Rafael Joaquim Oliveira Rodrigues

Bachelor of Science in Electrical and Computer Engineering

Adaptive Geolocation of IoT devices for Active and Assisted Living

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Electrical and Computers Engineering

Adviser:	Ricardo Luís Rosa Jardim Gonçalves, Full Professor, NOVA University of Lisbon
Co-adviser:	João Filipe dos Santos Sarraipa, Invited Assistant Professor, NOVA University of Lisbon

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

July, 2020

Adaptive Geolocation of IoT devices for Active and Assisted Living

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Dedicated to my family

ACKNOWLEDGEMENTS

To begin with, I would like to thank both Professor João Sarraipa and Researcher Jorge Calado for giving me the opportunity and the means to to develop the investigation regarding the theme of this thesis.

"Stay hungry, stay foolish." - Steve Jobs

ABSTRACT

Recent developments in IoT devices and communication systems, have brought to light new solutions capable of offering advanced sensing of the surrounding environments. On the other hand, during the last decades, the average life expectancy has increased, which translates into a considerable rise in the number of elderly people. Consequently, in view of all these factors, there is currently a constant demand for solutions to support an Active and Assisted Living (AAL) of such people.

The presented thesis intends to propose a solution to help to know the location of IoT devices that may be assisting people. The proposed solution should take into consideration the risk factors of the target group at each moment, as well as the technical constraints of the device, such as its available power energy and means of communications. Thus, ultimately, a profile-based decision should autonomously be made by the device or its integrated system, in order to ensure the usage of the best geolocation technology for each situation.

Keywords: Active Assisted Living, IoT, Geolocation, Wearable Device

RESUMO

Desenvolvimentos recentes em dispositivos IoT e em sistemas de comunicação, trouxeram consigo novas soluções capazes de oferecer uma deteção avançada dos ambientes circundantes. Por outro lado, no decorrer das últimas décadas, a esperança média de vida aumentou, o que se traduz também num considerável aumento do número de pessoas idosas. Por conseguinte, perante o conjunto destes factores, existe actualmente uma procura constante de soluções de suporte a uma Active and Assisted Living desse grupo de pessoas.

A presente tese tenciona propor uma solução que ajude a conhecer a localização dos dispositivos IoT que possam estar a ajudar pessoas. A solução proposta deve ter em consideração os fatores de risco do grupo-alvo em cada momento e também as restrições técnicas do dispositivo, como a energia disponível e os meios de comunicação. Deste modo, em última instância, uma decisão baseada num perfil deve ser tomada autonomamente pelo dispositivo ou pelo seu sistema, para garantir a utilização da tecnologia de geolocalização mais adequada em cada situação.

Palavras-chave: *Active and Assisted Living*, *IoT*, Geolocalização, *Wearable Device*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

ABP	Authentication By Personalization.
AES	Advanced Encryption Standard.
AOA	Angle of Arrival.
API	Application Programming Interface.
BLE	Bluetooth Low Energy.
BPSK	Binary Phase-Shift Keying.
BSSID	Basic Service Set Identifier.
CRS	Cell Specific Reference Signal.
CSS	Chirp Spread Spectrum.
FCT/UNL	Faculdade de Ciências e Tecnologias, Universidade Nova de Lisboa.
GNSS	Global Navigation Satellite System.
GPIO	General Purpose Input/Output.
GPRS	General Packet Radio Service.
GPS	Global Positioning System.
GSM	Global System for Mobile Communications.
GUI	Graphical User Interface.
ICIST	International Conference on Information Society and Technology.
IoT	Internet of Things.
IP	Internet Protocol.
ISM	Industrial Scientific and Medical.
JSON	JavaScript Object Notation.
LoRa	Long Range.

LoRaPHY	LoRa Physical.
LoRaWAN	LoRa Wide Area Network.
LPWAN	Low Power Wide Area Network.
LTE	Long Term Evolution.
M2M	Machine to Machine.
MQTT	Message Queuing Telemetry Transport.
OTAA	Over The Air Authentication.
OTDoA	Observed Time Difference of Arrival.
PRS	Positioning Reference Signal.
PwD	Person/People with Dementia.
QoS	Quality of Service.
REST	Representational State Transfer.
RSSI	Received Signal Strength Indicator.
SIM	Subscriber Identification Module.
SNR	Signal-to-Noise Ratio.
SSID	Service Set Identifier.
SWOT	Strengths Weaknesses Opportunities Threats.
TDoA	Time Difference of Arrival.
ToA	Time of Arrival.
TOA	Time On Air.
ToF	Time of Flight.

CHAPTER

1

INTRODUCTION

This chapter aims to present, throughout its sections, an introduction to this thesis. Furthermore, the research question is identified and an hypothesized solution is indicated - being its main purpose to contribute to the initial stages of development of this research. Lastly, the outline for this thesis is presented.

1.1 Motivation

In the coming years, a high number of Internet of Things (IoT) devices are expected, some forecasts point to more than 20 billion. These devices are connected to the Internet and to each other, as well as to people around them. On the other hand, an issue that is expected to occur within the following years, directly related to a population that is ageing [1, 2] as a result of the currently higher life expectancy, is the growth in the number of diseases such as dementia. In fact *"Dementia is an age-associated impairment that could affect about 135 million people worldwide by the year 2050"* [3].

As verified by the numbers revealed in the 2019 Organisation for Economic Co-operation and Development (OECD) report, it is estimated that almost 20 million people in OECD countries were diagnosed with dementia during that year. If the present trends continue, that number will more than double by 2050, achieving nearly 41 million diagnosed people. Age is still the biggest risk factor for dementia: across the 36 OECD countries, average dementia prevalence increases from 2.3%, among people aged 65-69, to almost 42% among people aged 90 or more. This means that, as countries age, the number of people living with dementia will also increase, particularly as the proportion of the population over 80 rises.

Consequently, nowadays countries with some of the oldest populations have the highest prevalence of dementia. Across OECD countries, on average, 15 people per 1000 population are estimated to have dementia and by the year 2050 the prevalence of dementia will be more than

20 people per 1000 population. Concerning Portugal in particular, within a 30 year spam, more than 1 in 25 people will be living with this condition [4]. However, the long-term care workers in Portugal per 100 people aged 65 and over is 0.5 a ratio that represents 10 times less than the OECD median [5].

Thus the author of this thesis, and, the Carelink [6] project in which it is integrated try to join both factors with the aim that the increasing number of IoT can have a positive effect on the decreasing of the risks associated with dementia.

The CARELINK-AAL/0001/2016 - Living with Dementia, is a European project targeted at Person/People with Dementia (PwD), that is a subgroup of the Alzheimer neurological disease. This condition causes people to forget where they are, leading them to wandering behaviors, that can put their lives in danger. The purpose of this project is to develop a solution for such people, the solution is based on a wearable device and a platform, the wearable device communicates with the platform through several types of communication methods, and in the platform there are several services, being one service the tracking service.

The solution developed by the author is an adaptive geolocation system that combines the different geolocation methods, resulting in an overall location that provides the best results in terms of precision and performance-wise. This will try to help people today with dementia, by trying to reduce the number of deaths associated with wandering behaviours, giving a better quality of life for the patient and his caregiver.

1.2 Research Method

For carrying out this investigation, the approach taken was divided into 3 different steps, as shown in Figure 1.1. The first is regarding to research, which comprises the literature review, compilations of similar works, and debugging. The next phase is development, where the previous knowledge acquired from the research is used in practice and the tests are defined.
The last phase is testing, that consists of using the results from the development phase and applying them in real life. After testing, the iteration is repeated until the final result is achieved. When the final result is achieved, and the results could help other researchers, the work is published.

Figure 1.1: Work Approach

1.3 Research Question

The purpose of this research is to evaluate if a wearable device, with dynamic geolocation, based in various system factors such as available communication and power management, can contribute to an improved localization process and position accuracy. The main issue is to be able to know the location of such devices. Currently, there are solutions based on Global Navigation Satellite System (GNSS) and in this thesis will work as a reference, for a GNSS free alternative. Ergo, the application of an adaptive geolocation solution for elderly people raises the following research problem:

- **RQ:** What is the impact of Low Power Wide Area Network (LPWAN) enabled solutions in wearable devices, especially those dedicated to people with dementia?

1.4 Hypothesis

Can LPWAN enabled solutions, helping in perform the location of the person, while saving battery, reduce the overall cost, and comply with the technical constraints of wearable devices. This may also have also the possibility to dynamically choose, the best location available in that moment.

1.5 Dissertation Outline

This dissertation is divided in 6 chapters, which are presented below:

- **Chapter 1 - Introduction**: This chapter describes the purpose of the dissertation and the motivation behind the research project, as well as the adopted research method. In addition, it presents the research question that motivated this work and its hypothesis, in order to solve this question;

- **Chapter 2 - State of the Art**: Throughout this chapter, the State of the Art is outlined. Overall, it represents the collected information necessary to build a model capable of validating the previously defined hypothesis. In order to do so, first it takes place an overview of the current solutions for people with dementia. Next, are described several LPWAN that can be used for the communication of geolocation data. After that, are shown the different Geolocation Algorithms and methods. Finally, it is described how LPWAN can be used for adaptive geolocation and the similar work already done is introduced.

- **Chapter 3 - Adaptive Geolocation**: In this chapter it is explained in detail the approach to the proposed model. To do so, the architecture of the system, where the model is working, is presented as well as the modules to be implemented

- **Chapter 4 - Implementation**: This chapter concerns the description of the implementation of each part, along with the technologies and methodology used throughout.

- **Chapter 5 - Results**: This fifth chapter, presents the methodology used for the development of the solution along with the proposed use cases, as well as the practical field tests conducted for the project.
- **Chapter 6 - Conclusions**: This dissertation is concluded with the presentation of the final thoughts and remarks, as well as the hypothesis presented for solving of the research question and problems previously identified. At the end of this chapter, takes place the subsection "Future Work" 6.1, with the proposal for possible continuation work.

CHAPTER 2

STATE OF THE ART

In this chapter, the state of the art on the main research subjects is presented. Firstly, an overview of current Solutions for People with dementia is described, followed by a review of the main IoT LPWANS and a description of the current trends of the supported emergent LPWANS technologies, complemented by a comparison table. Afterwards, are stated the current Geolocation solutions, along with the advantages and disadvantages of the application of these technologies on the Active and Assisted Living. The related work on the research topic concludes this section.

2.1 Solutions for People with Dementia

The solutions available for wearable devices, used as trackers for people with dementia, still to this day bear an issue being that continuous tracking devices suffer from reduced battery life. This is caused by the power consumption of the GNSS module, and the use of cellular networks, such as Global System for Mobile Communications (GSM), that are not optimised for the size and power efficiency needed for this particular use case.

These devices are sought by caregivers to reduce one of the critical risks of people with dementia: wandering onto unsafe areas. At the moment, the best device on the market capable of tracking people with dementia has less than 10 hours of battery usage, and for that it uses a large battery. The problem with this sort of devices is that it requires from the patients or caregivers a daily battery charging. The Figure 2.1, is based on the one from [7], and the devices in June of 2020 are available at the following links TicTocTracker, BlueWater watch, mCare watch, SmartSole. In Figure 2.1 is possible to observe a comparison between the mentioned devices, demonstrating the battery duration in live tracking mode and standby.

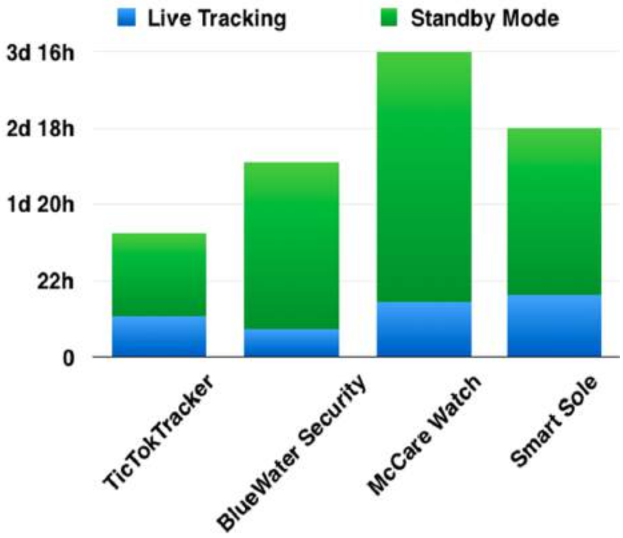


Figure 2.1: Device Battery Comparison

The Table 2.1 shows a comparison between the above-mentioned devices, where the price of the product is presented. In the technical part, a comparison of the geolocation technique and communication method is also shown. By reading this table, is possible to understand that: the wearable type of these solutions are always similar, with the exception of the Smart Sole; geolocation is always based on Global Positioning System (GPS); the communication is always cellular, except in the TicTocTracker product, were WiFi is also an option. The conclusion is that all the solutions are somehow similar, expensive, and not optimized for power efficiency.

Company	TicTocTrack	Bluewater	mCare	GTX
Product	TicTocTracker	Bluewater watch	watch	SmartSole
Price	182€ Device + SIM Plan	544€ Device 32€/month	453€ Device 23-63€/month	272€ Device 23€/month
Technical:				
Geolocation	GPS	GPS	GPS	GPS
Comunication	3G / WiFi	NaN	4G	GSM
Wearable type	watch	watch	watch	sole

Table 2.1: Solutions for PwD comparison

2.2 LPWAN

"By the year of 2025, up to 75 billion devices would be connected in Internet-of-Things (IoT), with a potential economic impact of around 11.1 trillion $ a year" [8]. The need for an LPWAN emerges when Machine to Machine (M2M) communication becomes a necessity and other wireless networks, for example, the ones with a short-range (Bluetooth, RFID, Wifi or Zigbee) are not a good fit for our application, LPWANs are suited to connect devices that only need to send a small amount of data over a long distance (100m to 15km or more). The key features in all of the different LPWAN technologies will be a long-range and low power consumption. For achieving this the data rate will also be low, as it is possible to observe at Figure 2.2.

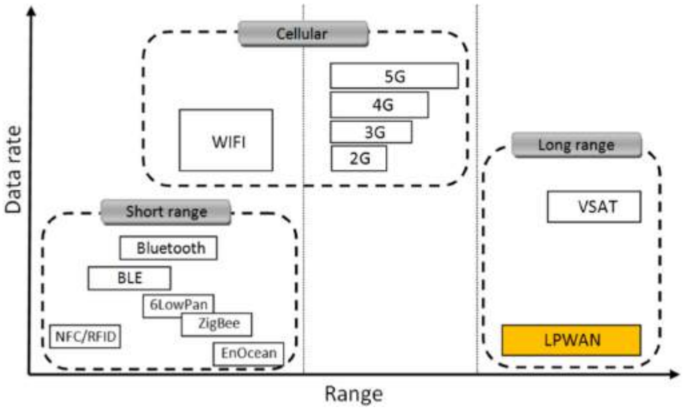


Figure 2.2: LPWAN Position in Range VS Data rate [9]

There are two main types of a network configuration for the LPWAN. First Mesh Topology, where all nodes are connected in a unstructured way and can communicate in order to distribute data amongst each other. Although this is not entirely true for some implementations of this topology, one of the most well known is Zigbee. In this implementation there are three different roles: the coordinator, which connects to the internet; the end nodes and, in between, the mesh routers (these cooperate to relay the message to the coordinator).

Secondly, the Star Topology network is the widely used approach. The end nodes of a star network are connected directly to the gateway and, in its turn, this gateway is connected to the internet - similarly to WiFi. In a star topology when there is the need for better coverage and reliability there is always the possibility to add more repeaters [10]. Besides the network configuration, LPWANS can be further divided into two groups: Licensed and Unlicensed, depending on which part of the wireless spectrum they work on [11].

2.2.1 LoRa

LoRa is an LPWAN, initially developed by the French company Cycleo and later acquired by Semtech. It is a semi-proprietary standard, and is composed of two major components.

First Long Range (LoRa) represented in orange at figure 2.3, is the physical layer and it is the proprietary part of the stack. This layer is based on Chirp Spread Spectrum (CSS) modulation which, in comparison with other radio technologies, maintains the same low power characteristics as Frequency Shif Keying (FSK) modulation, but with a higher communication range. The CSS has been in use for decades because of the long distances that can be achieved and the immunity to external interference. This technology operates in the Industrial Scientific and Medical (ISM) bandwidth, in Europe that is 868 MHz. The second component, named LoRa Wide Area Network (LoRaWAN) (represented in the color blue on Figure 2.3), is responsible for defining the communication protocol and the system architecture for the network. The protocol and network architecture have a role in determining the battery lifetime of the end device; the quality of service; the network capacity; the security and the variety of applications served by the network.

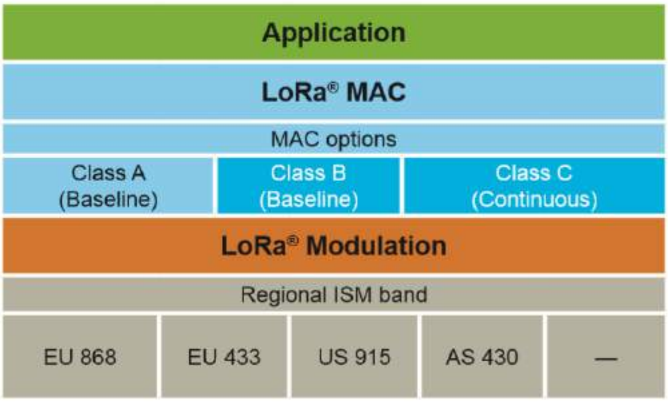


Figure 2.3: LoRa Stack [12]

In terms of specifications, LoRa as data rates from 300 bps to 50 kbps. This occurs because LoRa has an adaptive data Rate, that is dependent of the spreading factor in use, following the next equation:

$$R_b = SF * \frac{1}{\frac{2^{SF}}{BW}} \qquad bits/sec \tag{2.1}$$

In the previous equation:

- R_b - Bitrate
- SF - Spreading Factor
- BW - Bandwidth

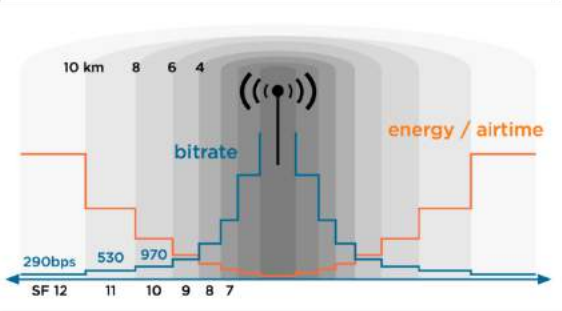


Figure 2.4: Spreading Factor Vs Range bitrate Energy and Time

In Figure 2.4, it is possible to identify six options for the Spreading factor, from SF 7 to SF 12. Although new LoRa chips, called Semtech SX1262, have a spreading factor ranging from 5 to 12, every level represents a compromise between data rate and range. Each time the SF is incremented, the time on air doubles for the same quantity of data, thus decreases the data rate but increases the signal resistance to interference noise. The optimal value of this factor should take into consideration the available bandwidth and the Signal-to-Noise Ratio (SNR).

Network Architecture

LoRa, compared to the several already deployed networks, does not utilize a mesh network architecture. Because in a mesh network, the single end-nodes forward the information of other nodes for better communication range and for a lower cell size of the network. While this will increase the range, it will also add more complexity, since nodes will receive and forward messages that are destined to other nodes, that most of the times are irrelevant for the ones forwarding the said message and that could be in a sleep state, saving battery. This results in a reduced network capacity, and, ultimately, a reduced battery lifetime.

LoRa, uses a long-range star architecture (Figure 2.5) that increases the battery when long-range connectivity needs to be achieved. In a star topology, there is a central gateway, usually connected to a power supply, in charge of receiving any uplink message and sending all the down-link messages [13].

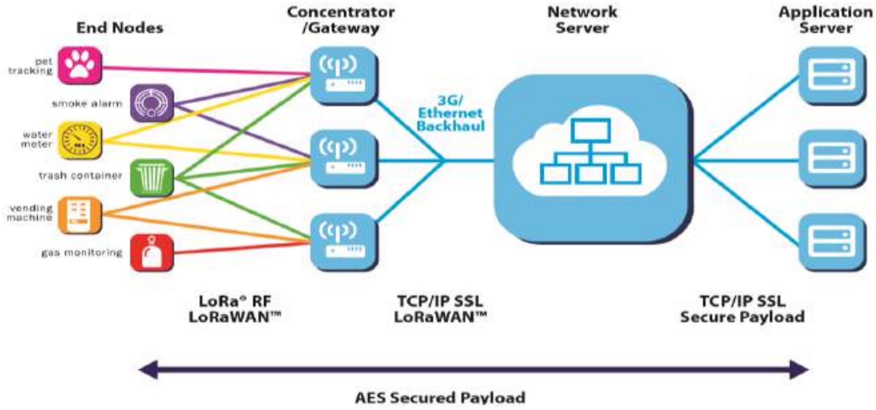


Figure 2.5: Network Architecture [13]

Security It is a major concern for any kind of LPWAN the implementation of security. LoRaWAN utilizes two layers of security, as represented in Figure 2.6 - the blue color corresponds to the network and the pink color to the application. In the same figure is also represented the two types of activation: first the Over The Air Authentication (OTAA) and secondly the Authentication By Personalization (ABP). The network security key ensures the authenticity of the node in the network, while the application layer of security ensures that the network operator can not have access to user application data. The security algorithm is the Advanced Encryption Standard (AES) and is used with the key exchange using an IEEE EUI64 identification [14].

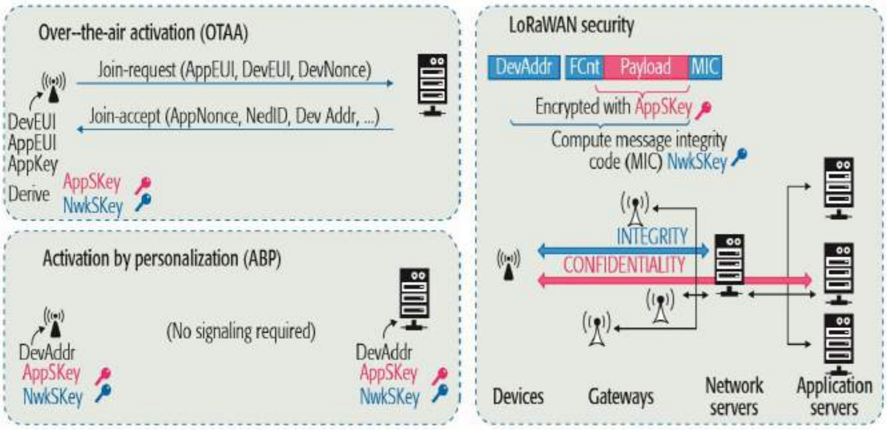


Figure 2.6: LoRaWAN Security [14]

LoRa Classes: There are 3 types of classes for LoRaWAN end devices, as represented in Figure 2.7. The different classes exist because of the different applications and requirements of the end devices. The classes trade-off network downlink latency for battery lifetime.

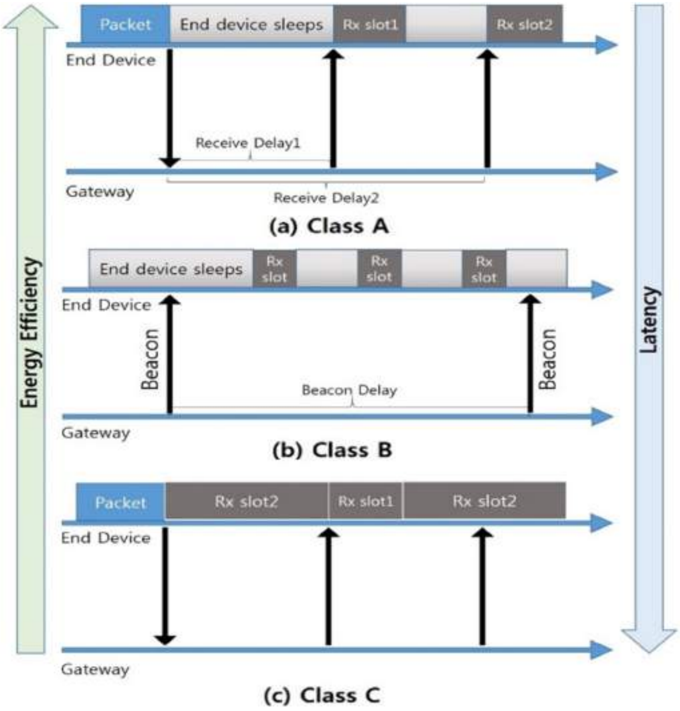


Figure 2.7: LoRaWAN Classes [15]

- **Class A**- supports bi-directional communications, after an uplink transmission, two downlink receiving windows are open. The scheduling of the transmission slot is decided by the end device and it is based on a random time basis (similarly to ALOHA protocol). This class is the one with better energy efficiency and it is ideal for use cases where downlink communication from the server is only required shortly after the device sent an Uplink communication. In case the network server chooses to start a communication with the end-device, at another period of time, this server will have to wait until the next uplink transmission. This kind of devices are usually battery powered sensors [16].

