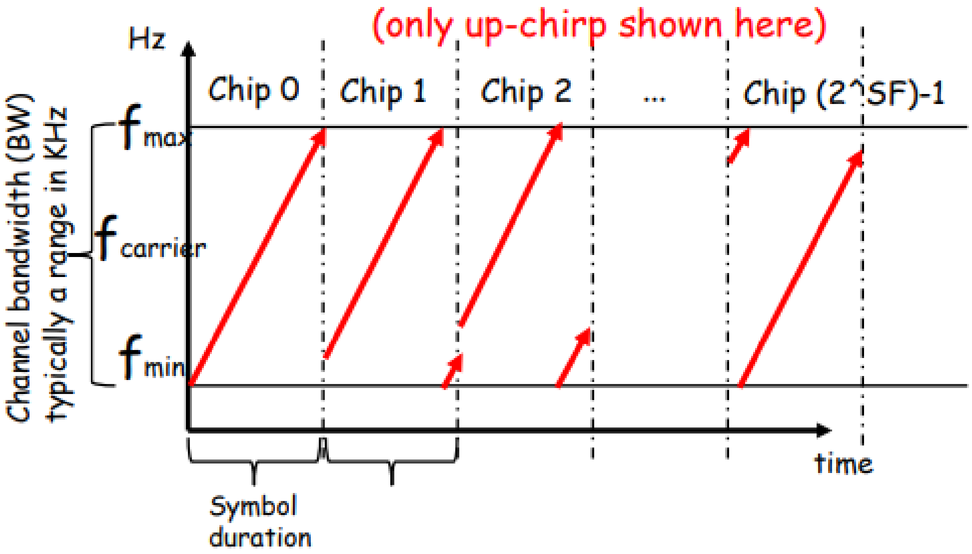

Figure 2: Different Chips (symbols) that a Chirp can take for a given Spreading Factor SF

The Spreading Factor determines the duration of a symbol (duration of the sweep) given by Equation 1, but can also be interpreted as the number of bits carried in one symbol. SF can take values from 7 to 12 included, in the EU region, and every time it is increased by one, the number of possible Chips doubles, and so does the symbol time.

$$T_s = \frac{2^{SF}}{BW} \tag{1}$$

This indicates that a higher Spreading Factor will increase reliability and area coverage, at the expense of transmitting slower the information over time (you add just one bit per symbol but double the time it takes to send it). An additional disadvantage of increasing SF is the doubled energy consumption, since as mentioned before, the time on air of the transmission is twice as long. Figure 3 sheds some more light into the matter by showing a visual comparison of the time it takes for each Chirp to be sent, when increasing the Spreading Factor.

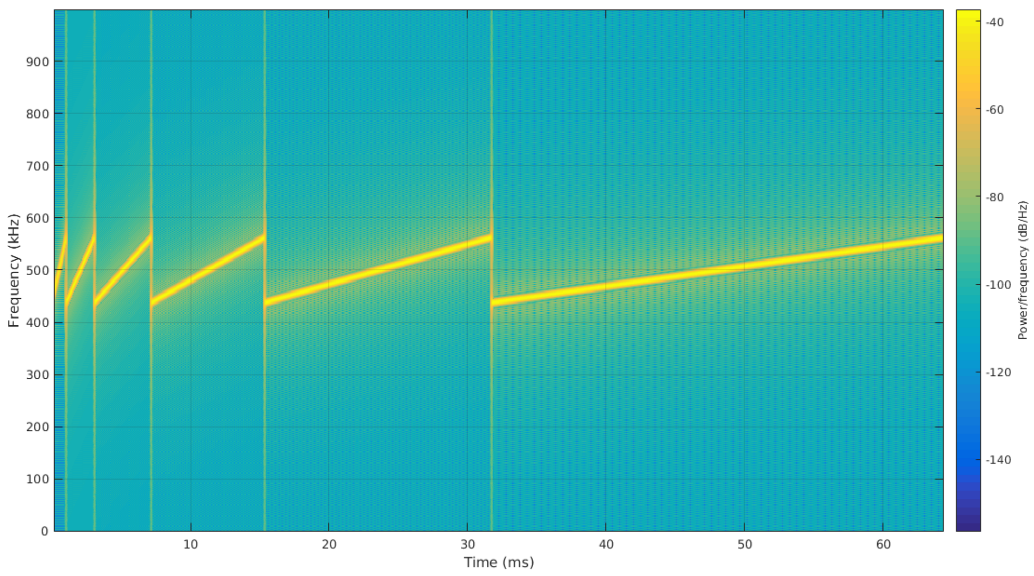

Figure 3: Comparison of SF7 to SF12

3.2 Gateways

LoRaWAN gateways are the bridge between the edge devices and the servers. They use high bandwidth networks, for example WiFi in this application, to connect to the servers. All gateways within reach of a device will receive its messages. In this application, the available receiver only has one channel, so technically it is not a gateway, but a packet forwarder, because this kind of devices are not LoRaWAN compliant.

3.3 Network Server and Application Server

The LoRaWAN stack ends with the Network and Application servers. The first one is in charge of performing duplicate messages detection and filtering. This is necessary because multiple gateways can receive the same message and forward it. It also implements Adaptive Data Rate (ADR) control of the devices and decrypts messages. Then, the application server performs application-specific data processing and makes the devices data available for its use by other applications. For example, it can send the data through an HTTP request.

Devices have two ways of joining the network, Activation by Personalisation (ABP) and Over the Air Activation (OTAA). The ABP is simple because it does not require a join procedure, each device has in its firmware a unique set of Network and Application Security Keys used for encryption. Then, to start transmitting the device just sends packets. On the other hand, OTAA is better in terms of security, since the Network Server generates Session Keys dynamically for devices joining the network. This requires downlink support, which is not the case of this application.

4 Edge: GPS sensor and LoRa Tx, Arduino

The end device used consists on an Arduino UNO measuring its location with the NEO-6m GPS Sensor and sending them through LoRaWAN with the Dragino LoRa Shield, as depicted in Diagram 4 or the real implementation in Figure 5.

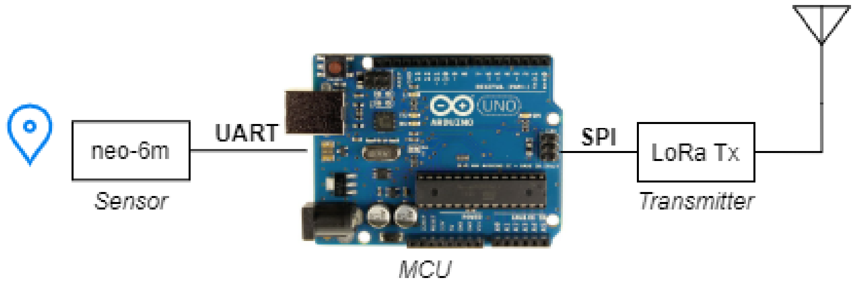

Figure 4: End Device Diagram

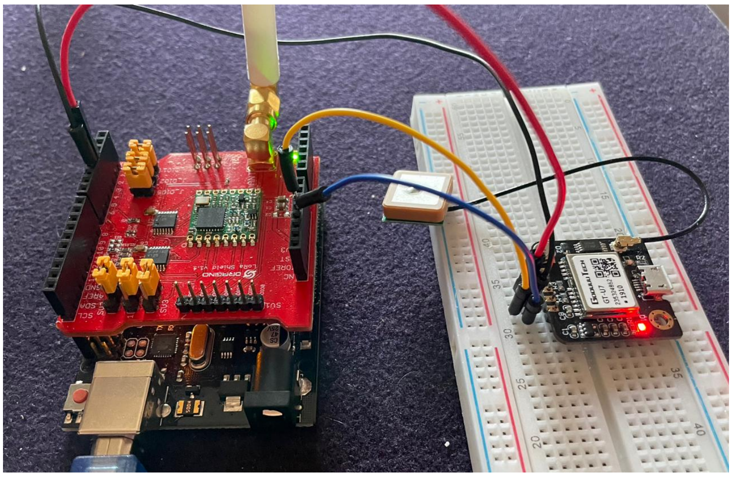

Figure 5: End Device Implementation

The NEO-6m can provide all the information given in the NMEA Standard which besides GPS location is, for example, speed, date and time. However, the only variable of interest for this application is the location.
All the information is sent to the MCU through the UART serial interface. Since the Arduino Uno has just one UART unit which was used for Serial printing, an extra software UART was needed. The most popular library for doing this is SoftwareSerial, but it is really inefficient, that is why the NeoSWSerial library was used, which handles better the resources and uses less memory. Afterwards, the sensor data needs to be parsed, for which another library is used. Here TinyGPS is a popular but inefficient option, because it compiles the code for all the functionalities even if they are not used. For that reason, the chosen library is NeoGPS, which allows

to choose which functionalities are compiled to the MCU.
Finally, for the LoRaWAN transmission the MCCI LoRaWAN LMIC Library is in charge of sending the packets using ABP. The frequency used is 868.1MHz, and SF is fixed, usually SF7.