- **Class B**- in the complement of the Class A random receive windows during the downlink period. Class B devices can open, an extra receiving window at a specific scheduled time. The duration of this window is defined by a beacon frame, sent by the gateway on a periodical slot of time called a "beacon delay". After receiving a beacon, a receiving window, called "ping slot" is open. The Class B devices allow the server to control when they should listen. These devices are normally actuators powered by batteries.
- **Class C**- opens the same two receiving windows of class A, plus another one that is almost continuously open, that only closes during an uplink transmission. Therefore this class is the one with lower latency. On the other hand, it is also the more power demanding and should be used for an application that has a high amount of energy. The end devices using this class are typically actuators connected to a power supply.

In short, all classes allow bi-directional communications. Class A supports downlink communications after an uplink transmission, wereas class B enables downlink scheduling and class C is always available for downlink communication, except when a device has to send an uplink message. The last difference between classes is that A only supports Unicast messages, while the classes B and C afford Unicast and Multicast.

To conclude, in the following Table 2.2 is a comparison between 4 devices using the LoRa technology. These devices as of June 2020 are available at the following: links Fipy, The Things Node, TTGO LoRa32, CubeCell. From the data in Table 2.2, it is possible to observe that the first device has a different programming language than the others and that they all have distinct LoRa radio modules, as well as different microcontrollers, where the processing power and memory are also different. These are the characteristics necessary for the correct choice for each use case.

Company	Pycom	The Things Network	LILYGO	Heltec
Product	Fipy	The Things Node	TTGO LoRa32	CubeCell
Price	54€	60€	15€	12€
Technical :				
Microcontroller	Esp32	ATmega32U4	Esp32	ARM Cortex M0+ Core
LoRa module	Semtech Sx1272	Microchip RN2483	Semtech SX1276	Semtech Sx1262 (Latest)
Programing	Python	Arduino IDE	Arduino IDE	Arduino IDE

Table 2.2: LoRa devices comparison

2.2.2 SIGFOX

SIGFOX LPWAN consists in a cellular like type of network, where the SIGFOX company sets up its own antennas and is capable of offering end-to-end connectivity to IoT devices, used for applications with a low-throughput. The transmission occurs in the unlicensed part of the spectrum, specifically in the 868 or 915 bands, depending on the country where they operate. SIGFOX wireless systems uses Binary Phase-Shift Keying (BPSK) as a type of UNB (Ultra-Narrow band) radio modulation, allowing communication with very low noise levels and efficient bandwidth usage.

This LPWAN is half-duplex and it makes possible to send very small amounts of data (12 bytes uplink and 8 bytes downlink), alongside a bit rate of 100 or 600 bps, depending on the region. The long-range achieved by SIGFOX is the result of very long and very slow messages, for example, a message with a 12 byte payload, has 2.08s air time with a rate of 100 bps.

One other aspect of this technology is that, although the chipsets are open, the network is closed. With this the use is limited to the SIGFOX proprietary network, which is already deployed around the world (Figure 2.8). However, it is not possible to create a private network and, in order to use the existing one, a subscription plan is needed.

Figure 2.8: Sigfox Coverage January 2020 [17]

The SIGFOX network has a horizontal architecture which is divided into two layers: the network equipment and the SIGFOX Support Systems. The first is composed by the base stations, responsible for receiving and sending messages from the end devices and delivering them to the SIGFOX cloud. These messages are transferred between the two layers through a backhaul, which usually uses DSL and 3G or 4G as a backup. When the two options are not available, satellite can also be used as an alternative backup.

In the cloud portion of SIGFOX Support System, the back-end servers handle the message processing. The core network servers monitor the current status of the network and are responsible for managing the base stations. There are replicated messages that arrived on the core network, but only one should be stored. In the storage part, the messages are stored in two locations, first in the metadata, which can be used for building services, and secondly in a customers message storage, so that the customers can retrieve them for later use.

Finally, the web interface and the Application Programming Interface (API) allows messages access. This is done through a web browser or the with use of Representational State Transfer (REST) API. In the end, the messages are pushed downlink to the device, as it is possible to observer in figure 2.9.

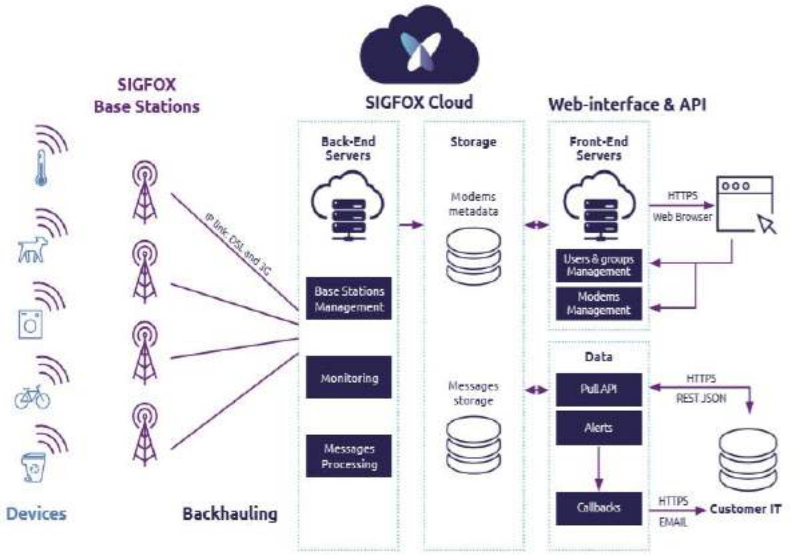
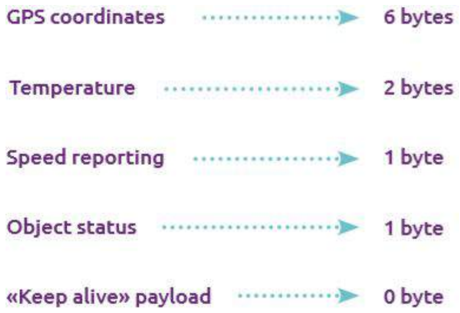


Figure 2.9: Sigfox Architecture [18]

The message payload size goes from 0 (called "keep alive messages") until 12 bytes while operating in an uplink. A 12 byte message is small but its enough to transfer sensor data, the status of devices, GPS coordinates or even application data as seen in Figure 2.10.
Meanwhile, in downlink operation messages have a fixed payload of 8 bytes.



GPS coordinates	6 bytes
Temperature	2 bytes
Speed reporting	1 byte
Object status	1 byte
«Keep alive» payload	0 byte

Figure 2.10: Sigfox Payload Size [18]

This is enough for triggering an action, managing a device or setting parameters remotely. The actual regulation in Europe dictates that amount of time Sigfox can occupy the public spectrum is 1% of the time (approximately 30 seconds of transmission time per hour), this translates into an average of 140 uplink and 4 downlink messages per day.

The key difference between SIGFOX and other LPWAN technologies is the modulation in use. SIGFOX uses the modulation UNB, that consists in transmitting a signal over a very small bandwidth, resulting in a signal with high PSD (Power spectral density). Implicitly, the energy required to pass the noise floor will be lower. In addition, this type of signals have a natural resistance to interference, which can be an advantaged in unlicensed and usually crowded bandwidths. Sigfox draws on these properties since its using a 192KHz wide bandwidth in Europe, but the messages are only 100Hz [18].

Even though UNB favourable effects on link budget, its properties raise other second effects because signals with small bandwidth are especially affected by the Doppler effect.The Doppler effect consists in small shifts in frequency, caused by the variation of the relative distance between a receiver and the source. Over a certain amount of time, these shifts can become bigger than the signal bandwidth itself, increasing the possibility of message collision, as well as hinder its detection/demodulation [19].

To solve the described issue, SIGFOX uses a random access feature. This feature consists of unsynchronized transmission between the network and the devices. In uplink operation the device emits a message on a random frequency, that is later followed by two copies transmitted on different frequencies and time, while the base stations monitor all the spectrum and search for UNB signals. This feature also slightly surpasses the reliability problems occurring do Sigfox lack of message arrival acknowledgement [9]. The downlink messages have to be initiated by the end device, the frequency in use is the same as the first uplink message plus a known delta. There is a delay of 20 seconds between the first transmitted frame and a reception window, being that this last one only lasts for a maximum of 25 seconds [18].

Finally, the high energy efficiency offered by SIGFOX relies on two main factors. The first one, as mentioned above, consists in the absence of pairing during transmissions, which means that no sync messages are exchanged between the end device and the base station, resulting in energy saving. The second concerns the end device's very low power consumption while idling; since this occurs for 99% of the time, it also contributes heavily to ensure long battery life.

To conclude, in the following Table 2.3 a comparison between 4 devices using the Sigfox technology is presented, mentioning the price and the technical features. These devices as of June 2020 are available at the next links: Fipy, B-L072Z-LRWAN1, SFJK-API-1-GEVK, MKR FOX 1200. From the data in Table 2.3, is possible to observe that programming interface are different, as well as the microntroller. These are two factors to consider while chosing a SIGFOX device.

Company	Pycom	STM	ON Semicondutor	Arduino
Product	Fipy	B-L072Z-LRWAN1	SFJK-API-1-GEVK	MKR FOX 1200
Price	54€	44€	70€	35€
Technical:				
Microcontroller	Esp32	STM32L072CZ ArmCortex M0+	AX-SFEU	SAMD21 ArmCortexM0+
Programing	Python	Arduino IDE	AX8052-IDE	ArduinoIDE

Table 2.3: SIGFOX device comparison

2.2.3 NB-IoT

NB-IoT, as the name suggest, is a narrow band cellular technology designed for the IoT context. It was standardized by the release 13 of Long Term Evolution (LTE), done by the third generation partnership project (3GPP), and it will be a part of the IoT scenario in the next decade, offering: low power consumption to low-cost hardware; improved indoor coverage; low delay sensitivity and the ability to handle a multitude of low-throughput devices [20].

The Cellular network protocols, such as General Packet Radio Service (GPRS), are already capable of performing M2M communication, but were not initially designed to deal with power constrains or to handle small message transmissions [19]. NB-IoT solves part of the problem because it is particularly suitable for the refarming of Global System for Mobile Communications (GSM) channels, allowing, in this way, the use of already established cellular networking infrastructures for long-range IoT application.

NB-IoT, opposed to the other technologies described above and being a cellular specification, operates in the licensed part of the frequency spectrum (700,800 and 900MZ), and it was designed in a way that enables it to coexist with LTE and GSM. The NB-IoT is a shrink version of LTE protocol, it discards any redundant characteristics for the IoT context and enhances the useful ones. It occupies a frequency bandwidth of 200 KHz, which is one resource block (RB) in a GSM and LTE carrier wave [9]. NB-IoT has three different modes of operation:

- **Stand-alone** - Using the existent GSM channels. On both sides of the spectrum is an unused 10KHz interval between each resource block(Figure 2.11a);
- **In-band** - Using the resource blocks present in a normal LTE carrier(Figure 2.11b);
- **Guard-band**- Taking advantage of the unused resource block within an LTE carriers guard band(Figure 2.11c).

(a) Standalone (b) In-band (c) Guard band

Figure 2.11: NB-IoT Operation modes [21]

NB-IoT uses QPSK and OFDMA modulations during downlink operations, and BPSK or QPSK during uplink ones. Transmission rates may go from 160 to 250 k/bits per second, while in uplink transmission, with the use of a single sub-carrier, the maximum speed will be 200 k/bits per second.

Concerning the network architecture, it follows a common Internet of Things architecture, consisting of 5 parts, represented in Figure 2.12.

To further detail, NB-IoT terminals comprise the sum of all end devices into the system and have access to the network as long as the correct Subscriber Identification Module (SIM) card is installed.

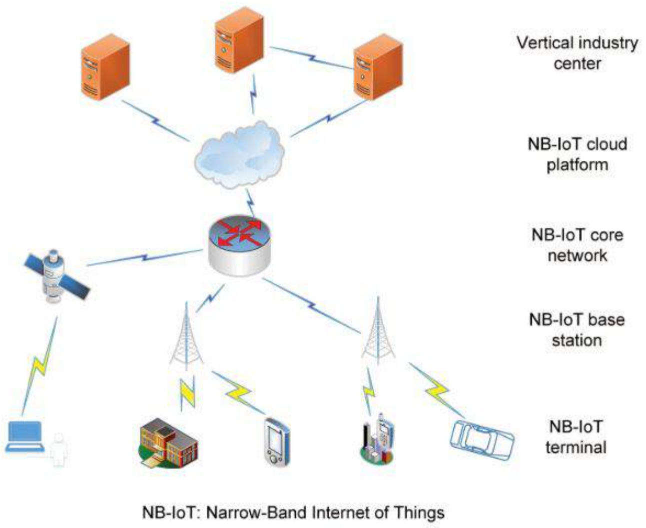


Figure 2.12: NB-IoT Network Architecture [22]

The base stations refer to pre-existing nodes that have already been deployed by telecom operators. Usually, these support all three types of deployment modes shown before.
Core network behaves like a bridge, enabling connections between the base station and a cloud platform. The cloud platform can process various services, then forwards the outputs to the vertical industry center whose function is up to the client or directly to the NB-IoT terminal.
Typically, the Vertical Industry layer has Graphical User Interface (GUI), for showing the data collected by the system, also it has control mechanisms for actuators or another device embedded into the terminal layer [22].

To conclude, in the following Table 2.4 was done a comparison between 4 devices using the NB-IoT technology. These devices as of June 2020 are available at the following links: Fipy, MKR NB 1500, SARA R412m, AXM0F243-868-1-GEVK. From the data in Table 2.4, is possible to observe that first two devices cost half of the other two. The microncontroller is always an ARM M0, except for the first device that is based on a Esp32. This means that the first device has microntroller with more cores (2 instead of 1) and, higher clock frequency but fewer General Purpose Input/Output (GPIO) pins. This microcontroller supports real multi-thread code, but is able to control less things. These are some of the characteristics needed for the correct choice for each use case.

Company	Pycom	Arduino	SODAQ	On Semicondutor
Product	Fipy	MKR NB 1500	SARA R412m	AXM0F243-868-1-GEVK
Price	54€	67€	115€	130€
Technical:				
Microcontroller	Esp32	SAMD21 ArmCortexM0+	SAMD21 ArmCortexM0+	ArmCortexM0
Programing	Python	ArduinoIDE	ArduinoIDE	AX8052-IDE

Table 2.4: NB-IoT device comparison

2.2.4 LTE-M (Cat-M1)

LTE-M [23], officially know as LTE Cat-M1, was first introduced in release 13 of the Third Generation Partnership Project (3GPP), as a response of the increasing interest in LPWAN solutions that can use standard LTE connectivity while answering to requirements and constraints of the LPWANS. LTE Cat-M1 is usually viewed as the second generation of LTE chips designed for IoT. It fulfils the cost reduction and power consumption efficiency that Cat-0 set the stage for. By using a maximum bandwidth of 1.4 MHz, as opposed to 20 MHz for Cat-0, Cat-M is ideal for LPWAN applications such as smart and wearable meters, where is only required to transfer a small amount of data.

Concerning the specifications which define Cat M1, there were features and functions improved in relation to the previous releases, usually referred to as Power Saving Mode (PSM), eDRX and Coverage Enhancement Mode A and B. The already established LTE timers are still utilized and other new timers were defined for supporting all of these new features. LTE CAT-M1 will allow for: Internet Protocol (IP) over Control Plane, this can be done both in UDP (User Datagram Protocol) or in TCP (Transmission Control Protocol); IP over User Plane (both UDP and TCP), including the original User Plane and an optimised version of this User Plane; Non-IP over Control Plane, from 3GPP Rel-13 using the Control Plane CIoT (cellular IoT) EPS optimisation with Non-IP PDN type and Non-IP over User Plane, including User Plane Optimised, and User Plane Original, from 3GPP Rel-13 using the User Plane CIoT EPS optimisation with Non-IP PDN type [23]. The minimum features required for the balance of roaming service and perform the power optimisation are:

- **PSM-** Power Save Mode;
- **LTE-M** Half-Duplex Mode;
- **eDRX-** Extended Discontinuous Reception;
- **CMM-** Connected Mode Mobility;
- **SMS-** Short Message Service;

Non-IP PDN type allows an EPS UE to transfer data without the need of operating an IP stack and obtaining an IP address. “Non-IP” transport is requested by the UE in a PDN Connectivity Request as part of an Attach Request or separately. By selecting “PDNtype = Non-IP” the possible values are IPv4, IPv4v6, IPv6 or Non-IP. Two mechanisms provided in HSS are currently defined for the delivery of Non-IP data to the Service Capability Server / Application Server (SCS/AS) [23]:

- Delivery utilising SCEF;
- Delivery making use of a Point-to-Point (PtP) SGi tunnel.

The NB-IoT, mentioned in the previous section 2.2.3, and LTE-M are both from the same release and, although they are in some aspects similar, there are still some differences between the two. One of the differences is the region of deploying in the world. The deploying of NB-IoT in the US will be extremely hard because of the ubiquity of LTE. In the end, it often comes down to the requirements of the use case, that is, NB-IoT is best suited to static uses, for example smart meters, while LTE-M can benefit applications that need roaming such as vehicles or drones. In this regard, LTE-M has some notable advantages. First, it has much higher data rates, which is important for data-rich use cases unlike NB-IoT, has relatively simple front-end.

In short, LTE is primarily a technology used in the North America. Although in Figure 2.13 is possible to observe the full coverage, there are other limitations to consider. For one, the power efficiency is still under evaluation with LTE-M. There are also the licensing issues to have in consideration. In addition, is likely to see major North American telecom companies pushing LTE-M, since these companies already invested billions in LTE technology. By contrast, the rest of the world, where the GSM spectrum is the norm, is expected a preference for the non-LTE NB-IoT protocol [24].

Figure 2.13: LTE-M Coverage as of January 2020 [25]

To conclude, in the following Table 2.5 is presented a comparison between 4 devices using LTE-M technology. These devices as of June 2020 are available at the following links: Fipy, MKR NB 1500, LTE CAT M1SARA-R4, XBeeLTE Cat 1. The table 2.5, shows that the first two devices (on the left) have a microcontroller, whereas the last two do not have and are being only sold as a shield. The last device has a built-in MicroPython support with 24KB RAM and 8KB Flash. These are the features needed for the correct choice for any use case.

Company	Pycom	Arduino	SparkFun	Digi
Product	Fipy	MKR NB 1500	LTE CAT M1 SARA-R4	XBee LTE Cat 1
Price	54€	67€	73€	93€
Technical:				
Microcontroller	Esp32	SAMD21 ArmCortexM0+	Shield	Shield
Programing	Python	ArduinoIDE	ArduinoIDE	Python

Table 2.5: LTE CAT M1 Device Comparison

2.2.5 Other LPWAN Technologies

Since there are other LPWANs technologies that are not present in this thesis analysis of hardware, a brief overview is presented below.

Weightless

Weightless [26] is an open standard and, for this reason, the hardware cost is lower compared to the other standards. Weightless defines three types of classes, similar to LoRaWAN, that classes share a transmit power of 17 dBm, unlimited number of devices and the possibility for roaming. Those classes are presented below:

- **Weightless-P** offers bi-directional communication, with Quality of Service (QoS). It has 12.5 KHz bandwidth on sub-GHz frequencies, the data rate goes from 200 bps to 100 Kbps, a packet size of a minimum of 10 bytes and the range up to 2 KM. Weightless-P is appropriate for private networks and uses cases were bi-directional traffic is necessary.
- **Weightless-N** is an ultra-narrowband system, similar to the SIGFOX system, and is focused for sensor based networks, its range is up to 3 Km. It makes use of the ultra-narrow band, approximately 200 Hz with 100 bps uplink with a packet size of a maximum 20 bytes, but there is the capability of downlink communication.
- **Weightless-W** makes use of the TV white space, with a 5 MHz channel width and up to 5 Km range but, compared to the other two, it has the lowest battery life. The other disadvantage is that frequency in which it operates may vary from city to city. The packet size of this version is the same as the "P" variant, 10 byte minimum, but a comparatively higher data rate, from 1 Kbps to 10 Mbps

Weightless uses Direct Sequence Spread Spectrum to improve the range and as a trade-off decrease of the data rate. The DSSS works by multiplying each transmitted symbol, by a code word, resulting in a longer and consequently effective bit duration, or a higher transmitted data rate. Because of Weightless lower costs, it fits where the use case requires devices massively deployed, that demand lower costs and do not necessarily need a big range, for example, smart home devices [27].

802.11ah

The 802.11ah [28], also referred to as HaLow, was introduced in 2017 by the IEEE, as competing standard for the WAN world. It has high data rate up to 347 Mbps, with a maximum packet size of 65535 bytes using aggregation (7991 bytes without), and a range of 1Km. HaLow has a transmitting power between 1 mW an 1 W and 26 MHz bandwidth. The use cases for the 802.11 ah technology can be the agricultural automation, smart metering, industrial automation and animal monitoring, since these type of applications do not require large amounts of devices or even range, while at the same time presenting the need for high packet delivery rate and low delay.

Another comparable technology developed by IEEE is 802.11af, that has a longer range of over 3 Km and, similarly to Weightless-W, takes advantage of the unused TV channels from 54 to 698 MHz.

RPMA

Random Phase Multiple Access (RPMA) is a proprietary standard, developed and patented by Ingenu [29], that uses the globally available ISM band of 2.4 GHz, which means there is no need to create different radio modules for different regions. It has a maximum range of close to 13 Km, while also having high data rates, with a 1 MHz channel, 625 Kbps uplink and 156 Kbps downlink and a transmit power of up to 20 dBm. This reflects in the need for fewer base stations to cover the same amount of area, while at the same time allowing a higher data throughput. This technology can perform roaming, and firmware updates over the air, being the last feature necessary to keep the devices future-proof. It supports packet sizes from 6 bytes to 10 KB, allowing for a wide range of information transmission [30]. In short, RPMA is a versatile technology, that supports high data rates. However, a higher frequency also means that penetration through most materials is less effective. This last factor contributes to less range in dense urban areas or large indoors facilities [31].

MIOTY

MIOTY [32], LPWAN's solutions, uses a very low rate to achieve extensive range, resulting in a very long on air time. This a problem in the licence free spectrum, because several technologies coexist in the same spectrum, the longer the on air time of a message, the more likely is to collide with other message sent at the same time, resulting in data loss. To overcome this challenge, MIOTY uses a technology, called telegram splitting, that, unlike other LPWANs MIOTY, does not transmit an entire single message at once, instead it splits the message into sub-packets and sends them at different times and frequencies. Since the on air time is much shorter, is less likely to collide with other messages and, even with a 50% of sub-packets collision, the full message will be successfully reassembled. This telegram splitting technology provides interference resilience, with deep indoor penetration, good scalability with 1.5 million messages a day, with a single base station, ultra low power consumption, with a battery life of more than 10 years, good mobility up to 120 km/h and a long range of 15 Km.

2.2.6 LPWAN comparison

In the Table 2.6 is given a comparison between the main LPWAN technologies for IoT. Every use case has different restrictions. For the use case proposed in this work, the main restrictions were in payload size, power efficiency and data rate. In the next sections the Lora and NB-IoT were used, as they were the ones that better met the needs of the use case.