5 Fog: Packet-Forwarder and ChirpStack, Raspberry Pi

The fog consists on the Dragino LoRa HAT connected to a Raspberry Pi 3 running the Packet Forwarder and the LoRaWAN servers. The Gateway bridge, Network Server and Application server are provided by ChirpStack and they are running locally on the Raspberry Pi. Figure 6 shows how the structure described looks and Figure 7 shows the real implementation.

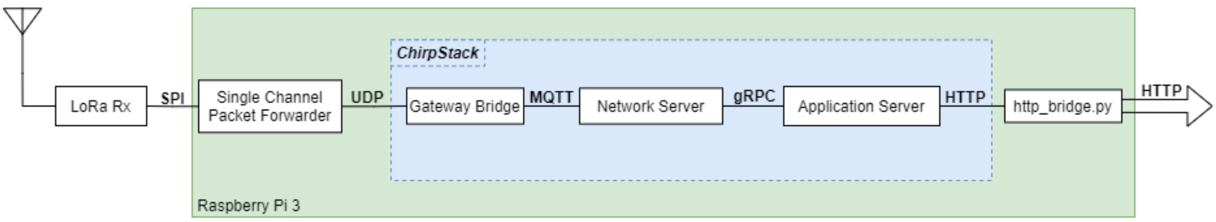

Figure 6: Fog Diagram

Figure 7: Raspberry Pi with LoRa Hat Implementation

Once the packet has been transmitted from the edge, the gateway is in charge of receiving the data. The single channel LoRa HAT running a Single Channel Packet Forwarder is connected to a Raspberry Pi listening to only one specific frequency so it is important that it is listening to the same frequency as the one the transmitter is using. The other parameter that should also be the same in the receiver and the transmitter is the spreading factor. Then, the packet is sent via UPD to the gateway bridge provided by ChirpStack. It converts LoRa Packet Forwarder protocols into a ChirpStack Network Server common data-format (JSON and Protobuf) and sends this new packet via MQTT to the Network Server. The Network Server is in charge of making sure that only data from registered nodes are forwarded and then sends the data to the Application Server which provides payload processing and decryption and makes the data available through wifi access point using HTTP protocol.

ChirpStack provides a dashboard where all the available data can be easily visualised. This is running in the local network listening to the port that the application server provides the data to. Figure 8 shows an example of how the uplink data packets are shown on the API already on the application server.