Table 2.6: LPWAN Comparison

	Sigfox	**LoRaWAN**	**NB-IOT**	**LTE-M**
Modulation	BPSK	CSS	UL: SC-FDMA DL: OFDMA	UL:SC-FDMA, 16 QAM DL:OFDMA, 16QAM
Spectrum	Unlicensed ISM bands	Unlicensed ISM bands	Licensed LTE frequency	Licensed LTE frequency
Band	Eu: 868 MHz US: 915 MHz Asia: 433 MHZ	Eu: 868 MHz US: 915 MHz Asia: 433 MHZ	UL:700-2100 MHz DL:882 ,1840 MHz	UL: 1.8-2.7GHz DL: 2.6 GHz
Bandwidth	100 Hz	125-500 kHz	180 kHz	1.4 MHz
Link Budget	151 dB	154 dB	164 dB	155.7 dB
Data Rate	UL:100 bps DL:600 bps	290bps-50kbps	20kbps	200 kbps- 1Mbps
Adaptive Data Rate	No	Yes (SF dependent)	No	No
Max. payload	UL: 12 bytes DL: 8 bytes	250 bytes (SF dependent)	1600 bytes	?
Max. messages/day	UL:140 DL: 4	No	No	No
Range	10 km (urban) 40 Km(rural)	5 Km (urban) 20 Km (rural)	1 Km (urban) 10 Km (rural)	3 Km (urban)
Interference immunity	High	Very High	Low	Medium
Latency	1 s -30 s	Based on class	1.6 s - 10 s	10 ms - 15 ms
Private Network Option	No	Yes	No	No
Over-the-air updates	No	Yes	No	Yes
Encription	No	AES 128b	LTE encryption	LTE encryption
Power efficiency	Very High	Very High	Medium High	Medium
Localization	Yes (RSSI)	Yes (RSSI, TDOA)	No (under Specification)	
Mobility	Limited	Yes	Limited	Yes
Availability Portugal	Yes	Yes	Not yet (available for testing)	No
Module size	Suitable for wearables			
Standardization	No (works in progress with ETSI)	LoRa-Alliance	3GPP	3GPP

2.3 Geolocation

Geolocation is an issue that for a long period of time has been solved mainly with the usage of GPS. However, with the emergence of low cost and low power Internet of Things (IoT) devices, the evolving demands placed on these devices require new solutions to old problems [33].
The author of this thesis, in order to do an adaptive geolocation model, will combine both GPS and GPS-free alternatives.

Geolocation consists of the identification, or estimation, of the geographical location of an object in the real world. This process involves the creation of a set of geographic coordinates, represented in latitude and longitude.
There are multiple techniques which can be utilised in order to estimate the actual position of the device, each one of them with different features and purposes. It is important to select the most appropriate one depending on the information available from the end-node.

2.3.1 Algorithms

The three widely used methods for doing the geolocation are triangulation (Figure 2.14), trilateration (Figure 2.15) and multilateration (Figure 2.16).

Triangulation works by using the angles of incidence, from the received signal sent by the transmitter. With this known information a triangle is defined, with two of them and the end-node, the approximately position is calculated by applying trigonometric formulas.

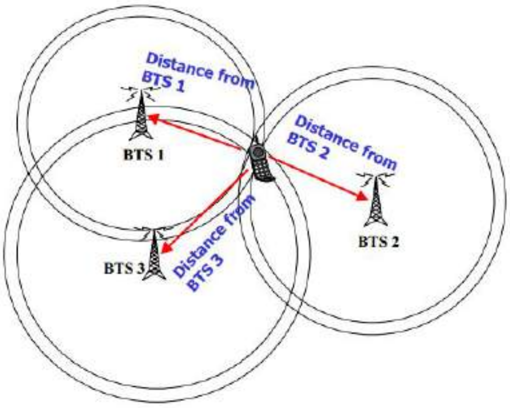


Figure 2.14: Triangulation [34]

For trilateration the distance between the transmitter and the receiver is required. This information can be obtained in different ways such as the Time of Arrival (ToA), the Time of Flight (ToF) or the Received Signal Strength Indicator (RSSI). The downside for this technique is the required synchronization between the transmitter and the receiver. The position is calculated by the intersection of the three circles obtained from the previous distances.

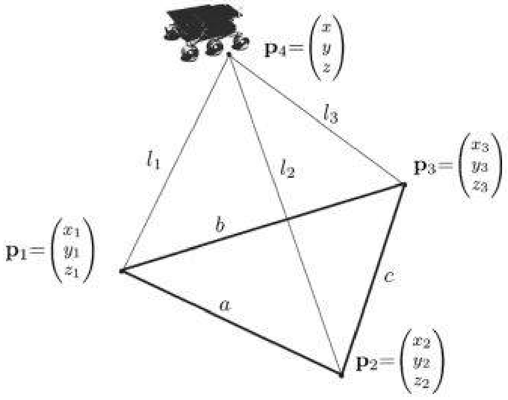


Figure 2.15: Trilateration [35]

Multilateration is similar to trilateration. However, the main difference is the feature used to calculate the estimated location, which, in this case, is the Time Difference of Arrival (TDoA). The transmitters are still synchronized to each other, whereas the receiver, in this particular technique does not need to be. Thus, the location in multilateration is the intersection of a minimum of two hyperbolas. For this method to be able to properly function, three antennas are required [36].

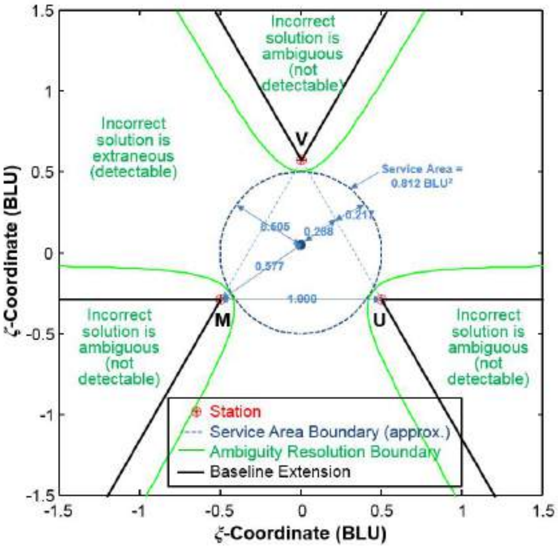


Figure 2.16: Multilateration [37]

2.3.2 GNSS

The Global Navigation Satellite System (GNSS) contains the Global Positioning System (GPS), which is a navigation system developed by the United States of America and its operation is supported by 28 satellites orbiting the planet Earth. The satellites and GPS receivers have an internal clock, which marks the time with an accuracy of nanoseconds. When the signal is emitted by the satellite, the time stamp of the emitted signal is sent by the satellite. This signal, which travels at the speed of light, is received and than it is calculated how long it took to arrive. As the position of the satellites is known, it is possible, through mathematical calculations, to determine the exact position of the user.

The mathematical calculation used is Trilateration. This is accomplished with distance measurements, in contrast to angular measurements, but requires a minimum of three measurements to determine the coordinates (longitude, latitude). This is the technique commonly used by all of the GNSS, such as Global Positioning System (GPS), the Russian Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS) [38] and the European Global Navigation System (Galileo) [39], which can achieve an average precision of 4.9m. More generally, for $n \geq 3$ different distance measurements, this process has the name of multilateration.

2.3.3 Other Geolocation Methods

The positioning systems are dominated by the GPS. Although, as GPS becomes more available and the size and cost of the hardware have been decreasing, there are still scenarios where it cannot be used. First, GPS signals are sensitive to obstacles, making indoor positioning difficult to implement. Second, the price and the battery consumption can be prohibitive for the use case. An example of one of those scenarios, is the situation where a sensor network is composed of a set of small battery-powered devices, where low cost and low power are the main requirements [40]. When GPS is not possible to use, or it is not the optimal solution, a GPS-free system is required. There is always the ability to combine both in a hybrid system, to take advantage of the best features of each one. A GPS-free system consists in the use of the aforementioned algorithms, in order to get a position estimation of a device, without the use of any GPS satellites, to achieve this any kind of radio technologies can be used. The requirements for these systems to work is the ability to use techniques including RSSI, ToF, TDoA, Time On Air (TOA), Angle of Arrival (AOA) [41].

2.3.4 LPWAN Geolocation Comparison

In the subsection 2.2.6, is possible to observe the Table 2.6, where the main LPWANs in the study for this thesis are presented. One of the most interesting fields in the mentioned table for the proposed work is the localization. In the Localization field, the first LPWAN capable of doing so is SIGFOX. In SIGFOX due to the limited amount of message per day, the only viable approach is to do localization based on RSSI. For this the work done by [42], proves it possible in rural and urban scenarios.

In comparison to SIGFOX, LoRa works with a much higher bandwidth which enables localization through Time Difference Of Arrival (TDoA). However, this method, whose architecture is represented in Figure 2.17, demands a very accurate synchronization between the receiving base stations. The distance between the sender and the gateways are estimated based on the time of flight of the message and a location estimation is performed using the triangulation algorithm. Semtech, the company behind LoRa, implemented a proprietary geolocation feature in LoRaWAN, which uses TDoA.

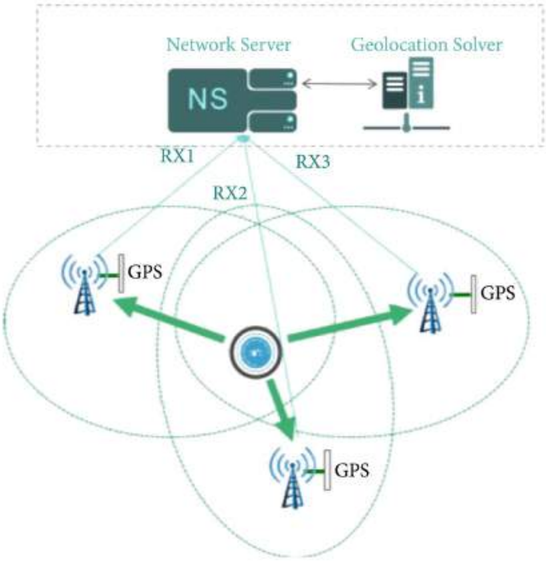


Figure 2.17: Geolocation Architecture [43]

The LoRa Alliance claims that this feature achieves an estimation error of 20 to 200 m, as represented in Figure 2.18. In [36] the TDoA method was evaluated and the conclusion is that is possible to get a location accuracy of approximately 100 m for the stationary sender.

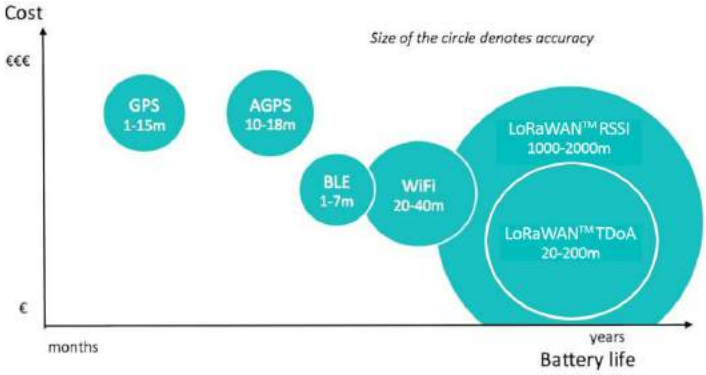


Figure 2.18: Geolocation Comparison [44]

In order to obtain a positioning for an NB-IoT device, one of the methods that could be used is Observed Time Difference of Arrival (OTDoA) localization, represented in Figure 2.19. For this method, the base stations need to be synchronized and transmitting a Positioning Reference Signal (PRS), which is then received by the device. In the event of PRS being unavailable, the Cell Specific Reference Signal (CRS) can be used for the OTDoA. The device needs to forwards the ToA of each transmitting base station to a geolocation server, where the difference between these TOAs and the PRS, is used to perform the calculation of the estimated location [45].

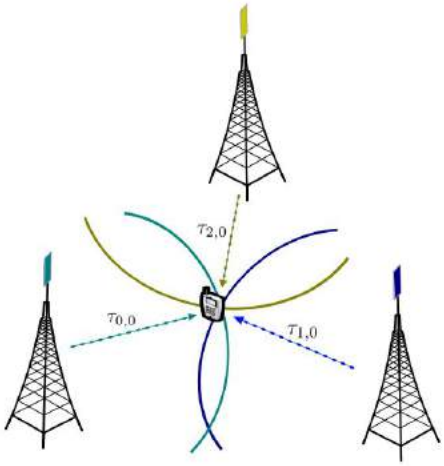


Figure 2.19: NB-IoT OTDOA [45]

2.4 Related Works

Most studies in the subject of geolocation have focused on the use of a single technology, or technique. Then there are the ones that uses multiple technologies and techniques, but it seems that do not explore so much the dynamic change of the working technology in functioning mode. An analysis of related work shows, in the first four examples, they focus only in different individual technologies and techniques. The last two, focus either in multiple technologies, or multiple techniques but, as already emphasized, without taking into consideration the use of multiple techniques and technologies simultaneously.

In [36] an IoT tracking system is presented, using LoRA, where the geolocation is calculated through a multilateration algorithm, on the gateways timestamp, with an accuracy of around 100m in statically test scenario.

In [46] the performance and accuracy of the Google API, for geolocation using WI-FI Aps evaluated the results in a urban environment, achieving a maximum accuracy of 20 meters, minimal 187 meters and median 39 meters.

In [47], an indoor localization monitoring system is presented and a wearable device was developed using FleckTM-3 wireless sensor platform, with a position error of 1 to 3.5 meters. The main disadvantage of this work is that it only works indoors and the technology is similar to ZigBee, meaning that it only works for short range applications.

In [48], was developed an indoor position system based on Raspberry Pi, used as Bluetooth Low Energy (BLE) scanner, and a MPU6050, used has BLE beacon. Measuring the RSSI, the results for indoor activity were 99% accurate in knowing in which division the patience was.

In [42] a dataset of messages was created from LoRa and SIGFOX containing the GPS coordinates and respective RSSI. The results of the median error in a urban scenario were 514.83 meters for SIGFOX and 273.03 meters for LoRa.

In [33], a Hybrid (Time of Flight and RSSI) approach for Geolocation system using using LoRa showed results similar to the work mentioned in [42], with a median error of 272 meters.

The work presented by the author improves the aforementioned solutions with an adaptive geolocation solution that combines the different methods, resulting in an overall location that provides the best results in terms of precision and performance wise, meeting the dynamic changes needed during the utilization scenarios.

CHAPTER 3

ADAPTIVE GEOLOCATION

In today's solutions for locating people with dementia, there is the lack of a system capable of locating a person and, at the same time, being able to dynamically choose the best location, thus saving battery power. In light of this situation, this chapter aims to describe this dissertation's intended model, proposed as a module that is part of the Carelink platform.

3.1 Overview

The above-mentioned platform is represented in 3.1, and it was designed to be safe, extendable and perform well in the future. To achieve this, the micro-services approach was adopted, where logically distinct elements of application code are developed and deployed separately. These services communicate with each other using a combination of REST and messages. This enables services to be continually improved and new services to be implemented whenever needed.

One of the services present in the platform is the tracking service. This micro-service acts as a consumer of the output, from the proposed model. The model acts as a bridge between the physical world, composed by the hardware devices, the respective sensors and the virtual world represented by the Carelink platform.

Determining the location of people with dementia demands a resilient and fast system, capable of offering real-time data, structured in such a way that it is easy to present and visualize. This is a key aspect so that the next module is capable, without much effort, to present the information in a graphical away, allowing the user to access it quickly. Since the situation where a person with dementia has an episode of wandering is highly stressful for the caregivers, or the person in charge, it is imperative that the system is able to provide an accurate location without being too complex. Because this system is intended to track devices that are connected to persons - not to assets or animals - is one main requirement to have high availability.

Figure 3.1: Carelink Platform Architecture [6]

The model must be prepared to receive, as well as, interpret messages from different devices and transmission types (LoRa, NB-IoT). In order to achieve this must be device agnostic, the way to solve this issue is using a naming convention for the protocol of communication Message Queuing Telemetry Transport (MQTT). Following the same ideology of the platform, it must also be divided into small modules, so that it can be easily scalable and allow updates without large changes to the core implementation - becoming, in this way more future proof. This approach also benefits the model with a more easy and manageable way of implementing security, monitoring and redundancy.

As the information may arrive at different rates, or even, in the worst-case, fail to arrive at all, the model functionalities work in modules that are independent and that can work isolated, as well as in parallel. To solve this problem inside of one particular module was used smart gate queuing and load balancing techniques.

To conclude, it was intended that the model, alongside with is integrated system, would be capable of providing a location, taking into account the remaining battery of the device and, in the end, creating a profile-based decision system in charge of dynamically choosing the best location. By these means, the adaptive geolocation capability is provided.

3.2 Adaptive Geolocation Solver

The Adaptive Geolocation Solver is the model part of the system, that compose the solution. The main objective for this model is receiving the messages from the devices and return the best possible location.

3.2.1 Model Schematic

Figure 3.2 introduces the model schematic for the adaptive geolocation solver. This model is composed by different modules, each of one with a different function. Following this image will be the explanation of each module, as well as the main functions present inside of each.

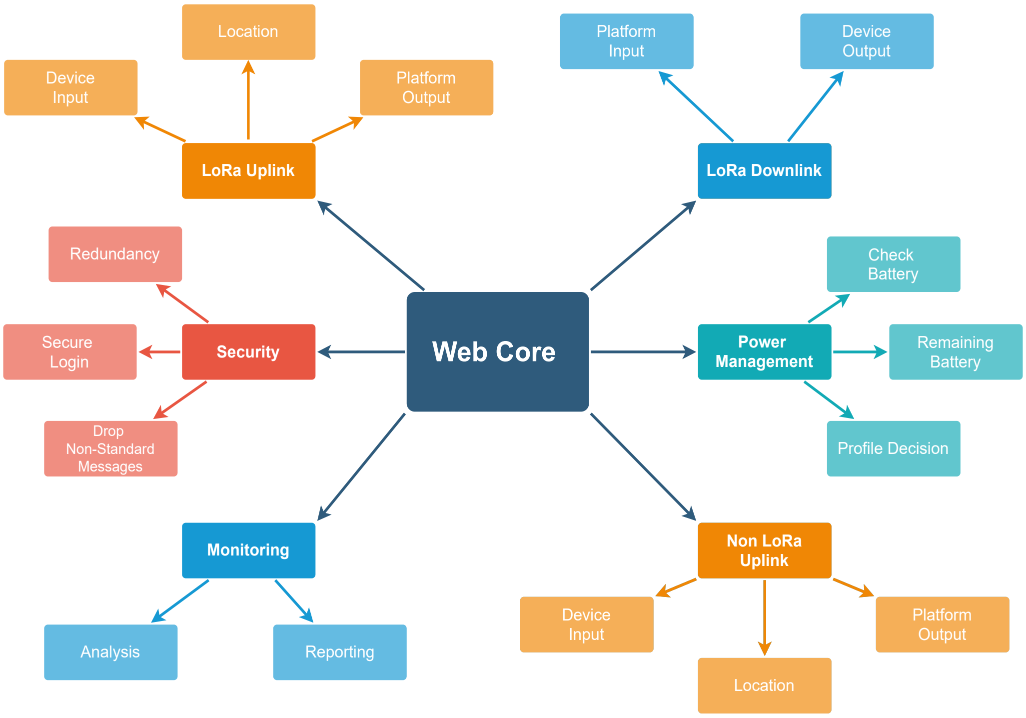


Figure 3.2: Mind Map for Model Schematic

- **Web Core**
  This is the core module where all the others are built on top of. The main characteristics of this module will be discussed in more detail in the next Chapter 4, but essentially it is a web server capable of running JavaScript functions.

- **LoRa Uplink**
  This first module is responsible for the incoming messages. when the previous transmission method was LoRa. This module is responsible for the next functions:

  - Device Input:
    The messages received from LoRa transmission and are stored in a smart gate queue;

  - Location:
    After the message is ready to be processed, it is filtered to look up for the location information. The best location is joined to the original message.

  - Platform Output:
    In the end, the right device is selected and the combined message is sent to the platform.

- **LoRa Downlink**
  The Carelink platform can communicate directly with devices, through MQTT, but this is not possible directly using LoRa (as explained in the next Chapter 4), so this module acts as a middle layer for the communication.

  - Platform Input:
    The different MQTT topics are subscribed, and the message is received. Then is converted to JavaScript Object Notation (JSON) and encrypted to Base64.

  - Device Output:
    After the message passes to Base64 and the correct topic is selected, this function selects the right device and sends the message.

- **Power Management**
  This module is in charge of all the power management capabilities of the model and it is composed by the following functions:

  - Check Battery:
    This function is responsible for analysing the received message, verifying the battery level and the communications that are currently in use.

  - Remaining Battery:
    Where the battery level is converted into the remaining battery duration.

  - Profile Decision:
    Using the previous information, it decides which profile to use.

- **Non LoRa Uplink**
  This module shares the same functions as the "LoRa Uplink", however is designed to be more communication agnostic, and therefore serve more devices.

- **Security**
  The "Security" module is responsible for the correct operation of the model. This module consists of the following functions:

  - Redundancy:
    This function is achieved, through the ”LoRa Uplink" and ”Non LoRa Uplink", by having more than one path to transmit the message to the platform. Inside both, there is also a redundancy location mechanism. The last redundancy feature is the fact of the "Web Core" is running in a docker container.

  - Secure Login:
    The access to the "Web Core" is done using a login with username and password, but the password is not stored, only the hash.

  - Drop Non-Standard Messages:
    All the incoming messages that are Non-standard are filtered and drop, securing the model and making it more efficient.

- **Monitoring**
  Using the following functions, and with the ability to connect with the other modules, is capable of ensuring the normal operation for the model.

  - Analysis:
    This function is mainly done in the "LoRa Uplink" and "Non LoRa Uplink" modules, where the number of messages flowing, is registered in different points.

  - Reporting:
    Making use of the previous knowledge of "Analysis", this function generates a daily e-mail report and, at an abnormal situation where the non-processed messages achieves a defined threshold, this functions sends an SMS message.

- **Development**
  This last module exists but is not represented in the schematic, since it is not used in a production environment. The development module aims to provide a sandbox, where the following features can be tested, alongside with the other modules that are in a production environment, but without being in production.

  - Debugging:
    This function is used in case something goes wrong. It is possible to replicate to find the error.

  - Updates Testing:
    When the error is found, it is possible to develop and test an update. After some iterations, the update is finished and ready for production.

  - Map visualization:
    In order to have a place for visualization of the data, in a real world map, to get an idea of what is going to be shown next, in the platform, and for easy interpretation of the data, this function was created.

3.2.2 Stages & Operation

The presented work follows a particular model on its development and implementation organisation. Such model intends to define what technologies to use and how the geolocation data from different sources should be processed, so that it can later be used for knowing the actual position of the patient wearing the device. This same model also characterises and formalizes the different types of solutions, called stages.

The diagram 3.3, represents the three different functional stages, called: “hierarchical”, “advanced” and “smart” in a kind of Venn diagram. Then an explanation of each stage is presented that follows the proposed profile based decision system implementation.

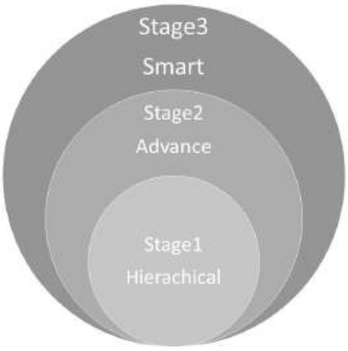


Figure 3.3: Model Stages

- **Stage 1** - Hierarchical

The first stage is called hierarchical, in which the best location is decided using the approximate accuracy values of each method. Taking into account the documentation, and the tests performed, ordered the next technologies in relation to accuracy in the following order from the best to the weakest: $GNSS > WiFi > LoRa$.