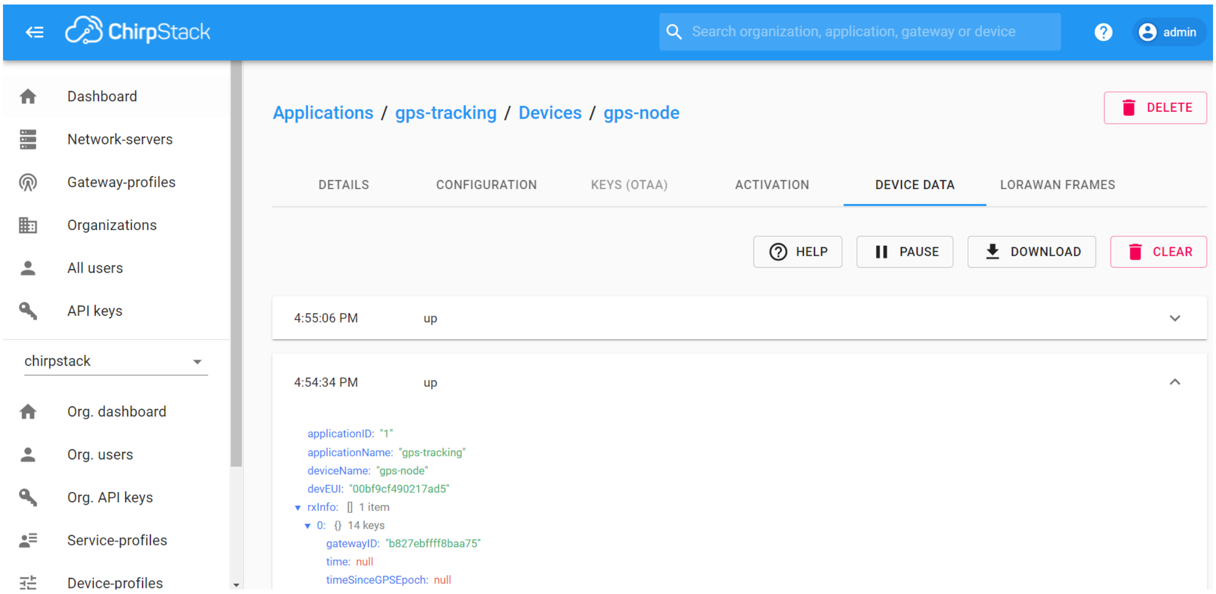

Figure 8: ChirpStack API

The reason to use ChirpStack instead of the most popular solution which is The Things Network is that single channel gateways are not supported by this network.

6 Cloud: ThingSpeak, IFTTT and Web

A diagram of the part of the application running in the cloud is shown in Figure 9. It is based in ThingSpeak, an IoT platform running MATLAB which enables the use of visualisations, analysis and actions on data. There is also a Node.js server running on Heroku, providing the information of which is the zone where the bikes should be parked and serves a website with some visualisations of the data. Finally, IFTTT acts sending an email notification when a bike is parked outside the safe zone.