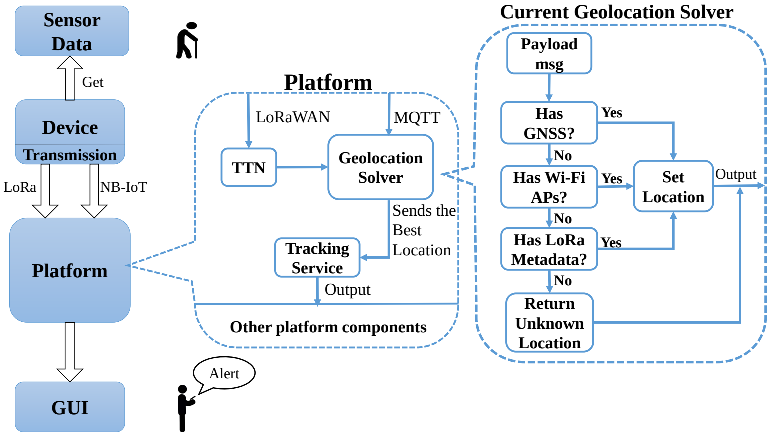


Figure 3.4: System and Model Architecture - Hierarchical version A

In this initial phase, it only includes the "hierarchical" functional stage. Its architecture is represented in Figure 3.4. In the right dotted square from this figure, is the workflow behind the "Geolocation Solver" block, that it's where the model is working. Its operation is described up next. First, as described at 3.2.1 on page 31, an input message from the device is received containing a JSON, where it exists an object with the status of the device. This status message has the following fields:

```
{
    "timestamp": "YYYY-MM-DDThh:mm:ssZ",
    "location": {
        "lat": float,
        "lon": float,
        "alt": float,
        "hdop": float,
        "vdop": float,
        "pdop": float
    },
    "batteryLevel": integer,
    "accompanied": boolean,
    "sensor": {
        "accelerometer": {
            "x": float,
            "y": float,
            "z": float
        }
    },
    "wifiAPs": {
        "mac_1": string,
        "rssi_1": integer,
        "mac_2": string,
        "rssi_2": integer,
        "mac_3": string,
        "rssi_3": integer
    }
}
```

Figure 3.5: JSON Status Payload

In the first block "Has GNSS?", the location field of 3.5 is analyzed. There are two options, for the coordinates: either valid or not invalid. For the coordinates provided to be valid, the latitude and longitude fields need to be simultaneously different from 0. If the information is valid, the location is set and then forwarded to the Tracking service in the Carelink platform. The unused information for the other methods is dropped.

On the other hand, the received messages where these fields are zero can be considered invalid since they come with the default value from the device. In the case where the coordinates provided arrive with a type different from float, it will also be considered invalid since a transmission error occurred.

In case the location field is empty or the coordinates are considered not valid, the following block is the “Has Wi-Fi APs?”. In this, the "wifiAPs" field from the above-mentioned JSON object is analysed. If it is different from null, this payload is used to perform the assisted location, based on the Wi-Fi data. This data will then be sent to three different APIs, using a load balancer, which is based in the sequential Round Robin. This is done to obtain the best information for the location, and, at the same time, use fewer API calls, to prevent the APIs from refusing the requests. The three used APIs in this work were: Google Geolocation API; Here Position API and OpenCelliD Cellular Geolocation API. This load balancer uses the following equation:

$$h(x) = \begin{cases} HO & , x \in 3n+1 \\ HG & , x \in 3n+2 \\ OG & , x \in 3n+3 \end{cases} \quad n \in \mathbb{N}_0 \tag{3.1}$$

In the previous equation 3.1:

- H - Here
- O - OpenCelliD
- G - Google

After receiving the response, the information is compared between the two used APIs, in order to guarantee redundancy, as well as the best accuracy possible. Then it is combined in the JSON Status and the unused fields are discarded. In the end, this JSON is also forwarded to the platform.

If the communication method in use was LoRa, then the LoRa metadata will also be used to apply different geolocation algorithms, such as multilateration, based on received signal strength or the time difference of arrival. The result is the approximated device location. For this last method work with reliability, a minimum of three gateways in range is needed. In the scenario where all of the three previous blocks failed to return a valid location, it is returned location unknown.

```
5/5/2020, 6:31:13 PM   node: a22a9763.d535a8
msg : Object
 ▾object
   app_id: "lora-test-otaa"
   dev_id: "node1"
   hardware_serial: "00C2383741993362"
   port: 1
   counter: 0
 ▸payload_raw: buffer[3]
 ▾metadata: object
     time: "2020-05-
     05T17:31:14.220953173Z"
     latitude: 38.660084
     longitude: -9.203909
 ▸payload: buffer[3]
   _msgid: "7db65134.7f0b2"
```

Figure 3.6: LoRa Location Metadata

- **Stage 2** - Advance

The second stage formalises the location decision function based on two characteristics at the same time: the battery level of the device and the location accuracy for the chosen technology, thus being more "advanced". This stage also adds another location method, the location through beacons BLE, that can be used indoors. These beacons have a short-range and are used in the situation where GNSS is not working and there is no WiFi available, for example, inside the house of a person with dementia, in a rural place. Alongside the addition of this location method, there is also an addition of WiFi, for transmitting the data, in the device side. This is not model related, since the model is agnostic to the transmission method, but it was a step taken in this stage.

In this stage, for assuring the high availability needed for this work, the geolocation function should be capable of knowing the remaining battery. In case the person is in a dangerous situation, or in an area that was previously assigned as unsafe, this stage does the balance between the optimal location and the more power efficient location. As represented in the following Figure 3.7, where the different numbers $(1, 2, 3, 4)$ represent the different levels of Geolocation accuracy and battery saving. Using path 1 exists more location sources, so it has higher accuracy, but also require more battery.

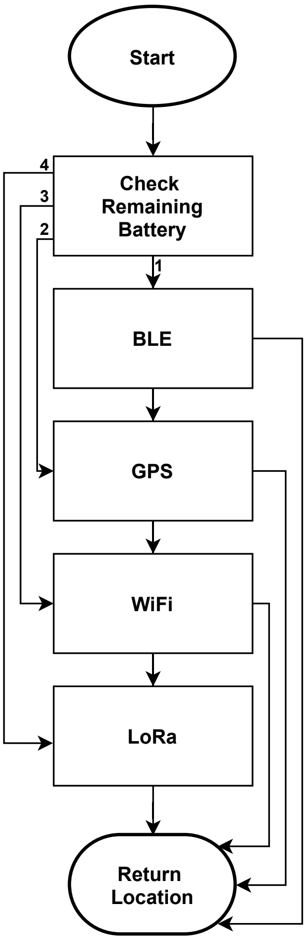


Figure 3.7: Stage 2 State Machine

- **Stage 3** - Smart

In the last stage, a more advanced location function is formalised. Its decision capability for choosing the technologies to use, is based in the surrounding environment. For example, if a patient is not showing any activity in a long period of time, it is night time and he/she is in a safe place such as their home, it can be deduced that the person is sleeping and therefore the sampling time for the location can be reduced, thus saving battery. Another example is, by combining additional sensor data, such as an accelerometer that can detect if the PwD has fallen, the priority can be given to the method with the best accuracy, because of this dangerous situation.

This stage introduces the addition of SIGFOX and Pymesh for transmitting the data in the device side. SIGFOX has a daily limit for the Uplink messages, and a maximum size that is too small to transmit a full message. Therefore, it is only used in the worst-case scenario. The Pymesh uses LoRa, but only the LoRa Modulation, to create a mesh network between the devices. This is the method for the last resource, when all others have failed.

To conclude, this last stage combines a set of different factors to categorize and do a profile-based decision, as it is possible to observe in Table 3.1.

		HOME		Outside Safe Zone		Outside Unsafe Zone	
PwD STATUS		Accompanied	Alone	Accompanied	Alone	Accompanied	Alone
Normal	Day Time	BLE WiFi Sensors 20 min	BLE WiFi Sensors 15 min	LoRa NB-IoT GNSS 30 min Sensors 30 min	LoRa NB-IoT GNSS 15 min Sensors 30 min	LoRa NB-IoT GNSS 15 min Sensors 15 min	LoRa NB-IoT GNSS 10 min Sensors 10 min
	Night Time	BLE WiFi Sensors 1 hour	BLE WiFi Sensors 30 min	LoRa NB-IoT GNSS 1 hour Sensors 30 min	LoRa NB-IoT GNSS 30 min Sensors 30 min	LoRa NB-IoT GNSS 30 min Sensors 30 min	LoRa NB-IoT GNSS 5 min Sensors 5 min
Warning	Day Time	BLE WiFi Sensors 15 min	BLE WiFi NB-IoT Sensors 5 min	LoRa NB-IoT GNSS 15 min Sensors 15 min	LoRa NB-IoT GNSS 10 min Sensors 10 min	LoRa WiFi NB-IoT GNSS 10 min Sensors 5 min	LoRa WiFi NB-IoT GNSS 5 min Sensors 5 min
	Night Time	BLE WiFi Sensors 30 min	BLE WiFi NB-IoT Sensors 10 min	LoRa NB-IoT GNSS 15 min Sensors 15 min	LoRa NB-IoT GNSS 10 min Sensors 10 min	LoRa WiFi NB-IoT GNSS 5 min Sensors 5 min	LoRa WiFi NB-IoT GNSS 2 min Sensors 2 min
Danger	Day Time	BLE WiFi Sensors 10 min	BLE LoRa WiFi NB-IoT Sensors 1 min	LoRa WiFi NB-IoT GNSS 10 min Sensors 5 min	LoRa WiFi NB-IoT GNSS 5 min Sensors 1 min	BLE LoRa Mesh WiFi SigFox NB-IoT GNSS 5 min Sensors 1 min	BLE LoRa Mesh WiFi SigFox NB-IoT GNSS 1 min Sensors 1 min
	Night Time	BLE LoRa WiFi NB-IoT Sensors 5 min	BLE LoRa WiFi NB-IoT Sensors 1 min	LoRa WiFi NB-IoT GNSS 10 min Sensors 5 min	LoRa WiFi NB-IoT GNSS 5 min Sensors 1 min	BLE LoRa Mesh WiFi SigFox NB-IoT GNSS 5 min Sensors 1 min	BLE LoRa Mesh WiFi SigFox NB-IoT GNSS 1 min Sensors 30 sec

Table 3.1: Operation Mode Profiles

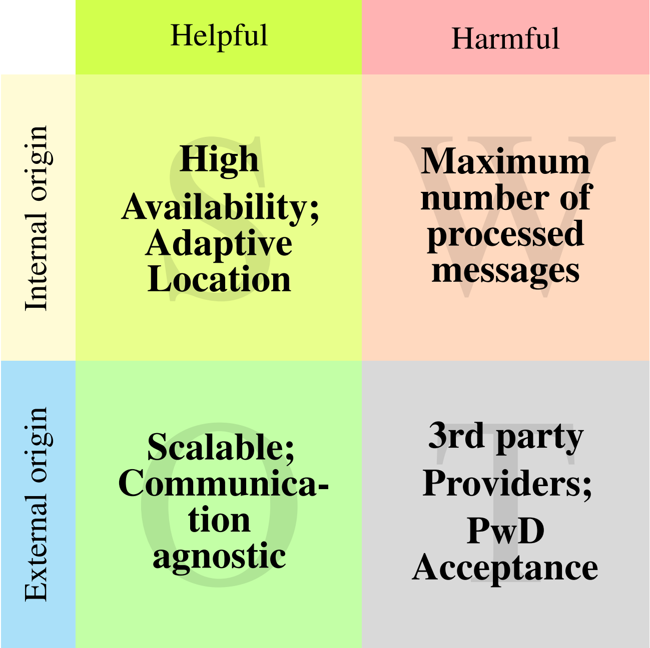
3.3 Summary

In summary, the proposed model will be contributing, as one of the building blocks, to the system illustrated in Figure 3.4

The first step in the proposed system is to collect sensor data from the PwD wearing the device. Afterwards, a specific transmission method is selected, of which the options may vary from LoRa to NB-IoT. The second step, already inside of the platform, is to combine the previous information in the proposed model, called Adaptive Geolocation Solver, in order to get the best location possible. This model had different stages of development, as described in Stages & Operation, and is constituted by 6 modules as it is possible to observe in Adaptive Geolocation Solver.

The Carelink platform, alongside with the model, is responsible for managing the devices and ensuring the high availability needed, in order to always know the location of the patient, especially when this one is having an episode of wandering and its lost. The final geolocation information is then passed on to the corresponding micro-service, in this case the tracking service, and then, the final step, is sending the response from the tracking service to the GUI. Once in the GUI the user responsible for the patient can be alerted, when a geofencing alert is raised, with the information that the person has crossed to an unsafe area.

To conclude, the next SWOT analysis of the model summarizes the different aspects of this work. From the data in the top right (orange) corner is possible to observe that one drawback of this work is the maximum number of processed messages. In the bottom right (grey) corner is presented to the reader the threats of this work, being the first the dependency from 3rd party providers, for the locations done through API. The last threat is the PwD Acceptance, which consists of convincing the PwD and the respective caregiver that the proposed solution could benefit both.



	Helpful	Harmful
Internal origin	**High Availability; Adaptive Location**	**Maximum number of processed messages**
External origin	**Scalable; Communication agnostic**	**3rd party Providers; PwD Acceptance**

Figure 3.8: SWOT Matrix

CHAPTER 4

IMPLEMENTATION

Following the concepts introduced in the previous section, this chapter covers the implementation of the system, the methodology and the chosen technologies for this process. It starts by explaining the device part and then it covers the model part for each individual component.

4.1 Architecture

For this implementation, there are mainly 3 entities that are connected together composing the proposed architecture (Figure 4.1), being those the following: the Wearable Device; the TTN and the Adaptive Geolocation Solver (model) that is inside of the Carelink Platform. Each one of the entities has its own implementation and programming language and also are independent of each other. This is needed to ensure the scalability of the system. If there is the need to change the inner implementation of any of the entities, it must not affect the overall operation of the system.

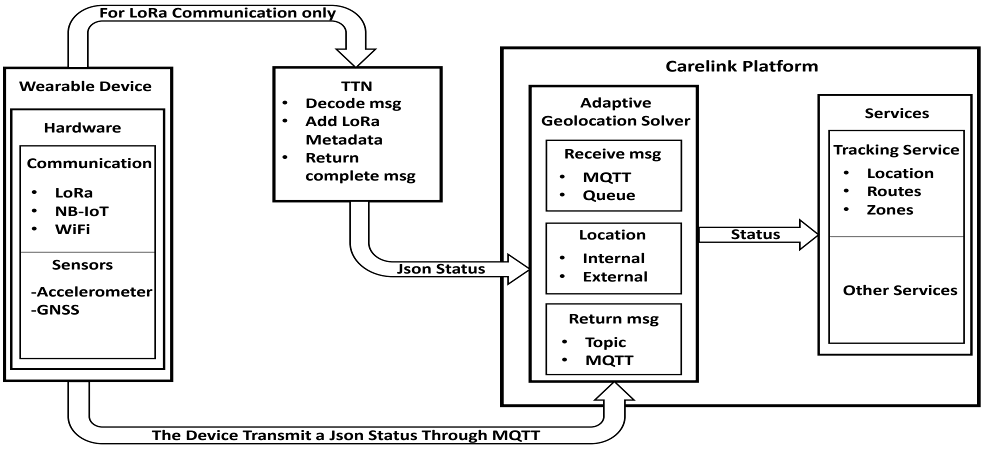
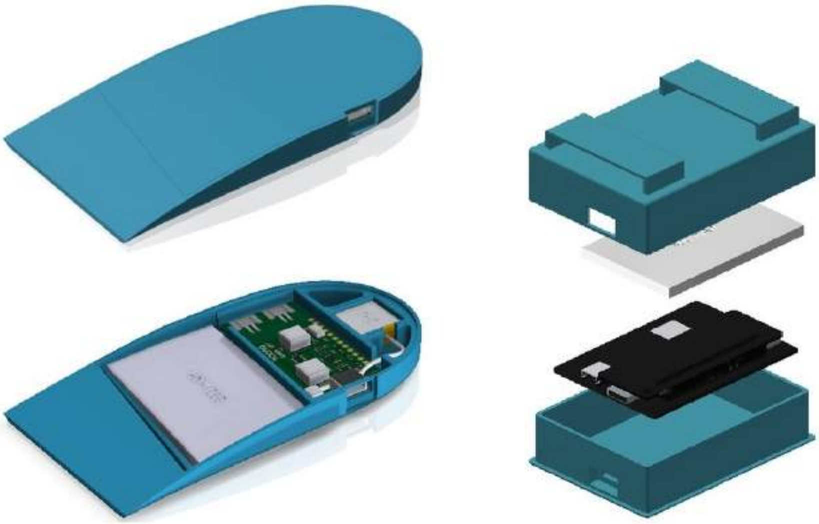


Figure 4.1: Architecture

A major part of the system is the communication between the different entities. These entities will be connected through different protocols and communication standards. The following section explains the protocols used, as well as the more technical part of this work

4.2 Wearable Device

There are two types of wearable devices for the Carelink project: the one represented in 4.2a, as a shoe insole and the other in the 4.2b, as a belt box. These two formats allow for different types of utilization, according to the use scenario and user preferences. The belt box is designed to be attached to the belt of the user, in a comfortable position. The shoe insole is intended to replace the existing detachable insole of the shoes and consists of 2 rigid heels and 2 comfort insole fillings. The rigid plastic heel houses the hardware on the right foot insole and a dummy counterweight on the left. Both types of wearable plastic enclosures have a micro USB connector slot for device's charging purposes.



(a) Insole

(b) Belt Box

Figure 4.2: Wearable Devices*

* These figures are property of Joana Andrade

4.2.1 Hardware

In terms of hardware, both of the above wearables offer GPS location tracking and NB-IoT network communication, and they are powered by an 800 mAh LiPo battery.

The hardware inside the Insole is a SODAQ SARA R412m, presented in the NB-IoT device comparison, where the main advantage is the small form factor, but bearing the issue that it only supports NB-IoT communication.

On the other hand, the belt box was implemented using the FiPy [49, 50] from Pycom, mounted on PyTrack [51] development board (both of them represented in Figure 4.3). The FiPy and PyTrack were chosen for a number of practical reasons:

- The FiPy is able to run programs written in the Python programming language, allowing for rapid prototyping and development;
- The FiPy is capable of having Five networks in one small board (55mm x 20mm x 3.5mm), which is ideal for an Adaptive Geolocation model;
- The PyTrack development board contains both an accelerometer and a GNSS module, as well as provides an easy way of charging and the programming the device.

(a) FiPy [50] (b) Pytrack [51] (c) Combined [51]

Figure 4.3: Fipy & Pytrack

The sensor data from the accelerometer was used for predicting falls, while the GNSS module provided the location. The FiPy board includes a Semtech LoRa transceiver SX1276 radio, to which an SMA Tilt Swivel 1/2 Wave Whip Dipole antenna was externally attached for testing and, for the final product, a Molex ISM 105262 omnidirectional with 0.4 dBi Peak Gain at 868MHZ. For the WiFi and BLE was used the internal antenna and for NB-IoT was used a PCB trace antenna. The chipset of the development board is an Espressif ESP32 containing a dual-core Xtensa 32–bit LX6 capable of up to 600 DMIPS, and an extra ULP–coprocessor that can monitor GPIOs, the ADC channels and can control most of the internal peripherals during deep–sleep mode while only consuming 25uA.

4.2.2 Operation

The Device operation loop is represented in diagram 4.4 below. First, the device is powered up, then proceeds to initial verifications and setup. The next block, that can be seen in more detail in Figure 4.5, is the Connect. Following this operation, there are two options: either the device is not connected, and enters in backup mode; or is connected and starts the operation of getting the energy profile, pulling sensor data and getting the location. After this, it enters in sleep mode for saving energy and then the cycle starts again.

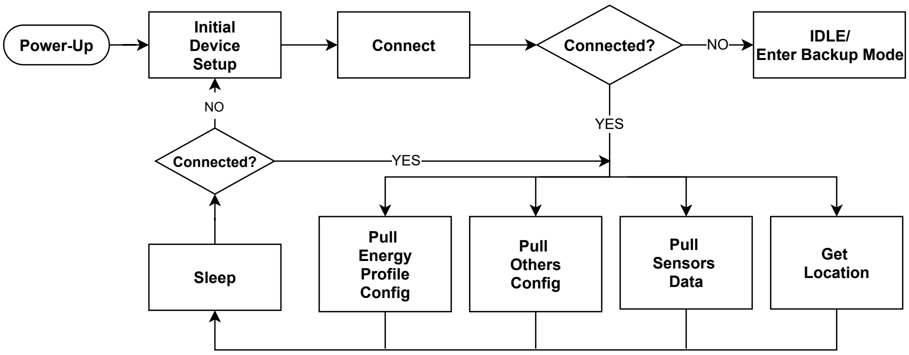


Figure 4.4: Main Operation

Inside the Connect block is a state machine, as it is possible to observer bellow in Figure 4.5. This Block is responsible for doing the Adaptive selection of the transmission method, assuring the high availability needed for this work. The selection option for the transmission may vary from the two LPWAN: NB-IoT or LoRa, and WiFi. This development board also supports BLE, that will be just for sniffing and collect location, not as transmission method, and SIGFOX. That is also an LPWAN, but, due to the maximum message size, and the daily message limit, is not suitable for the use case where this work is inserted.

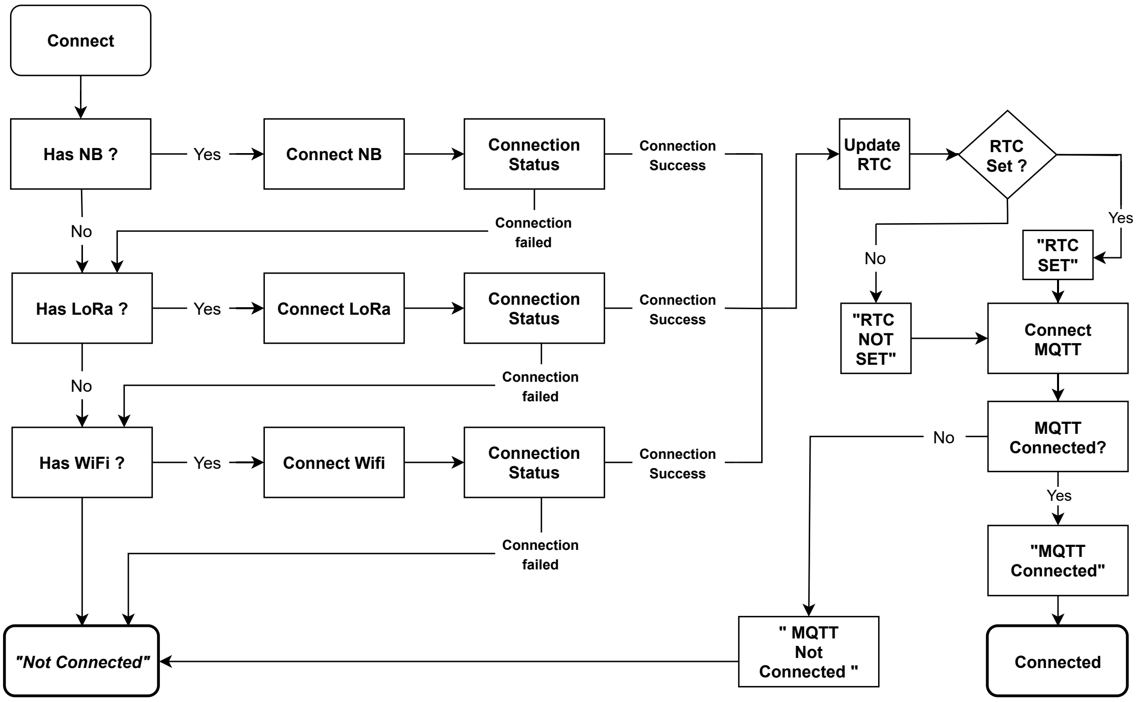


Figure 4.5: Connect

As it is possible to observe in the figure above, the first block is NB-IoT, then LoRa and, in the end WiFi. This is the usual operation but, can be changed according to the energy profile. This order was defined, not because of battery or range, but because of coverage. First, NB-IoT should have higher coverage, then, if this one fails, there is LoRa with the possibility of using community gateways or even creating a private ad-hoc network and, lastly, WiFi is only used in case none of the others work.

Following the previous diagram, the next Figure 4.6, represents how the connect LoRa block works. First, one of the classes that were represented by the previous Figure 2.7, is set, followed by the adding channels and, after that, there is the activation. The selected mode, for security reasons 2.6, was OTAA. When this activation is finished and there is a gateway listening and accepting the join request, the LoRa transmission starts. The other method for activation is ABP.

Connect LoRa is, in fact, a python class that was created and, the following diagram, represents the `__init__` method - after creating the Lora object, exists a thread running for doing the LoRa transmission.

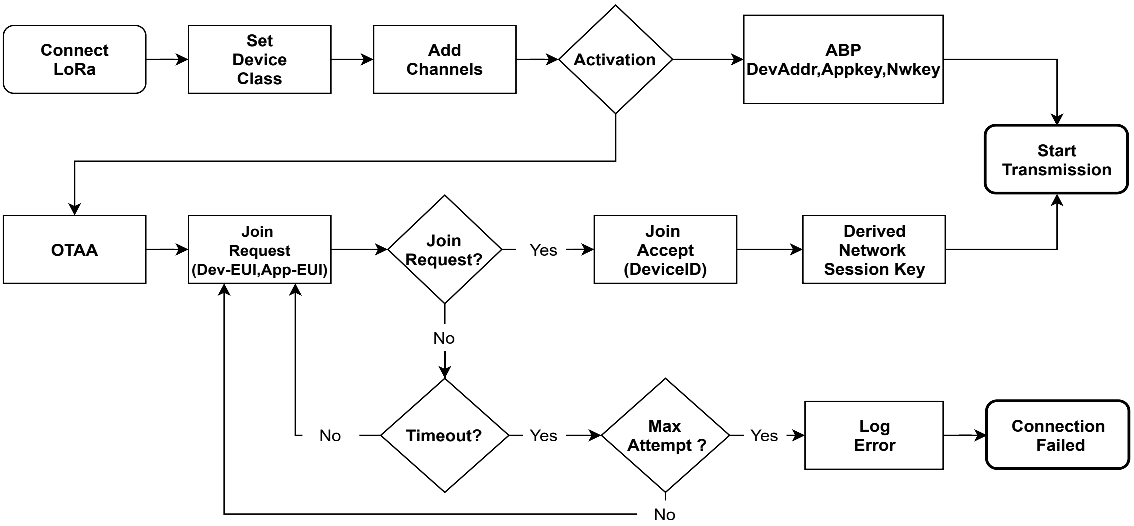


Figure 4.6: Connect LoRa

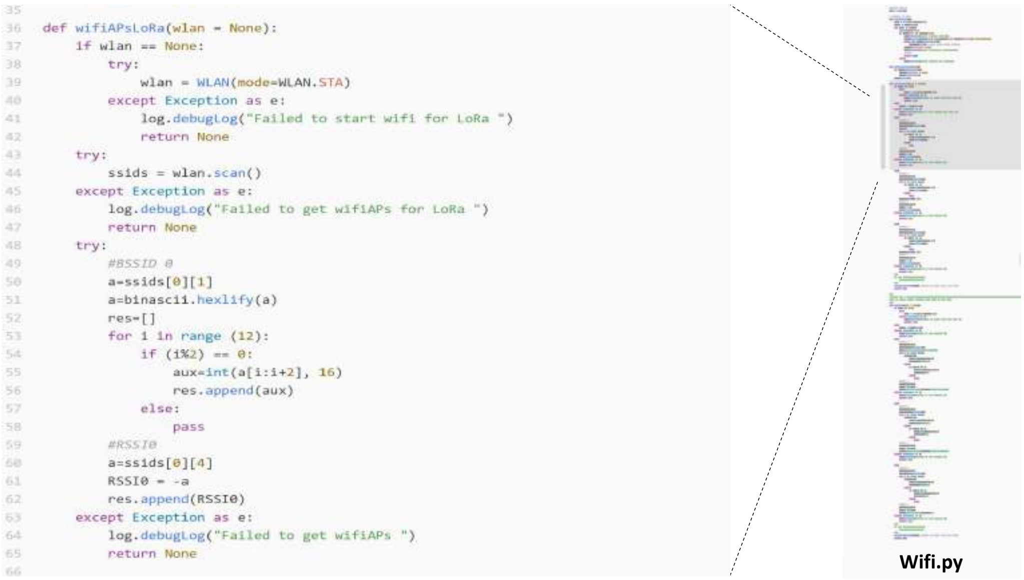
On the device, the work done regarding communication consists in the use of MQTT between the device and the model. When the transmission method used is LoRa there is the need for an extra layer of "translation" as explained in the next section LoRa Network Server.

The next Figures represent the piece of code that is executed inside the "Start Transmission" block. In this function, all values of JSON Status (Figure 3.5) are assigned with the default value of 0, even for a timestamp that is the begin of the UNIX time. This is going to be applicable in later stages for debugging and making decisions in case these values do not change.

```
'''
loraLogic:
    loraLogic inputs: timestamp, location, batteryLevel, accelerometer
    Where the packet is send or received
    Check if there is wifiAPs
    Send to TTN max 242 bytes
    Receive from TTN
'''
def loraLogic(self, timestamp='1970-01-01T00:00:00Z', lat='0', lon='0', alt='0',
 hdop='0', vdop='0', pdop='0', batteryLevel='0', x='0', y='0', z='0'):
    if state.LORA_CONNECTED:
        location = lat +","+ lon +","+ alt +","+ hdop +","+ vdop +","+ pdop
        accelerometer = x +","+ y +","+ z
        wifiAPs = None
        if state.WIFI_ACTIVE:
            #check if there is wifi
            wifiAPs = wifi.wifiAPsLoRa()
        if wifiAPs is not None:
            pkt_status = bytes(wifiAPs) + "," + timestamp + "," + location + ","
            + batteryLevel + "," + accelerometer
        else:
            pkt_status = timestamp + "," + location + "," + batteryLevel + ","
            +accelerometer
        '''
        Transmit the packet
        '''

        self.s.send(pkt_status)
        self.log.debugLog('LoRa Uplink: {}'.format(pkt_status))
        time.sleep(1)
```

Figure 4.7: LoRa Transmission from loraLib.py [52]

The last aspect in the hardware side, for the Adaptive Geolocation model, is the ability of the device to sense the surrounding environments. This can be achieved, for example, through WiFi sniffing. With the gathering of this data (Basic Service Set Identifier (BSSID) and RSSI), as represented in the following Figure 4.8, is possible to estimate the location of the device using assisted location.



```

def wifiAPsLoRa(wlan = None):
    if wlan == None:
        try:
            wlan = WLAN(mode=WLAN.STA)
        except Exception as e:
            log.debugLog("Failed to start wifi for LoRa ")
            return None
    try:
        ssids = wlan.scan()
    except Exception as e:
        log.debugLog("Failed to get wifiAPs for LoRa ")
        return None
    try:
        #BSSID 0
        a=ssids[0][1]
        a=binascii.hexlify(a)
        res=[]
        for i in range (12):
            if (i%2) == 0:
                aux=int(a[i:i+2], 16)
                res.append(aux)
            else:
                pass
        #RSSI0
        a=ssids[0][4]
        RSSI0 = -a
        res.append(RSSI0)
    except Exception as e:
        log.debugLog("Failed to get wifiAPs ")
        return None

```

Figure 4.8: WiFi APs Code from wifi.py [52]

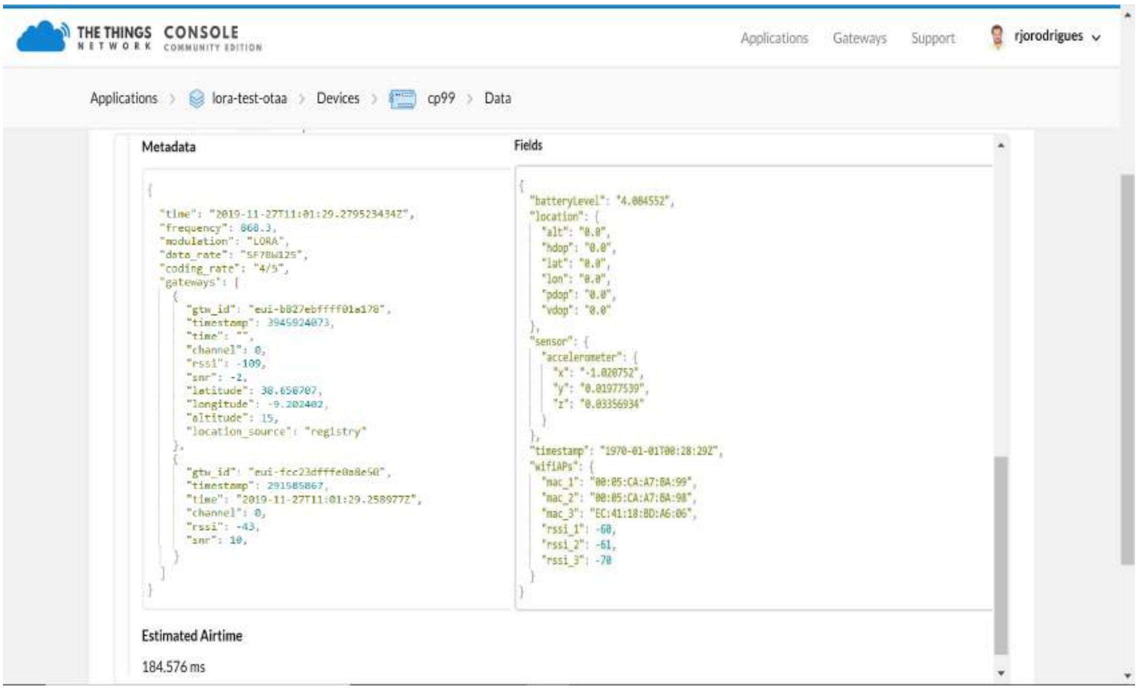
The two previous images 4.8 and 4.7, show pieces of code retrieved from two source files, where the author had contributed. This two files, combined with other 18, are what defined the working principle of the device and can all be found at [52].

4.3 LoRa Network Server

4.3.1 TTN

The Things Network (TTN) [53] acts as the network server and is responsible for providing a bridge, between LoRa communication and the internet. Also, when TTN receives the information, a decoder function, that can be found in appendix C, is used for converting the received message payload to the adequate data format. Moreover, still in the TTN server, the devices are registered in the right application, and the LoRa cloud API integration is used. This API will calculate the current location of the device based on the metadata present in the uplink messages. In the end, TTN will send the result message through MQTT to the Adaptive Geolocation Model.

The next Figure 4.9, shows the TTN console for the application. In this console, is possible to observe the metadata of the message, with information about the gateways, as well as the estimated last location of the device. In addition is possible to observe the *Fields*, where the decode JSON Status is represented.



```
{
  "time": "2019-11-27T11:01:29.279523434Z",
  "frequency": 868.3,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-b827ebffff01a178",
      "timestamp": 3945924073,
      "time": "",
      "channel": 0,
      "rssi": -109,
      "snr": -2,
      "latitude": 38.658707,
      "longitude": -9.202402,
      "altitude": 15,
      "location_source": "registry"
    },
    {
      "gtw_id": "eui-fcc23dfffe0a8e58",
      "timestamp": 291585867,
      "time": "2019-11-27T11:01:29.258977Z",
      "channel": 0,
      "rssi": -43,
      "snr": 10,
    }
  ]
}
```

```
{
  "batteryLevel": "4.084552",
  "location": {
    "alt": "0.0",
    "hdop": "0.0",
    "lat": "0.0",
    "lon": "0.0",
    "pdop": "0.0",
    "vdop": "0.0"
  },
  "sensor": {
    "accelerometer": {
      "x": "-1.020752",
      "y": "0.01977539",
      "z": "0.03356934"
    }
  },
  "timestamp": "1970-01-01T00:28:29Z",
  "wifiAPs": {
    "mac_1": "00:05:CA:A7:8A:99",
    "mac_2": "00:05:CA:A7:8A:98",
    "mac_3": "EC:41:18:8D:A6:06",
    "rssi_1": -60,
    "rssi_2": -61,
    "rssi_3": -70
  }
}
```

184.576 ms

Figure 4.9: TTN Console

For this work, TTN was selected because it was the LoRa network server with better community support, documentation and also fulfilled the requirements for the use case in study. There are other alternatives for TTN, such as loriot [54], thingsboard [55], Mbed OS [56] and Mozilla IoT [57].

4.4 Adaptive Geolocation Services

For the presented work, to achieve the adaptive Geolocation was developed and implemented a model called "Adaptive Geolocation Solver". This model is responsible for receiving the data through MQTT from the different communication methods, doing the calculation for the best geolocation and returning a JSON called *status* through MQTT to the Carelink [6] platform.

4.4.1 Node-red

The platform chosen by the author for the development of the Adaptive Geolocation Solver, was Node-Red [58]. Node-RED is Low-code programming for event-driven applications. It uses development tools for visual programming and was initially developed by IBM with the intention of wiring together devices, APIs and other online services that were part of the IoT. Node-RED uses a web browser-based flow editor that can be used to write JavaScript functions. The run-time is built on top of Node.js. The flows created in Node-RED can be saved using JSON. In 2016, Node-RED was contributed by IBM to JS Foundation as an open source project. The instructions needed to replicate this work are in Appendix B, and the complete guide can be seen in [59]. Even so, the six main modules will be described below.

- **LoRa Uplink**
  As it was introduced in the previous Chapter 3, the "LoRa Uplink" module is intended to be used for LoRa Uplink transmissions. Its operation can be seen in Figure 4.10.

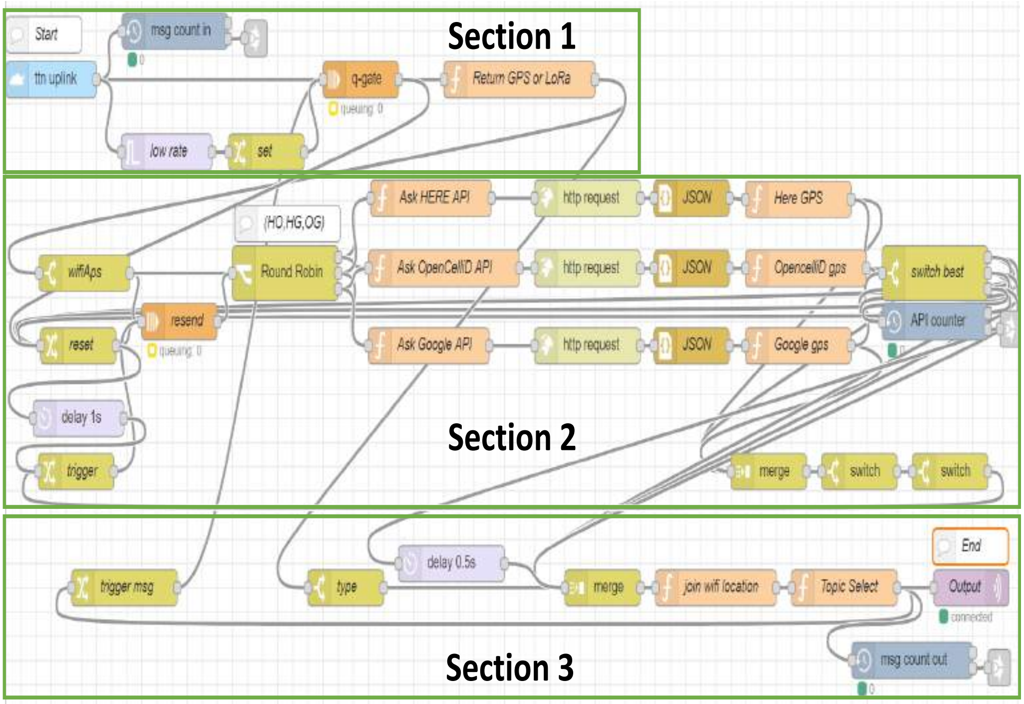


Figure 4.10: LoRa Uplink Flow

As it is possible to observe in the previous Figure 4.10, the LoRa Uplink module is divided into three sections. In order to better understand its implementation, those sections will be described below.

- Section 1
  In section 1, the model receives an input message from the TTN, as explained in Chapter 4.3, and this message is stored in a queue with a gate. This queue gate can be trigger activated, which means that a message is only released when the last message was processed. This received message serves as input for monitor counter and for the LoRa Downlink module, for later use. After the queue comes the first function, where the message is analysed. If the message contains valid GPS data, the next section is section 3; if, on the other hand, there is no GPS data, but Wi-Fi data is available, the next section is section 2.

- Section 2
  At Section 2, the Wi-Fi data is first analysed and then is sent to the load balancer, at the same time that it is stored in a second queue. The load balancer distributes the messages through the 3 APIs, following the equation presented in 3.1. If the two chosen APIs, failed to delivery the location, the gate is activated and the message is resent to another API. If the location is successfully received, the queue is cleared and the location with the best accuracy is sent to Section 3.

- Section 3
  In this last section, the received information from Sections 1 and 2 is analysed. The first function ensures that the previous information is not null, then the original information is merged with the information from Section 2. After that, the correct topic is selected and the final information is sent to the MQTT broker of the Carelink platform. At the end of the function responsible for the selection of the correct topic is the monitor counter and a connection to the trigger of the first queue, in order to release the next message.

- **LoRa Downlink**
  The Carelink platform is capable of communicating directly with devices, through MQTT, but this is not possible directly using LoRa, as explained earlier in Section 4. With that being said, this module acts as a middle layer for the communication. This module is represented in Figure 4.11, on the left side of this figure are the subscribed MQTT topics. In the bottom of the figure is the input message from the LoRa Uplink module, which was used to send a "ping" message to the device, used for knowing that the device was still alive. The other solution was to use Uplink confirmations, but this was less power efficient. The workflow behind this module is the following: first, subscribe to the MQTT topics, then convert the information to base64, select the correct node and send to TTN.

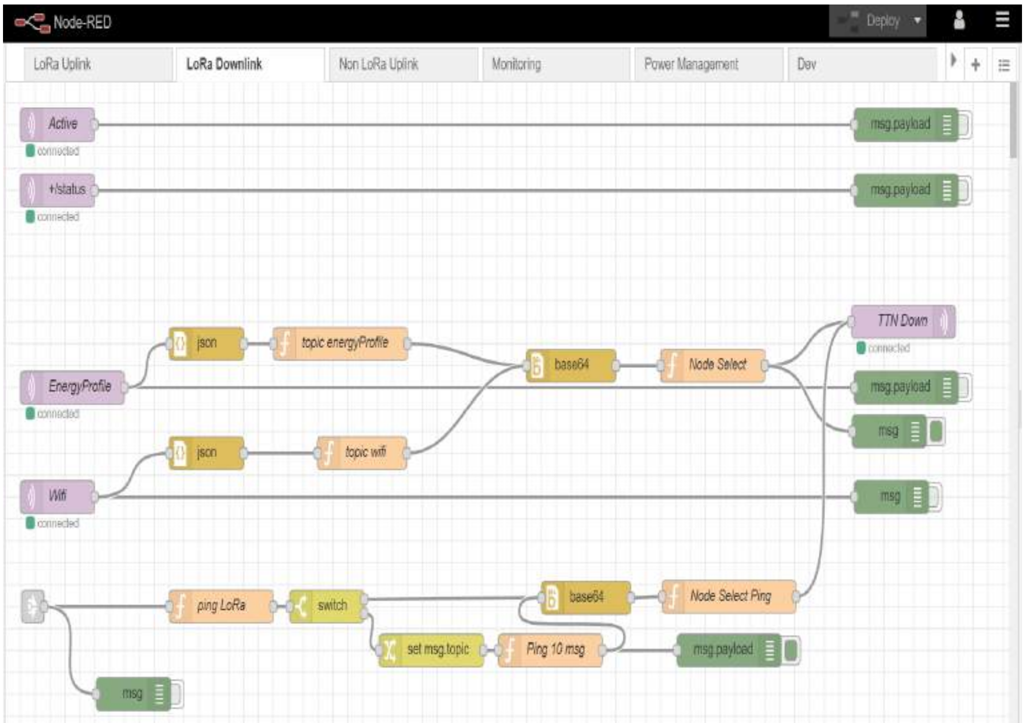


Figure 4.11: LoRa Downlink Flow

- **Power Management**

  This module is responsible for the power management of the devices and it is used mainly in the last two functional stages ("Advanced" and "Smart"). The module is represented in Figure 4.12. The working principle is the following, first, on the left side of the figure, the status topic is subscribed, then, the message is converted to JSON and is filtered only the messages from Pycom devices. The message is, analysed and the battery level is checked. If the battery level is valid, the remaining battery is calculated, as well as the percentage of the total capacity, for example, "CP20" 10 hours 90%. Then, based on this information, the active components and the sample rate of them are adjusted. For the later stage, this module could take into account if the device is paired with carer smartphone, by checking the "accompanied" flag of the status message, and by subscribing "zones" topic check where the device is working, being the possibilities the following: home, regular, dangerous. This was defined by Jorge in [60], but not implemented in this module.

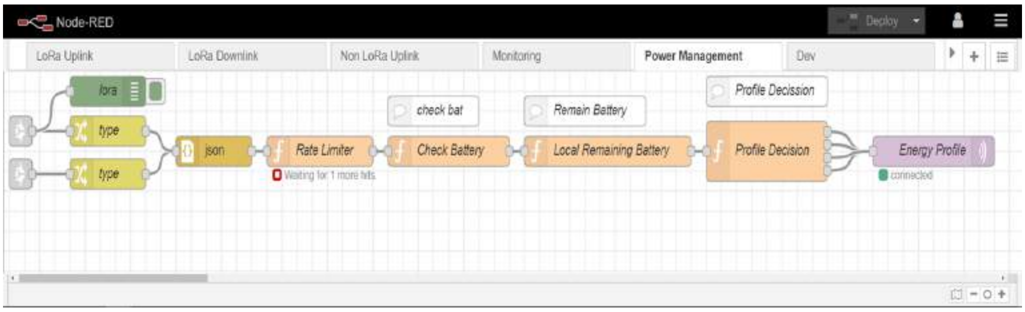


Figure 4.12: Power Management Flow

- **Non LoRa Uplink**
  This module comprehends the same functions as the "LoRa Uplink", but for Non LoRa communications, the working principle is the same, with the exception that the first section is changed by the one represented in Figure 4.13.

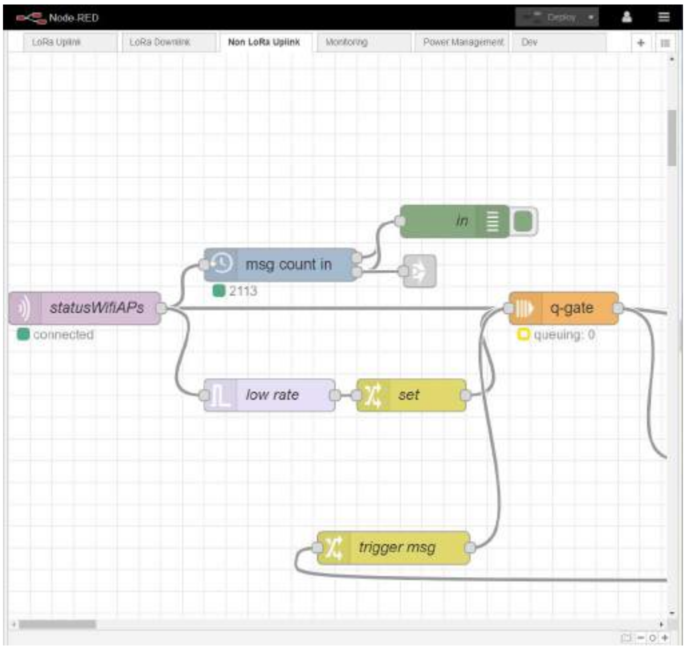


Figure 4.13: Non LoRa Uplink Flow

- **Monitoring**
  The Monitoring module described in Figure 4.14, has the ability to connect with the other modules, as it is possible to observe in the left side of the said figure. This module uses several counters in different check points and combines this information to create a report. This report is then sent by e-mail to the person in charge of the model- for this work was used a daily e-mail. This module has also another function that is always analysing the different counter and when a certain threshold is crossed, sends an SMS to the same person.