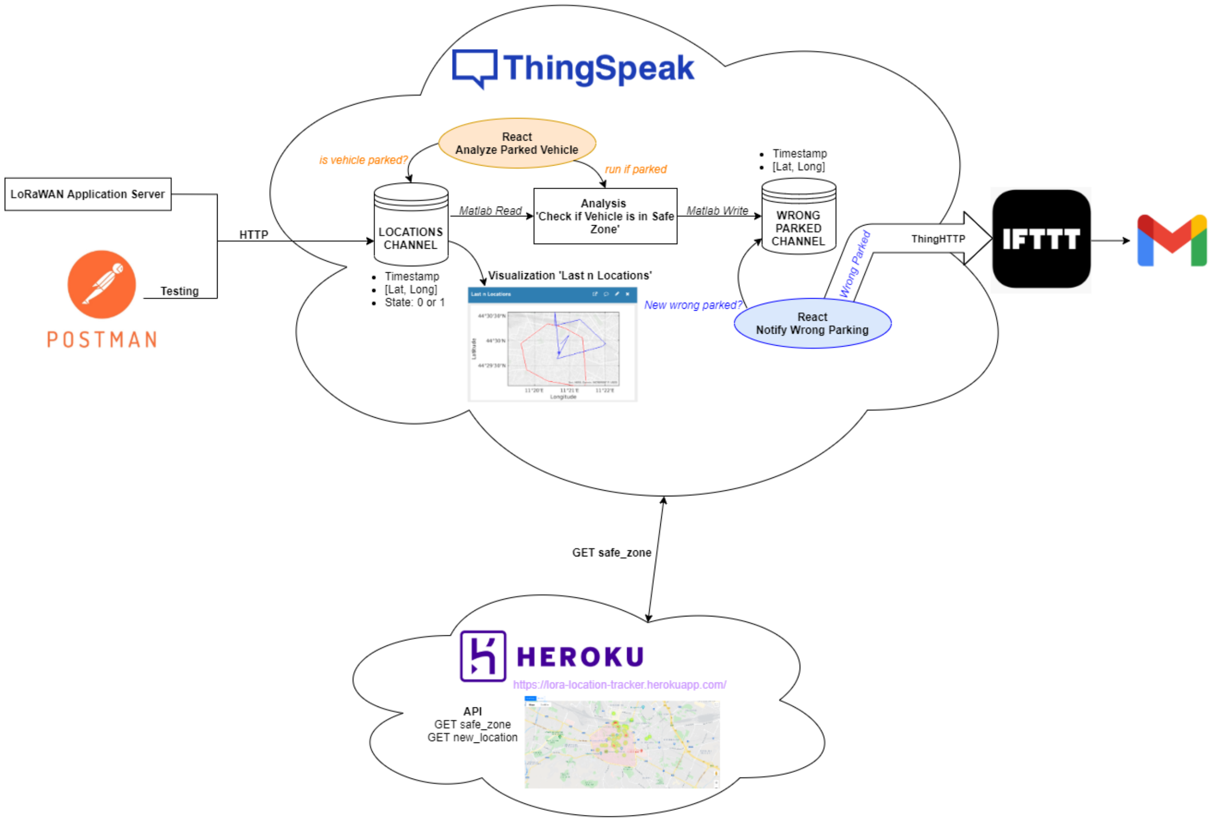

Figure 9: Cloud Diagram

The locations are sent by the Raspberry Pi to this ThingSpeak channel, through HTTP requests, and they are visualised as depicted in Figure 10. Then, every time a location is inserted, if the bike is parked a MATLAB Analysis is run to check whether it was parked inside the safe zone. In case the bike is outside, this event is saved in another channel, and an HTTP request is sent to IFTTT, which is set to send an email when it receives this request, as shown in Figure 12.

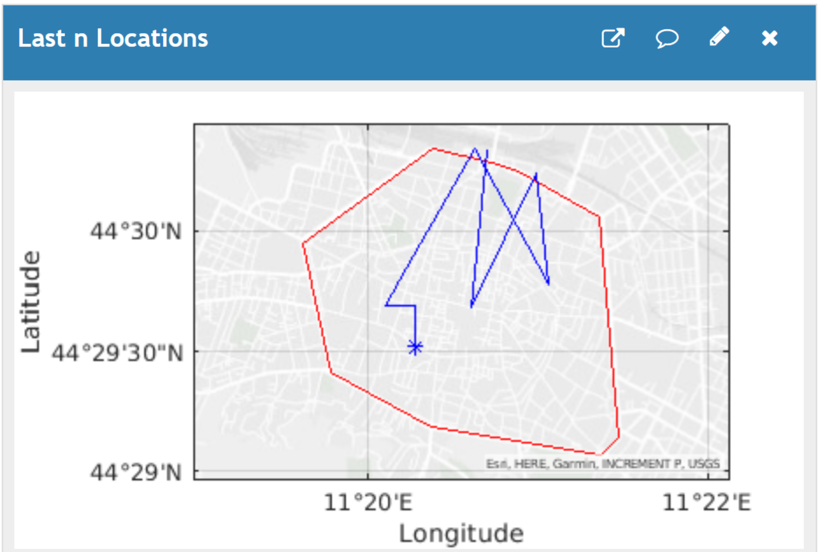

Figure 10: Last Locations Visualisation

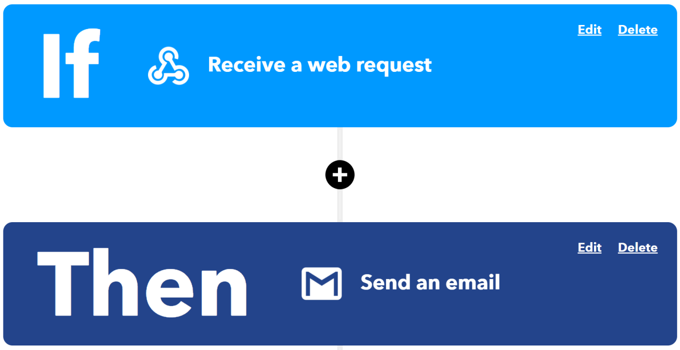

Figure 11: IFTTT service

Finally, the website gathers the locations and shows both a heatmap and the bike's route. For this visualisation, the google maps API was used. The heatmap is a really useful tool because, for example, if this was used for a bike provider that has to place its parking stations, the heatmap would be a useful tool to know where to place them, in the most frequented places.

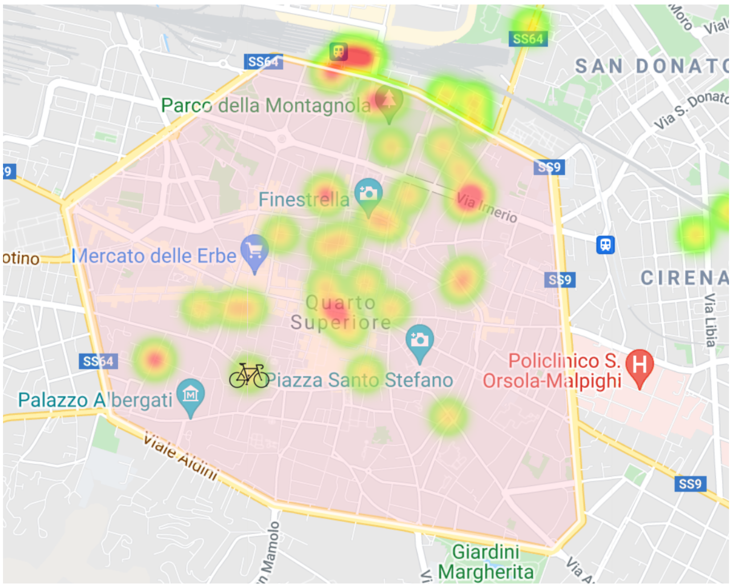

Figure 12: Website Heatmap

7 Measurements

Finally, measurements on-field were performed to evaluate the system's performance.