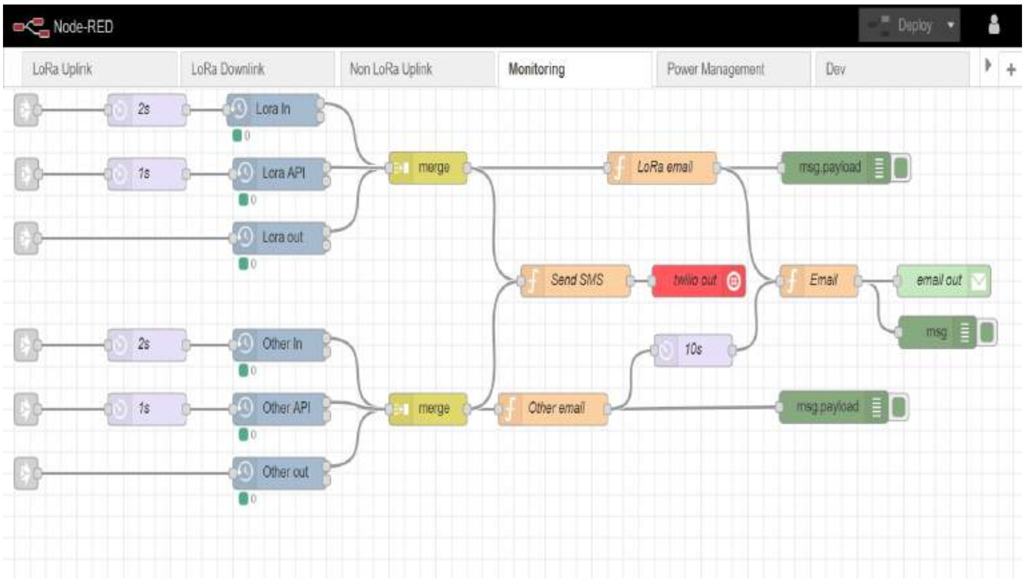
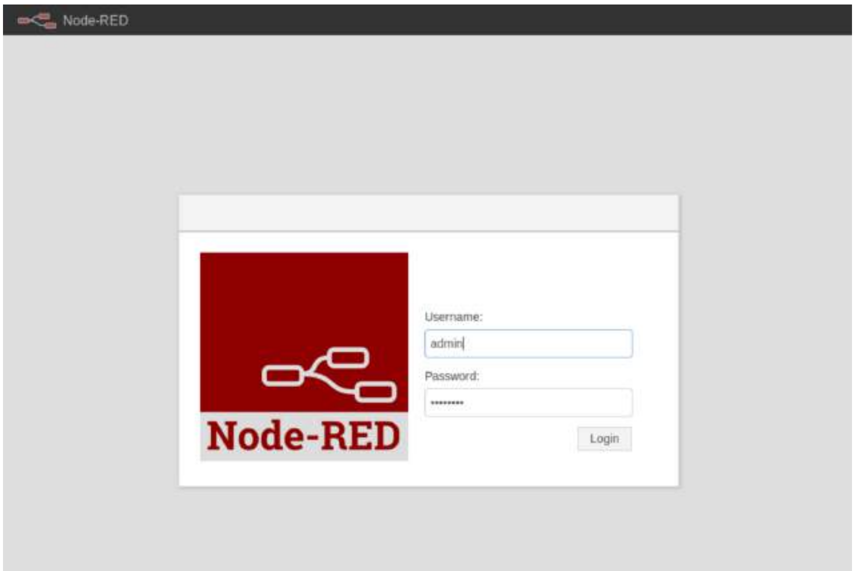


Figure 4.14: Monitoring Flow

- **Security**
  The "Security" module is not a separated flow, but a combination of functions built in the other modules. The next Figure 4.15, shows the Login Screen of the model that requires a username and password authentication, making the "Secure Login" function. The "Drop Non-Standard Messages" is done by the first function of both uplink modules.



Username: admin

Password: ********

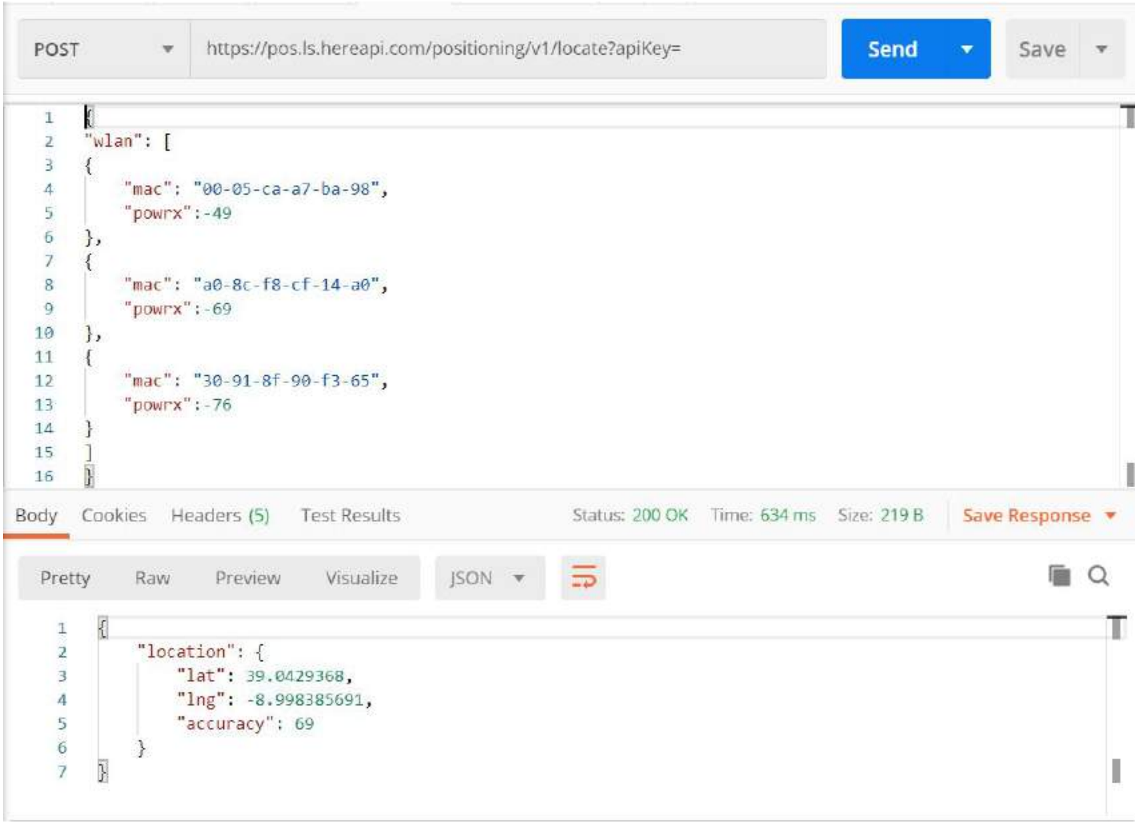
Figure 4.15: Login Screen

4.4.2 API's

An application programming interface (API) is used as an interface between the Node-Red and the server that it is responsible for doing the calculation of the location. APIs are often used as an abstraction so it is easier to use a service.

4.4.2.1 Wi-Fi

The Wi-Fi [61] assisted location is achieved through the APIs. These APIs consume the surrounding Wi-Fi Access points and the respective RSSI. With this information, they do a cross-reference check in a database and the result is sent back to the Node-Red. For this work, three APIs are used, as described in chapter 5 of Appendix2 B. The three used APIs were Google Geolocation API [62], Here [63] and OpenCellID [64]. As an alternative, the Mozilla location APi [57] was also studied, but at the moment of writing there were no keys available. The next Figure 4.16, represents a test example for the Here API, using the Postman software.



https://pos.ls.hereapi.com/positioning/v1/locate?apiKey=

```
{
"wlan": [
{
    "mac": "00-05-ca-a7-ba-98",
    "powrx":-49
},
{
    "mac": "a0-8c-f8-cf-14-a0",
    "powrx":-69
},
{
    "mac": "30-91-8f-90-f3-65",
    "powrx":-76
}
]
}
```

Status: 200 OK Time: 634 ms Size: 219 B

```
{
    "location": {
        "lat": 39.0429368,
        "lng": -8.998385691,
        "accuracy": 69
    }
}
```

Figure 4.16: Here API test

4.4.3 LoRa

The LoRa API, used in this work, has integration with TTN server. When the message is sent to the Node-Red server, the LoRa location is already presented in the Metadata. For this location to works, a minimum of three gateways able to listen to the transmitted message are needed.

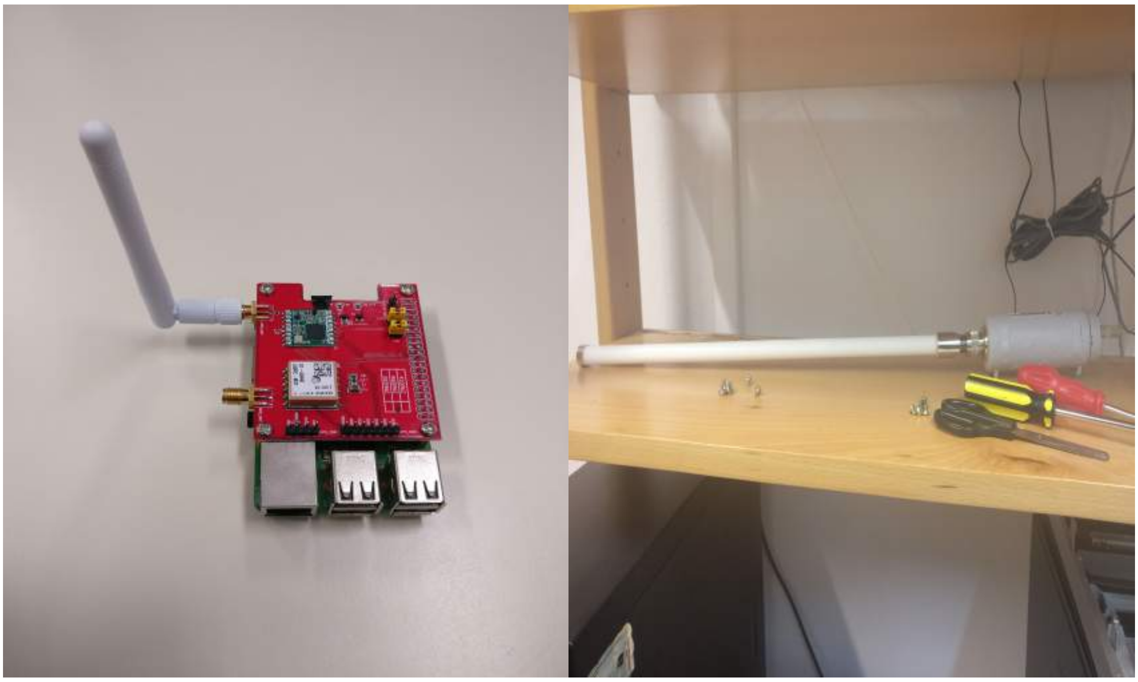
The API responsible for the calculations is LoRa Cloud [65], formerly known as Collos. This API has three versions, all of those were used for development and testing, and the Version 2 was the one with best results, the requests made were RSSI Singleframe, TDoA Singleframe, RSSI Multiframe and TDoA Multiframe.

The ability to use WiFi is also available but not used since there is another section for this type of assisted location.

The API works in the following way: TTN receives the uplink message from one or more Gateways and instead of throwing away duplicates, it keeps the meta-data from each one (primarily that is RSSI and SNR plus TOA, if available). TTN, then, forms a query with all the meta-data. The Query is posted to the endpoint in LoRa Cloud, specified in the configuration of the integration, and LoRa Cloud responds with a location.

TTN console then looks at the locations coming back and, if there are multiple, it sends the most accurate through MQTT to Node-Red.

In order to use LoRa, it was necessary to get coverage of the site where this work was developed. For this two gateways were deployed as shown in Figure 4.17, the complete guide for the instalation and configuration of the Gateways can be found at [59].



(a) Rpi with Dragino hat

(b) Lorix one

Figure 4.17: LoRa Gateways

CHAPTER 5

RESULTS

This chapter covers the results of the system. It will start by explaining the used Methodology, followed by the presentation of the use cases where this thesis is integrated. Then, the geolocation, communication resilience and power results, are presented to the reader, concluding with the a comparison and a discussion of the obtained results.

5.1 Methodology

To respond to the presented research question, this work explains the development of an Adaptive Geolocation Solver model, capable of autonomously decide the best location method. To reach a model able of the mentioned autonomous feature, it must address distinct variables as remaining battery and the availability of communication technologies. This availability can be analyzed through advanced sensing of the surrounding radio signals. This means that the model has to be aware of the environment in which the device is working. The environment's main characteristic relates to indoor or outdoor and rural or urban. These characteristics have a direct impact on what kind of technology has to be used in each working moment, for each environment. Thus, the final idea is to define a model that responds to all of these situations and, consequently develops a correspondent profile-based decision system capable to respond to each situation.

For the development of the profile-based decision system, tests were conducted into two different environments: urban and rural. Thus, these tests used geolocation technologies such as GNSS, Wi-Fi and LoRa. Furthermore, these tests will consist of the gathering information from different location coordinates, at different speeds as well as stationary, to evaluate the performance of the different technologies against each other.

In addition, these tests were integrated into the Carelink project. It was used a geo-fencing polygon, previously defined by the carer of the PwD in the Carelink platform. Consequently, the device communicates with the platform to give its position and receive support on the technologies to use accordingly to the profiles of the respective PwD. As such, in the end, these tests were conducted with real patients in real life trials. To perform the communication between the devices and the Carelink platform it was utilised LoRa and NB-IoT.

5.1.1 Material

In order to conduct the tests of this work, it was selected 5 FiPy devices, used in the belt box format. That were used by the participants. The remaining material needed was a LoRa Gateway to ensure the coverage of LoRa signal, and a laptop to analyse the data.

5.1.2 Participants

The participants in this experiment were 6. Later in this chapter will be described the three different environments where the tests took place. The participant for the first environment is male 24 and does not have dementia. In the second environment, the first tests were conducted two participants, both males, of 24 and 60 years old, both without dementia. After these tests were concluded, the final ones were performed with 10 persons with dementia, 5 males and 5 females, with ages between 60 and 100 years old, and with a dementia level from mild to moderate. Amongst these participants, only 5 of them used the Fipy devices.

5.1.3 Procedure

The adopted procedure for this work, during the development phase, was to first use a FiPy; each location method was tested individually and afterwards followed the testing of each communication method. Later, all the methods were combined. In the Real People with dementia test phase, one device was given to each participant to use during the day. The data from these tests were then analysed by the author.

5.2 Use Cases

In this section are described the different use cases for this thesis. The three use cases are FCT-Laboratory, Lisboa- Field Test and Swiss- Real People with Dementia. In these use cases the objective was to get the results for the geolocation. To accomplish this goal, tests were conducted for the geolocation with both GNSS and GNSS-free technologies. These tests consisted of gathering information from different location points, at different speeds, to discover the geolocation accuracy of such technologies.

In the following table 5.1, is possible to observe the Maximum and Minimum Accuracy for the 4 geolocation technologies used in this section. These values will serve as reference for the ones obtained during the tests.

	Max Accuracy	Min Accuracy
BLE	1	7
GNSS	1	15
WiFi	20	40
LoRa	20	2000

Table 5.1: Device Specification Accuracy (m)

5.2.1 FCT- Laboratory

The first use case described for this section is the Laboratory scenario. This is where all of the development phase occurred and is situated in the University campus of the Faculty of Sciences and Technologies from the Nova University of Lisbon.

5.2.1.1 BLE

The first Geolocation method tested was the BLE Beacons. For this experiment was used two FiPy, one in the belt box enclosure and another without the enclosure. The antennas used were the internal antennas.

The test consisted of one fixed FiPy acting as BLE Beacon and another mobile one simulating the real patient. The test was conducted indoors since the main usage for this type of Beacons is indoor usage, although they can work outside, but, due to their short range, it is not the best solution for large outdoor spaces.

In the Beacon side was created an advertisement for the beacon called $Pybeacon$ and a service called $GPS-Bluetooth-B1$, where the following data is presented $BLE->coordinates$, then exits a variable called GPS where the actual coordinates can be set by the user.

In the device side, there is a scan for Bluetooth advertisers, if one matches the $Pybeacon$ the device reads the values from the service, getting the data from the Beacon. This can also be done for managing the accompanied flag, that was presented in 3.1. This flag is used to know if the device can listen to the Bluetooth name from the smartphone of the carer in charge of the PwD.

The location accuracy is the one defined by the user, so it is going to be the best possible. The communication distance using both internal antennas was 2 m with obstacles, and 5m with a line of sight. This value can increase to around 10 m with the use of an external antenna in the beacon side. The time used to retrieve the location is approximately 4s and the BLE cannot be used to transmit the full Status message.

	Max accuracy	Min Accuracy	Communication	Time
BLE Indoor	1 m	10 m	No	4 s

Table 5.2: BLE Results

5.2.1.2 GNSS

During the development phase in the laboratory scenario, the results obtained for the GNSS were done using one FiPy and one Pytrack shield. The Pytrack [66] shield has a Quectel GNSS L76-L [67], this receiver module supports Multi-GNSS including GPS, GLONASS, Galileo and QZSS systems. The antenna used for the GNSS system was the one from the shield (surface mounted 6 by 2 mm)instead of an external one, due to the design dimension constrains.

The test consisted, initially in obtain an indoor position, which, as expected, failed. The test was then conducted outdoor using LoRa as communication method and at stationary speeds. The results obtained are expressed in the next figure 5.1.

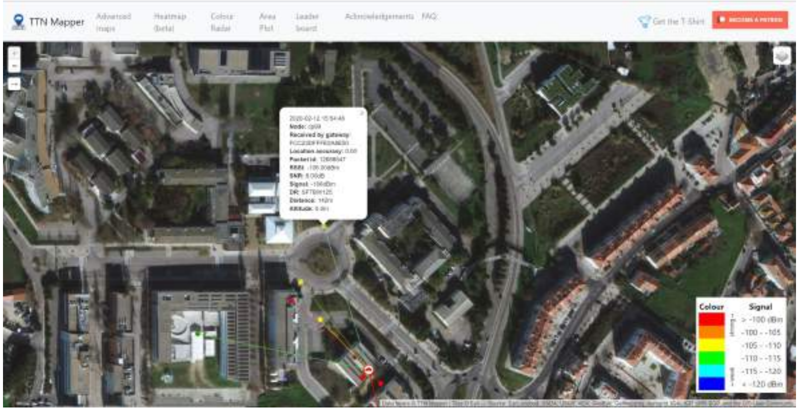


Figure 5.1: GNSS

The next test consisted of getting GNSS Location while walking, considering the walking speed of 5km/h, to simulate a tracking situation. The TTFF (Time to first fix) announced is between 1 to 35 seconds. The one from the tests is closer to 120 seconds from a cold start. The result, from using the device in a clear day and waiting for 5 minutes to fix position, and then start walking, was good, but often during 40 minutes walk the device misses some points and these points return the location empty. This can be due to the fact the GNSS has 33 channels for tracking and 99 for the acquisition of the location. The problem verified with the requisitions of the location could be solved by using an external antenna.

The location accuracy is from 5 m to 10 m. The time used to retrieve the first location after a cold start was approximately 120 seconds and the GNSS cannot be used to transmit the full Status message. The next Table 5.3 represents the resume of this geolocation technique.

	Max accuracy	Min Accuracy	Communication	Time
GNSS Outdoor	5 m	10 m	No	120* s

Table 5.3: GNSS Results

5.2.1.3 WiFi

The following Geolocation method tested was the WiFi assisted location. For this experiment was used the same FiPy as above, in the belt box enclosure. The WiFi antenna used was the internal one. For the software was used the Geolocation API from Google.

The test consisted of one FiPy mobile that was simulating the patient. The test took place indoors, since it was first tested using WiFi as a communication method, therefore the maximum coverage of the WiFi AP only permitted indoor usage (and much less outdoors).

Figure 5.2: WiFi lab

In the device side there was a scan for WiFi APs. The device started by reading the values from the APs (access points or routers). These values consisted of Service Set Identifier (SSID) which is the WiFi name of the network, BSSID (basic service set identifier) that is the MAC address of the AP, RSSI (received signal strength indication) that is the power present in a received radio signal. There were other fields advertised such as the network channel or security, but they were all discarded since they were not useful in this situation. After getting this information, a JSON object was generated in the device, to construct a request for the google API. The Google API then returns the approximate location as its possible to observe in the Figure. 5.2.

For this test, the JSON field *considerIp* was set to false, otherwise, Google takes the IP of the connection to identify the position, which later will not be useful when WiFi is not used as the communication technology. By doing this, it was necessary to handle the errors that existed when there were no WiFi APs, or Google databases had no position information about the previous sent WiFi APs.

The returned location accuracy was from 15 m to 100 m. The time used to retrieve the location was approximately 3 sec and the WiFi can be used to transmit the full Status message. The next Table 5.4 represents the resume of this assisted geolocation technique.

	Max accuracy	Min Accuracy	Communication	Time
WiFi Indoor	15 m	100 m	Yes	3 s

Table 5.4: WiFi Results

5.2.1.4 LoRa

The last Geolocation method tested was LoRa as a GNSS-free geolocation. For this experiment was used the same FiPy and for the first experiment were used three other FiPy acting as gateways. The LoRa antenna used was an external one (SMA Tilt Swivel 1/2 Wave Whip Dipole). For the software was used the LoRa Cloud (formerly Collos) API from Semtech, and the TTN network server.

The first test consisted of using one FiPy mobile simulating the patient and three others acting as gateways. For the gateways was used the LoRaWAN Nano-Gateway example from pycom.The only issue encountered was with the internet from the campus, that was blocking the UDP packets, so the gateways were not able to sync with NTP servers and, thus, were not capable to forward the messages. This problem was solved by using a smartphone as a WiFi hot-spot. The test was conducted indoor, with the gateways separated from each other in a triangle shape. This test failed to obtain the location, mainly due to the fact the gateways were closer than the maximum accuracy for this method.

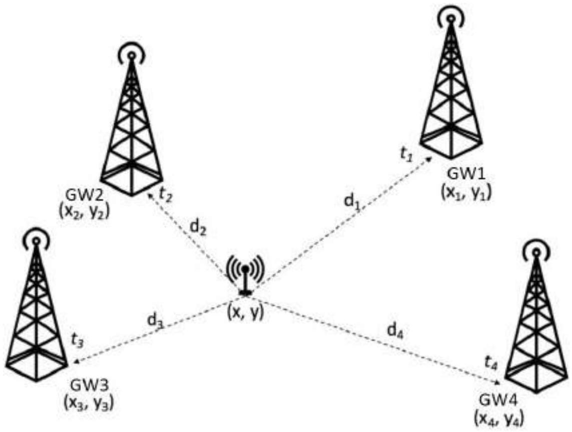


Figure 5.3: Lora 4 Gateways Location

The second test consisted of the same device, but now was conducted outdoor. For the gateways were used three although a fourth one was also in range by using the external antenna. The Figure 5.3, describes the test scenario and the positioning of the three used gateways, that can be seen in the next Figure 5.4. The right one is a raspberry pi 3 with a dragino hat, on the roof of a building, 230 meters away from the middle one. That was LoRix One at the time of the test was inside of the Uninova CTS building, and the last one 80m was also indoor in the Electrical Engineering department and was an ESP based Gateway. The other Gateway was located in Lisbon, 8km away, and it was also a LoRix one.

In the device side was sent a small 1 byte payload with the following specifications: spreading factor 7; bandwidth 125 kHz; the first channels in 868.1 MHz and 868.3MHZ; the coding rate at 4/5 and the LoRa class as C. If other specifications it were used the results could be others, but, due to the fact there were 2 single-channel gateways, the author considered that these were the best ones to ensure that all of the gateways could listen to the message. In the gateway side was simple a packet forwarded to the TTN server. In the TTN server was an Integrations with the LoRacloud

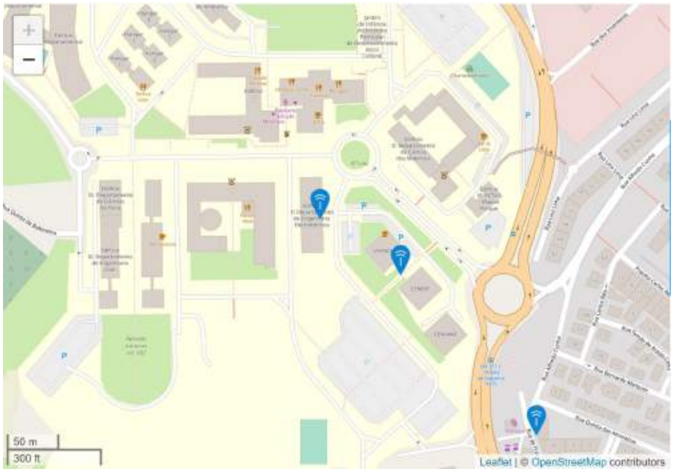
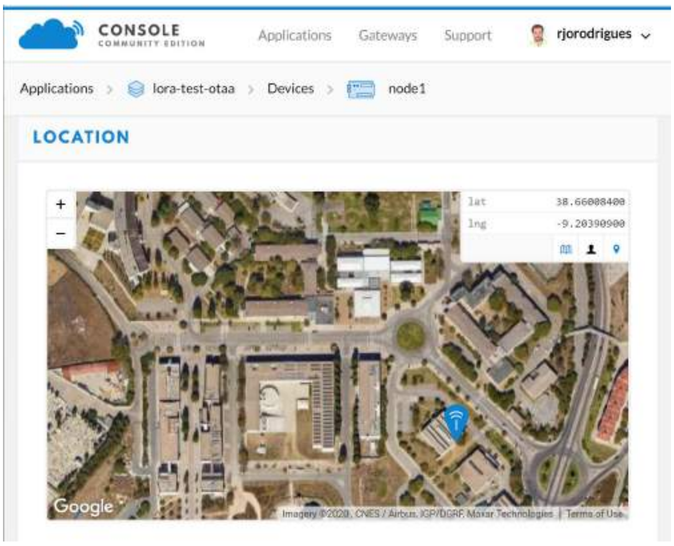


Figure 5.4: LoRa GWs

API. Using the LoRa Metadata from the messages a location was then returned, as it is possible to observe in the next Figure 5.5.



lat 38.66008400
lng -9.20390900

Figure 5.5: LoRa lab

The returned location accuracy ranges from 20 m to 200 m. The time used to retrieve the location was approximately 2 sec and LoRa was used to transmit the Status message, although it was not the full status message, it was only the full values from the Status message. These values were later constructed in the TTN Network server. The next Table 5.5 represents the resume of this GNNS-Free geolocation technique.

	Max accuracy	Min Accuracy	Communication	Time
LoRa Outdoor	20 m	200 m	Yes	2 s

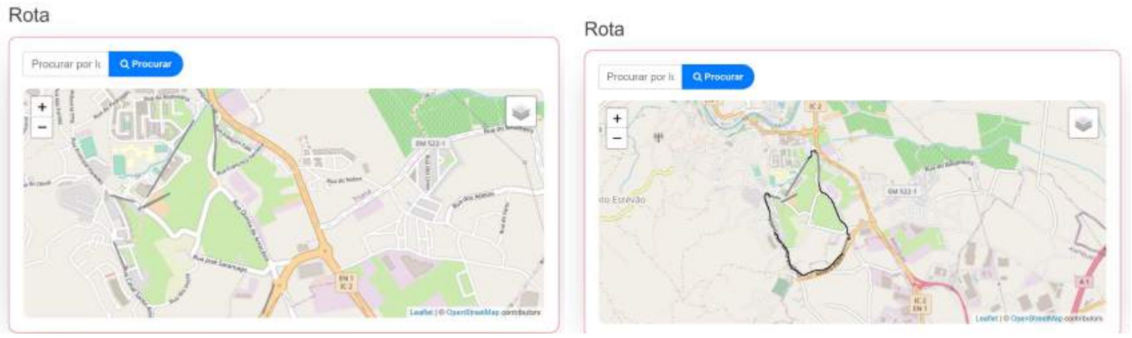
Table 5.5: LoRa Results

5.2.2 Lisboa- Field Test

The second use case described for this section is the Field Test. This use case was separated into two test locations. The first one intended to test the results from the lab but in the urban environment, which took place in Lisbon. The second location was Alenquer (40 km away from Lisbon) to simulate the rural environment.

5.2.2.1 GNSS

The GNSS test, in this scenario, used the same material as above, in 5.2.1.2. The communication method used in this case was NB-IoT. And the test consisted of gathering GNSS Location during walking, at an average speed of 5km/h, to test a real tracking situation. GNSS took about two minutes to acquire a position after a cold start. The result of using the device in a clear day, after waiting for around 5 minutes to fix position and then start walking, are possible to observe in the next Figure 5.6. On the left side(5.6a) is the registered GNSS path and on the right (5.6b) is the real path. The walk lasted for 50 minutes,non-stop, with a 4.3 km distance (about 100 location points), in a rural environment, so none of the locations points are from stationary locations. Comparing both figures it is possible to observe that some points are missing, this occurred during the reacquisition of the location since the sample time was 30 seconds. The problem verified with the requisitions of the location can be solved by using an external and more powerful (higher gain) antenna.

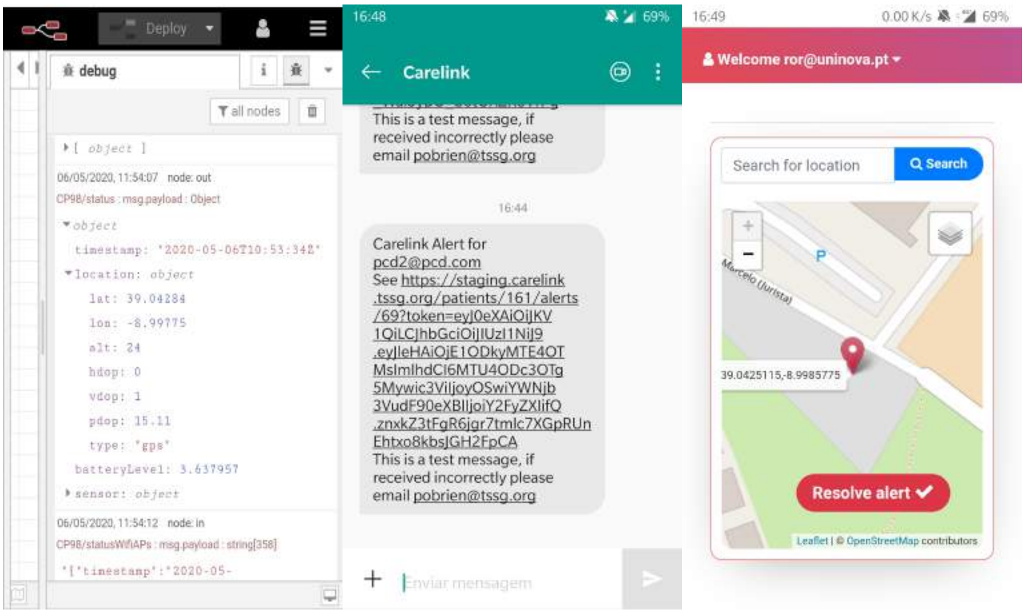


(a) GNSS path

(b) Real path

Figure 5.6: GNSS Results [6]

The next test consisted in crossing from a defined safe location area onto an unsafe zone and acquiring GNSS location. This location took place outdoors, at stationary speed. The device sent the location to the model, as it is possible to observe in the first Figure(5.7a). The model sent the information to the Carelink platform, which raised an SMS alert, described in the middle Figure 5.7b. Where there was a link to see the alert, as well as, the position of the patient, as shown in the Figure(5.7c). The combination of these three Figures 5.7 is the workflow behind an Unsafe zone Alert, here described working with GNSS location.



```
▸[ object ]
06/05/2020, 11:54:07   node: out
CP98/status : msg.payload : Object
▾object
  timestamp: '2020-05-06T10:53:34Z'
 ▾location: object
    lat: 39.04284
    lon: -8.99775
    alt: 24
    hdop: 0
    vdop: 1
    pdop: 15.11
    type: 'gps'
  batteryLevel: 3.637957
 ▸sensor: object
06/05/2020, 11:54:12   node: in
CP98/statusWifiAPs : msg.payload : string[358]
 '[{'timestamp':'2020-05-
```

This is a test message, if received incorrectly please email pobrien@tssg.org

16:44

Carelink Alert for pcd2@pcd.com See https://staging.carelink.tssg.org/patients/161/alerts/69?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE1ODkyMTE4OTMsImlhdCI6MTU4ODc3OTg5Mywic3ViIjoyOSwiYWNjb3VudF90eXBlIjoiY2FyZXIifQ.znxkZ3tFgR6jgr7tmlc7XGpRUnEhtxo8kbsJGH2FpCA This is a test message, if received incorrectly please email pobrien@tssg.org

(a) GPS Node Red (b) SMS Alert (c) Web Alert [6]

Figure 5.7: Unsafe Zone Alert

The location accuracy was from 5 m to 20 m. The time used to retrieve the location was approximately 120 seconds after a cold start, and later it was immediate.

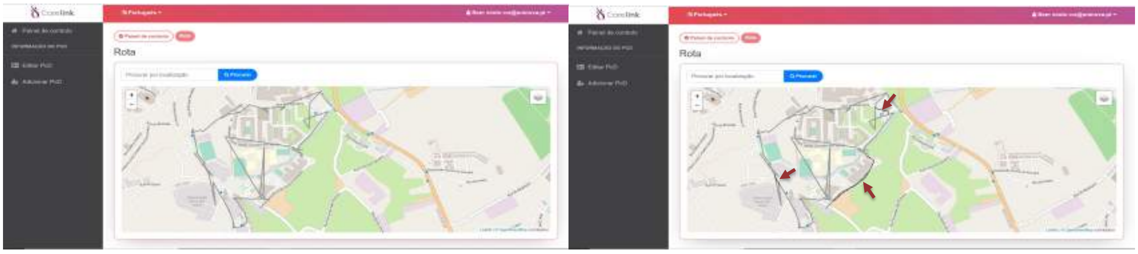
5.2.2.2 WiFi

For the WiFi assisted location test was used the same material as above. The test had the same objective, that was simulating a patient walking.The test was conducted outdoors and it was different from the previous scenario, in the way that the communication used was NB-IoT, allowing for more mobility and a bigger range for the walk.

The test consisted in using the WiFi passive scanning principle, to discover WiFi access points. This information was then used to do the assisted Location during the walk, at the average speed of 5km/h.

The result of using the device in an urban route is possible to observe in the next Figure 5.8, on the left side (5.8a) is the registered GNSS path, and on the right (5.8b) is the real path. The walk lasted for 40 minutes, having a 2.8 km distance (about 80 location points), in an urban environment. Comparing both figures, is possible to observe that in the figure from the right, in the bottom exists a residential area, close to the green area, that was no problem for the WiFi assisted location, but close to the other green area exist a sports hall with a WiFi network and a school with no WiFi, so this method was inaccurate in this part of the walk. The optimal scenario for this assisted location would be urban outdoor, where accuracies closer to GNSS can be achieved.

For this method to work, was used the same principle as above, wherein the device side there was a scan for WiFi AP's. The device reads the SSID, BSSID and the RSSI. After this, it generates an array and sends it to the model. In the model, this array is converted to a JSON to construct a request for the three used APIs (Google, Here, OpenCellID). The APIs then return the



(a) WiFi path (b) Real path

Figure 5.8: WiFi Results [6]

approximate location in different formats. These formats were standardized and the most accurate one was chosen and sent to the platform, as it is possible to observe in the Figure 5.8a. The returned location accuracy is from 15m to 200m. The time used to retrieve the location was approximately 3 sec.

5.2.2.3 LoRa

The LoRa Geolocation method was the last one tested. For this experiment was used the same FiPy. The LoRa antenna used was an external one (molex ISM 105262, omnidirectional with 0.4 dBi Peak Gain at 868MHZ). In terms of software was used the LoRa Cloud (formerly Collos) API from Semtech, the TTN network server, and Cayenne [68].

The objective of the test consisted of using the Metadata present in the LoRa messages, to get an assisted location for one mobile FiPy, that was simulating the real patient. The test was a walk with a duration of 30 minutes, in an outdoor scenario, for a distance of 2km, at an average walking speed of 5km/h. The result for using the device in an urban route is possible to observe in the next Figure 5.9.

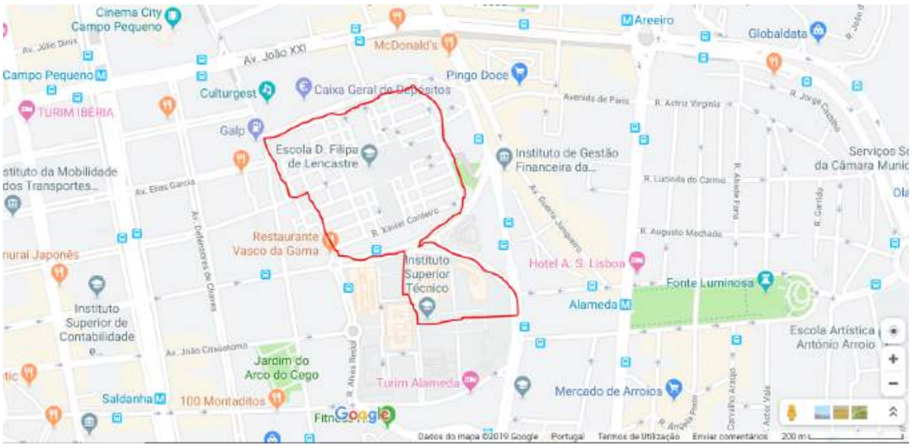


Figure 5.9: LoRa Real path

The results of this test were dependent on the number of gateways in reach of the device, to perform the multilateration a minimum of three gateways were needed. So, the location of the test was chosen taking in consideration this constrains. With that in mind, one of the best places in Lisbon, that met all the conditions at the time of the test, was the one indicated on Figure 5.9. For the testing were used three community gateways, with the objective to simulate a real usage scenario. From the data in figure 5.10, it can be seen that all of them covered the test area, which is represented in red.

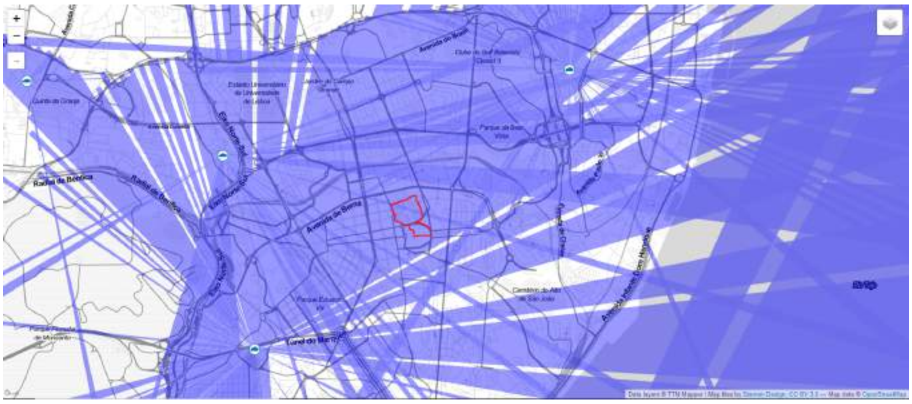
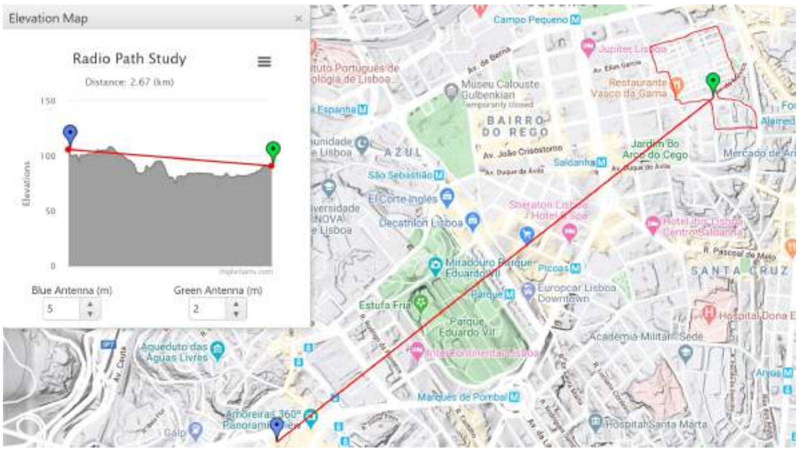


Figure 5.10: LoRa Gateways

The test was also conducted in an area central to all of the three gateways. The Figure 5.11 shows the first one at a distance of 2.67 km; the second, in Figure 5.12, is at 2.85 km and the last one, in Figure 5.13, is at 2.66 km.

From these figures it appears that none of the gateways had a direct line of sight. Although, all of them were placed outdoors, and perhaps on top of buildings, so it is possible they could have a line of sight, since the figures only represent terrain elevation and leave out building elevation. The first one is registered at 175m; the second at 100m and the third is unknown - this information was from the TTN [53] and it was set by the owner of the gateways.



Blue Antenna (m) 5

Green Antenna (m) 2

Figure 5.11: GW1 Distance and Line of Sight

Given the previously described information, a concentration of points closer to the first gateway could be expected, since it is the closest one with a better chance for having line of sight.

The third gateway should be the one with more difficulties to receive the packets. In the device-side was sent the full payload. The configurations in use were the following: spreading

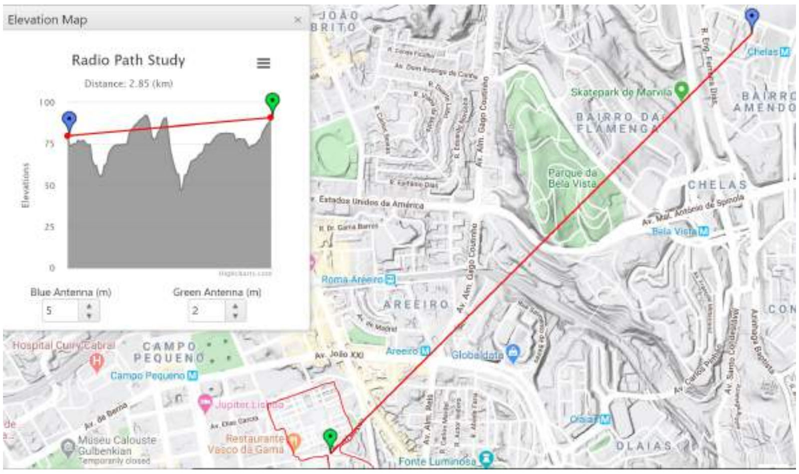


Figure 5.12: GW2 Distance and Line of Sight

factor 7; bandwidth 125 kHz; the first channels in 868.1 MHz; the coding rate at 4/5; the LoRa class as C and a Transmission power of 14 dBm. The use of other specifications could result in other outcomes, or using the adaptive data rate to change the Spreading factor could result in better reception from the gateways, but these specifications are the most common one from low-end gateways and this test tried to replicate the normal usage from a patient anywhere in the world.

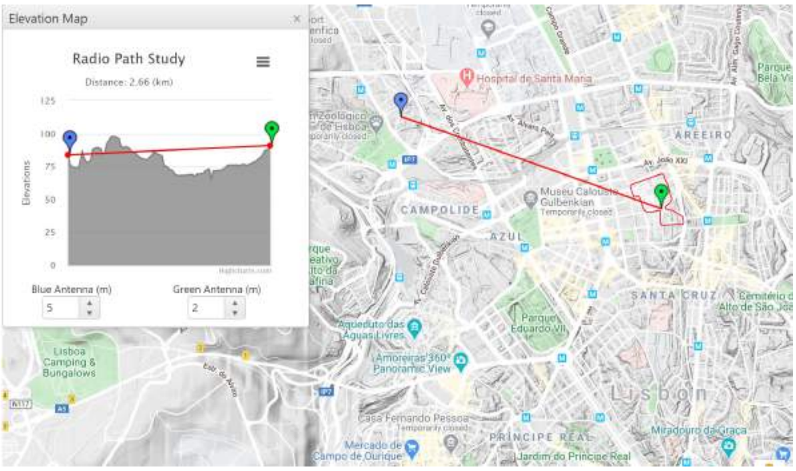


Figure 5.13: GW3 Distance and Line of Sight

In the gateway side was simple a packet forwarded to the TTN server, the gateway added the Metadata to the message. In the TTN server was an integration with the LoRacloud API.

The first result for the location is represented in the next Figure 5.14, where is shown a map of all the locations points, this map was part of the Cayenne dashboard. In this dashboard, the blue circles represents a location from the device, the red line that is connecting the circles represents the route and the blue balloon stands for the location of the device in a specific time, this one can be changed on the bottom slider.

For this first result was used the LoRacloud V1 single frame RSSI location. Which calculates a location for a device according to RSSI data, received by the gateway. The property "Location" of the gateway was not required. However, this property can be used if the gateway and its antenna location were not provisioned in the system. This means that the location will be more accurate if the gateways have GNSS capabilities, or the location of them are accurately defined in the TTN.

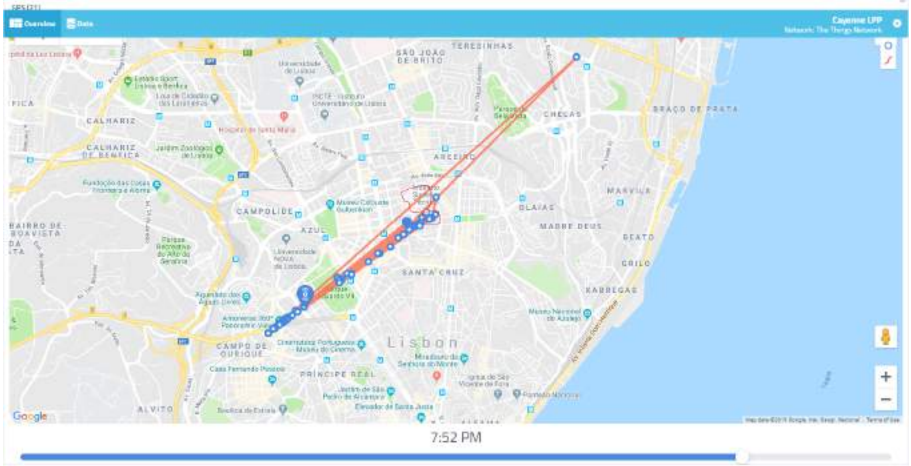


Figure 5.14: LoRa Location RSSI All points

In Figure 5.14, is possible to observe that the location points float within the Gateway 1 and 2, but are closer to the first. This is caused by the fact the first one was receiving the messages with better RSSI. The factors causing this phenomenon could be either a higher gain antenna on the receiver, a better line of sight, or a multipath route for the test location.

In The next Figure 5.15, is shown a zoom of the test location, therefore removing all the others outliers, and it is possible to infer that some of the points are close to the real path (red line). The announced accuracy for this method, previous stated in 2.18, was 1000 to 2000 meters while the observed was around 30 to 3000 meters.

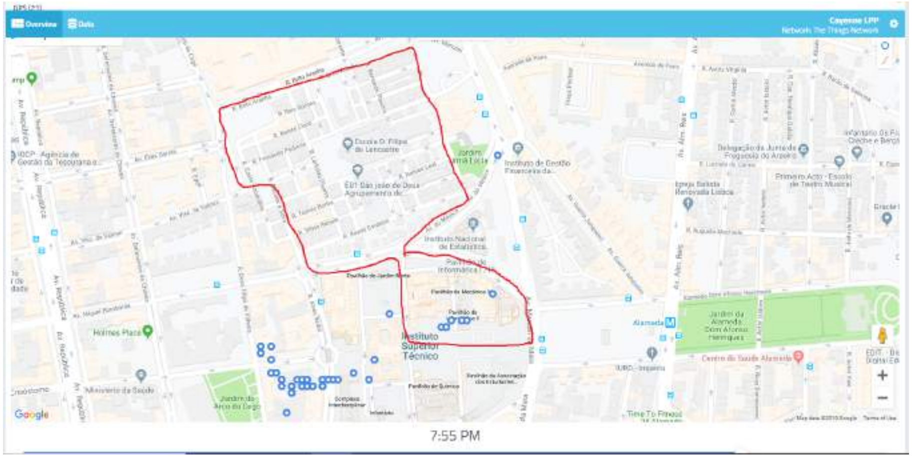


Figure 5.15: LoRa Location RSSI Zoom

For the second result was used the LoRa cloud V2 TDoA, which solves the location for the device according to RSSI Metadata combined with high-resolution time of arrival (TOA) Metadata. The time of arrival is measured in nano-seconds and is only supported by LoRa gateways with the necessary high-resolution time-stamping features, that are the ones with GNSS. The Figure 5.16 shows all of the points.