The following Graphs show the results obtained of two different parameters, Figure 13 depicts the values of RSSI in different distances while Figure 14 the SNR over distance.

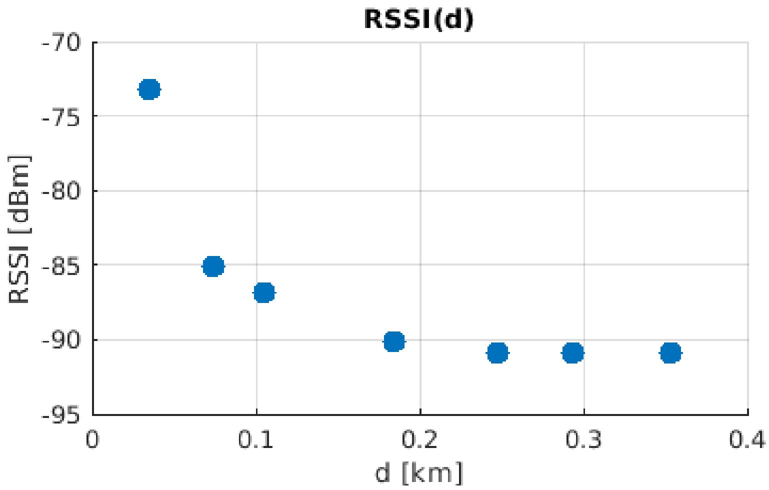

Figure 13: RSSI as a function of distance

The Received Signal Strength Indicator (RSSI) decreased with distance in a negative exponential way being -73dB at 30m and -92 at 350m, which was the maximum distance reached.

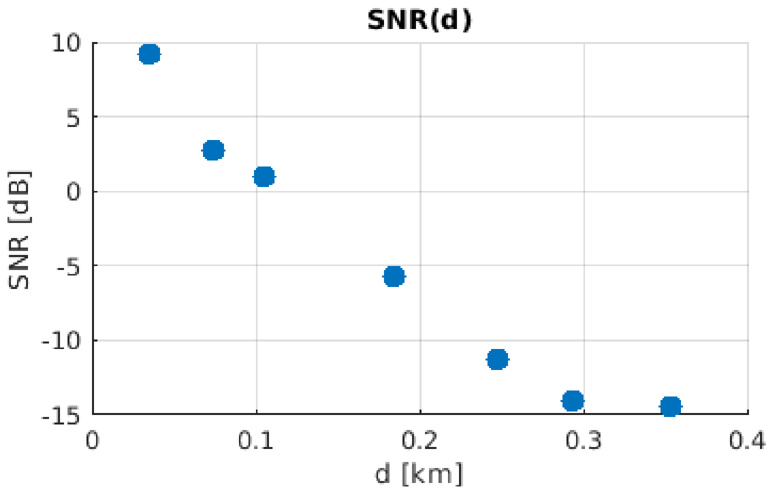

Figure 14: SNR as a function of distance

The Signal to Noise Ratio (SNR) is also a decreasing function of distance. The SNR received at 30m was 10dB while when going to a distance of 350m the value obtained i-15dB.

The range of the communication for three values of spreading factors was measured, the results are shown in Table 2. This values depend on the transmit power

SF	Range [m]
SF7	100
SF9	250
SF12	350

Table 2: Range for each SF

8 Conclusion

This report is concluded with satisfactory results, having been able to fully deploy an IoT pipeline based on the LoRaWAN protocol stack. A wide range of technologies was considered and real-life measurements were performed, in a crowded environment like the city centre of Bologna.

Future efforts should be mainly directed towards replacing the single-channel Packet-Forwarder with a multi-channel Gateway. This improvement will not only add the inherent capability of multi-channel reception, but will also give way to further enhancements like using The Things Network, since this is the technological trend followed nowadays for this type of technologies. In contrast, this implementation offers a limited but functional solution, at a reduced budget.