Figure 5.16: LoRa Location TDoA all pooints

Concerning the previous method, is possible to observe fewer outliers and also fewer points when the zoom is done in Figure 5.17. This was possibly caused due to the restrictions of having a high-resolution time-stamping and, perhaps, only the gateway 1 had it or, at least, had the better time-stamping. The announced accuracy for this method was 20 to 200 meters and, if the gateway location point is considered an outlier the result was close, from 30m to 350m. Could have been improved maybe with other gateways, but, comparing Figure 5.16 with Figure 5.14,it was already an improvement.

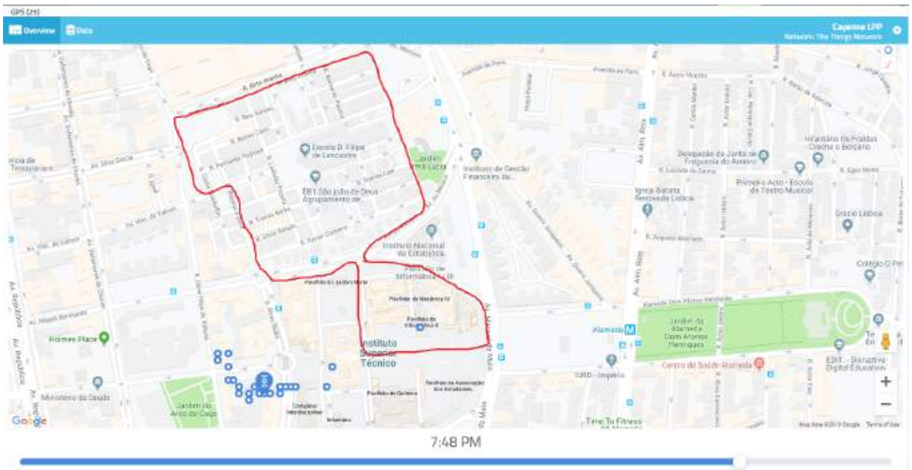


Figure 5.17: LoRa Location TDoA Zoom

The last results were from LoRa cloud V3 Multiframe. This technique calculates the location for a device according to the RSSI data captured from multiple LoRa frames (uplinks). This technique uses a sequence of single frames (consisting of multiple uplink objects) that are typically received by the gateways within a short time frame. These frames are assumed to be transmitted by the device in the same location. The result is a single location estimation.

Using the sequence of radio frames for a single estimation allows to determine a location with higher accuracy than the single-frame alternative. The multi-frame technique also combines all the available data (RSSI, TDoA, SNR) into a single calculation. This results in a location estimation that is generally more accurate than the average of multiple locations, that were calculated from several single frames.

This technique will work better with higher sample rates from the device, or with a stationary location. The following figure 5.18, represents the location points for this method. Comparing to the previous method the results were almost the same due to the fact that the locations were retrieved during walking, and were not stationary, and the sample time of 30 seconds was maybe too high for getting multiple frames in the same location, or at least in a close enough location.

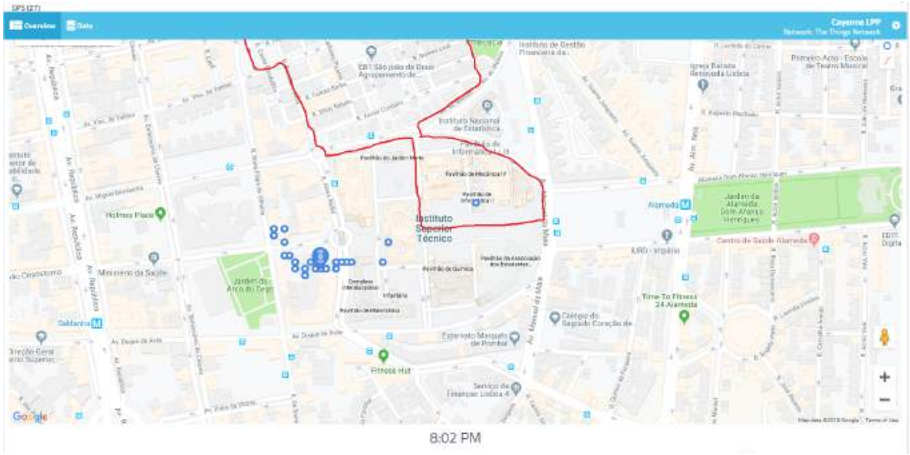


Figure 5.18: LoRa Location RSSI All points

The returned location accuracy for the 3 methods ranged from 30 m to 3000 m. The time used to retrieve the location is approximately 2 sec and LoRa was used to transmit the Status message, although was not the full status message, it was only the full values from the Status message, this one was later constructed in the TTN Network server.

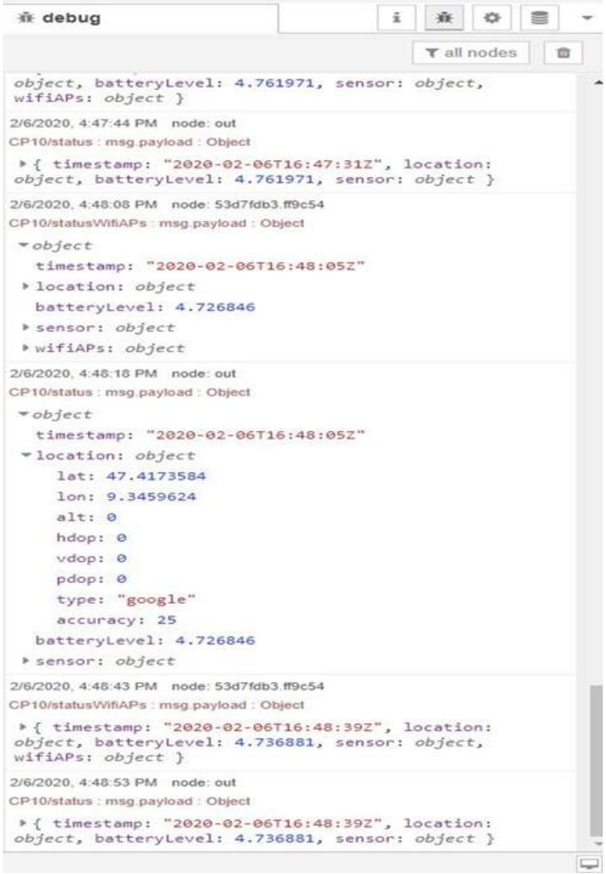
In short, LoRa had the worst accuracy from all of the methods, but, at the same time, was better in the battery usage. The previous results are, thus, conditioned by the fact that the Gateway 1 was better than the others. This location method has the possibility of creation of an ad hoc network, ensuring this way better location results. The optimal scenario for this assisted location will be urban or rural outdoor.

5.2.3 Swiss- Real People with Dementia

The last use case was held in Switzerland, where the first stage of the solution, previously described in Chapter 3, called hierarchical (where the decision is made using the approximate accuracy of each geolocation method) performed trials with real people with dementia.

The model, after the previous testing phases, needed to be validated, therefore was used in this use case. The first results for the model validation, obtained during the initial phase of trials, were a total of over 50000 received messages from 10/02/2020 until 10/03/2020. The difference in the received messages and the sent messages was less than 1%. The implemented system was able to support spikes of simultaneously received messages from different devices, and the tested version was able to process 1 message per 2 seconds. These first tests were conducted using 5 devices.

In the next Figure 5.19, is possible to observe the output, of the system in a real life test situation. In this figure is represented the device "CP10", which was used by a PwD and, first is possible to observe that the message from the device contains a JSON object with the "wifiAPs" field, which was used to perform the assisted location. In this situation, the assisted location was performed by the google API, as it is possible to verify by field type in the location object . This means that the system received a location from the GNSS, but this location was not valid, and then used the "wifiAPs" field do perform the assisted location, replacing the invalid GNSS data with this information. With this picture is possible to prove that system was working.



```
object, batteryLevel: 4.761971, sensor: object,
wifiAPs: object }
2/6/2020, 4:47:44 PM   node: out
CP10/status : msg.payload : Object
 ▸{ timestamp: "2020-02-06T16:47:31Z", location:
object, batteryLevel: 4.761971, sensor: object }
2/6/2020, 4:48:08 PM   node: 53d7fdb3.ff9c54
CP10/statusWifiAPs : msg.payload : Object
 ▾object
   timestamp: "2020-02-06T16:48:05Z"
  ▸location: object
   batteryLevel: 4.726846
  ▸sensor: object
  ▸wifiAPs: object
2/6/2020, 4:48:18 PM   node: out
CP10/status : msg.payload : Object
 ▾object
   timestamp: "2020-02-06T16:48:05Z"
  ▾location: object
      lat: 47.4173584
      lon: 9.3459624
      alt: 0
      hdop: 0
      vdop: 0
      pdop: 0
      type: "google"
      accuracy: 25
   batteryLevel: 4.726846
  ▸sensor: object
2/6/2020, 4:48:43 PM   node: 53d7fdb3.ff9c54
CP10/statusWifiAPs : msg.payload : Object
 ▸{ timestamp: "2020-02-06T16:48:39Z", location:
object, batteryLevel: 4.736881, sensor: object,
wifiAPs: object }
2/6/2020, 4:48:53 PM   node: out
CP10/status : msg.payload : Object
 ▸{ timestamp: "2020-02-06T16:48:39Z", location:
object, batteryLevel: 4.736881, sensor: object }
```

Figure 5.19: Model Debug Output

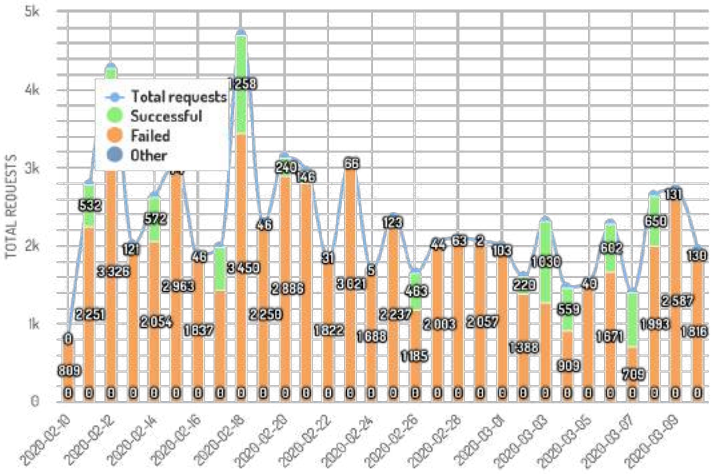


Figure 5.20: OpenCell ID

From the data in the chart 5.20, referent to OpenCellID API, is possible to conclude that over 50000 calls were made, meaning that the system has often used the WiFi data, and from the data of the same chart is possible to observe that the result of the location failed several times, that is the reason why it was important to have 3 APIs, has location sources.

To conclude, the system was able to dynamically choose the best location, from the information received. Unfortunately, due to the Covid-19 pandemic, the trials with elderly people were stopped, which means that the validation for the second stage was not done within real-life trials, being only tested by the author as described in the following section 5.3.

5.3 Other Analysis

In this section will be presented the other analysis needed for the implementation of the second functional stage, which is the "Advanced". This section will comprise the communication resilience analysis and the power management analysis.

5.3.1 Communication Resilience

In this analysis, the author performed tests for communication resilience. These tests were done to verify the communication resilience that was first introduced in Figure 4.5 from Chapter 4, since they were necessary for implementing the second functional stage.

These tests were conducted to the two different methods: LoRa and NB-IoT. To try out the communication resilience, the communication fallback capabilities were tested. If the device loses the NB-IoT link, it should fall back to using the LoRa stack. In case the communication network changes, then the Adaptive Geolocation Solver service should be able to handle the data transition and continue the processing of information. It should also be possible to force the change of the communication technology based on the energy efficiency settings.

The work done for the communication resilience in the device side was not done directly by the author, instead it was done by Researcher Jorge Calado and tested in this section.

The first test comprised using the device outdoor, with good NB-IoT signal strength, and then entering an underground parking lot, which was previously examined and had no NB-IoT coverage. A LoRa gateway was set inside the parking lot, to provide LoRa coverage. The test was conducted 5 times, and the device always changed to LoRa after a pre-defined connection time-out of the NB-IoT.

With this result, in the system side was possible to observe that with the change of communication and the type of the received message, the system could return the same output.

The second test consisted in sending a downlink message in a JSON format to the device with the communication technology and a boolean value, such as ‘{”ltenb“: “False”, “lora“: “True”}’.

On the device side, the response was similar to the previous test. The device received and decoded the JSON information and started the connection to the LoRa network. If the LoRa connection was unsuccessful, the device reboots and starts transmitting again with NB-IoT.

5.3.2 Power Management

For the purpose of getting the results for power consumption, needed in the implementation of the Advanced functional stage. An analysis with several tests was conducted to the different communication and location methods. These tests consisted of logging information from the device, with the aim to discover the battery duration, current and power, to perform more efficient power management. From the device manufacturer it is possible to observe the next Figure 5.21, where some values are already provided. All of the tests were conducted in the same fix place, at a stationary speed, with the same 2 FiPy and 2 LiPo batteries with 3.7V and 800mAh, the results are the average from the 4 tests.

Mode	Min	Avg.	Max	Units
Idle (no radios)	–	62.7	–	mA
Sigfox†	–	192	–	mA
LTE Transmit	–	TBD	–	mA
WiFi AP	–	126	–	mA
WiFi client	–	137	–	mA
Bluetooth	–	121	–	mA
Deep sleep*	–	TBD	–	mA

Figure 5.21: Power Table Fipy [49]

5.3.2.1 LoRa

The first communication method studied was LoRa. To prepare this test it was used the two FiPys and, after letting the batteries charge for 3 hours with a 2.1A charger connected to the devices. The first test was initiated using only communication, the Fipy was combined with a pysense (without GNSS and Accelerometer) shield, the initial battery was around 4.1V and the end battery, as expected, was 3.3V. The next test was done using the communication but reading one accelerometer sensor, and getting the GPS location, the only difference here was the use of pytrack instead of pysense. The result for the battery duration is lower as expected. For the last test was introduced also the WiFi scan for nearby access points. The data for all of the tests is in the next Table 5.6, each test was considered an operation mode.

	Communication Only	Communication GNSS, Sensor	Communication GNSS, Sensor, Scan WiFi
Battery Duration	12h	9h	8h 30m
Average Current	65mA	88mA	94mA
Average Power	240mW	326mW	351mW

Table 5.6: LoRa Power

The following Figure 5.22 represents the current results from the manufacturer, which gives the baseline for the above-mentioned tests. LoRa has different spreading factors - the configurations used in this test were the same used in the LoRa section of FCT- Laboratory, that were spreading factor 7, bandwidth 125 kHz, TX power 14 dBm, sample time 30 seconds, message size of 110 bytes plus 20 bytes when WiFi was used.

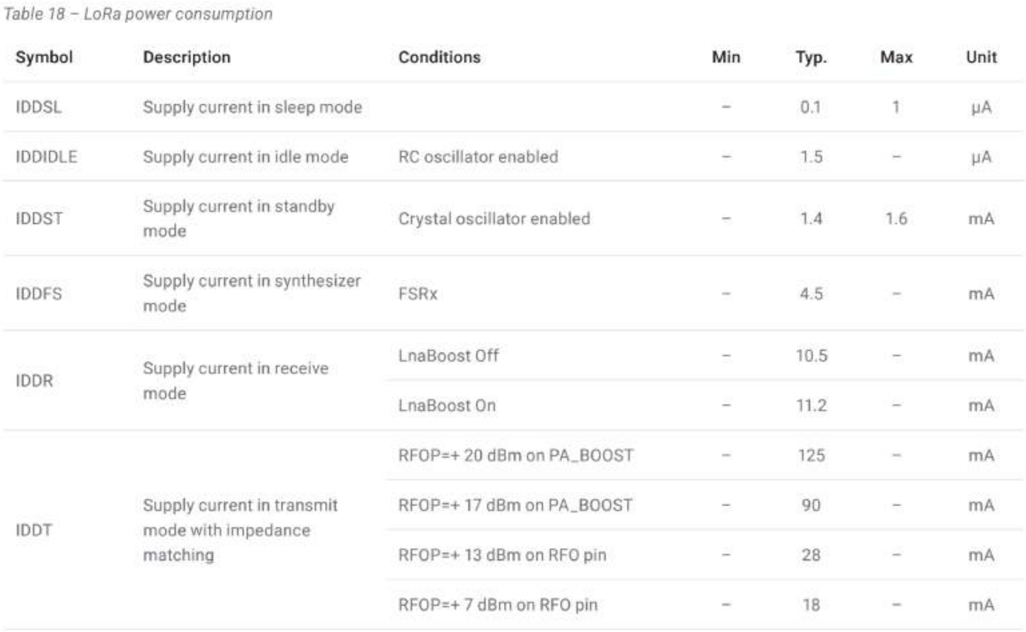


Table 18 – LoRa power consumption

Symbol	Description	Conditions	Min	Typ.	Max	Unit
IDDSL	Supply current in sleep mode		–	0.1	1	µA
IDDIDLE	Supply current in idle mode	RC oscillator enabled	–	1.5	–	µA
IDDST	Supply current in standby mode	Crystal oscillator enabled	–	1.4	1.6	mA
IDDFS	Supply current in synthesizer mode	FSRx	–	4.5	–	mA
IDDR	Supply current in receive mode	LnaBoost Off	–	10.5	–	mA
		LnaBoost On	–	11.2	–	mA
IDDT	Supply current in transmit mode with impedance matching	RFOP=+ 20 dBm on PA_BOOST	–	125	–	mA
		RFOP=+ 17 dBm on PA_BOOST	–	90	–	mA
		RFOP=+ 13 dBm on RFO pin	–	28	–	mA
		RFOP=+ 7 dBm on RFO pin	–	18	–	mA

Figure 5.22: LoRa Power Table [49]

5.3.2.2 NB-IoT

The next communication method used for this test was NB-IoT. The procedure was the same as above, starting by using only communication and then increasing the battery usage with GNSS and accelerometer, ending with all of this plus the scan of WiFi APs. The NB-IoT is expected to have poor performance compared to LoRa, because of the message size of 245 bytes plus 134 bytes if WiFi data is used, which is 2.22 to 2.71 times higher than LoRa. The results for this method are described in the following Table 5.7.

	Communication Only*	Communication GNSS, Sensor	Communication GNSS, Sensor, Scan WiFi
Battery Duration	5h 25m	4h 30m	4h
Average Current	148mA	185mA	208mA
Average Power	548mW	685mW	770mW

* Test not done with final code.

Table 5.7: NB-IoT Power

5.3.2.3 WiFi

The last communication used for these tests was WiFi, this method is expected to have the worst battery performance, but at the same time is the one with higher transfer speeds, meaning the airtime for the same amount of data is lower. So, under the right circumstances, it could perform better than the previous two, which was the objective of the test. WiFi could not be used as a viable options in the final code due to the short-range, and the need for authentication. Surprisingly, the results for the tests concluded that WiFi had better battery duration that NB-IoT. The reader should be informed that these tests were not done with the final code. They were only with hard-coded values, meaning they do not prove that WiFi is better than NB-IoT and will not be present in the next section for comparison. These values are expressed in the next table 5.8.

	Communication Only*	Communication GNSS, Sensor*	Communication GNSS, Sensor, Scan WiFi*
Battery Duration	5h 40m	5h 10m	4h 35m
Average Current	141mA	155mA	175mA
Average Power	522mW	574mW	648mW

* Test not done with final code.

Table 5.8: WiFi Power

5.3.2.4 Power Comparison

To close this analysis, a comparison with different communications methods can be observed in the next Table 5.9. This table combines all of the data from the previous ones, making it easier to compare the different methods. The conditions for the different tests should always be taken into consideration. As, expected LoRa had the best battery duration, followed by NB-IoT and WiFi had the worst battery duration.

	NB-IoT	LoRa
1- Communication only	5h 30m*	12h
2- Communication GPS, Sensor	4h 30m	9h
3-Communication GPS, Sensor, Scan WiFi	4h	8h 30m

* Test not done with final code.

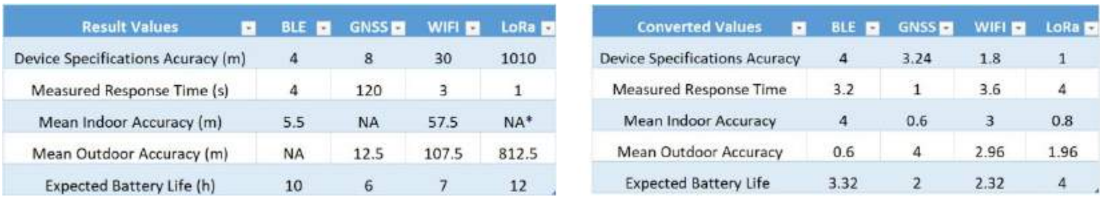
Table 5.9: Battery Duration

5.4 Results Comparison

In this section, the objective is to clarify and compare the different geolocation techniques from the three use cases.

The first row of Table 5.23a is the Device Specifications Accuracy measured in meters. The values in this row represent the radius of circle that has its center in a certain location point. These were calculated using Table 5.1, it was done the mean of these values using $(MaxAccuracy + MinAccuracy) \div 2$. The second row is the Measured Response Time, these values were represented in Tables 5.2, 5.3, 5.4, 5.5. This field is measured in seconds and it is the time needed to retrieve one location point for each method. The GNSS value is the time after a cold start (which is the worst value), after this cold start the time is similar to the others.

The next two methods, represented in the same table, are the Mean Indoor and Outdoor Accuracy, the same mathematical expression was used and the values are retrieved from the same tables as the Measured Response time, plus the information from subsection 5.2.2. In this two rows, is possible to observe 3 fields with NA, this value stands for "Not Available". The LoRa value with NA has an annotation since the test failed due to the minimum accuracy of LoRa, but LoRa can work indoor as long as the gateways are located far enough apart. The last row is the Expected Battery life, for this row it was used the measured values of Table 5.9 for LoRa, the other three values are expected values.



Result Values	BLE	GNSS	WIFI	LoRa
Device Specifications Acuracy (m)	4	8	30	1010
Measured Response Time (s)	4	120	3	1
Mean Indoor Accuracy (m)	5.5	NA	57.5	NA*
Mean Outdoor Accuracy (m)	NA	12.5	107.5	812.5
Expected Battery Life (h)	10	6	7	12

Converted Values	BLE	GNSS	WIFI	LoRa
Device Specifications Acuracy	4	3.24	1.8	1
Measured Response Time	3.2	1	3.6	4
Mean Indoor Accuracy	4	0.6	3	0.8
Mean Outdoor Accuracy	0.6	4	2.96	1.96
Expected Battery Life	3.32	2	2.32	4

(a) Table Result Values

(b) Table Converted Values

Figure 5.23: Geolocation Results Tables

Before the classification of the obtained values from Table 5.23a in the different levels, due to the difference between values, in some cases higher than 100×, it was defined a marking for each row. The considered levels are the following, 1 ⇒ Poor, 2 ⇒ Fair, 3 ⇒ Good and 4 ⇒ Excellent.

In the first row, the values from 4 to 10 were considered Excellent, from 10 to 20 Good, from 20 to 100 Fair and > 100 Poor. These decision is justified by the use case of this project, where in the location of person a value of 4 in case of BLE is Excellent, but a value of 1010 for LoRa is only Poor.

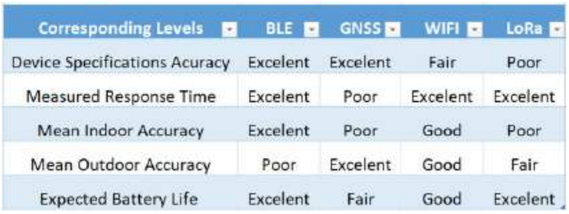
In the second row the considered intervals were < 5 Excellent, 5 to 10 Good, 10 to 30 Fair and the last one > 30 was considered Poor. These values were defined this way due to the minimum interval between communications (the minimum interval for this thesis was 30 seconds).

In the the mean indoor, the Excellent level was considered < 6, the Good level from 6 to 60, the Fair level from 60 to 100 and the last level > 100. The reason behind these values is the based on how well suited is the value for Indoor Location.

In the mean outdoor, the levels were defined < 13, 13 to 100, 100 to 300 and > 300 for the corresponding values Excellent, Good, Fair and Poor. The reason behind these values is how useful they can be for locating a person outdoor.

The last row differs from the previous, since the best value is the highest value. The Excellent value was considered from > 12 to 9, the third level from 9 to 6, the second level from 6 to 3, and the first level < 3 hours. These value were defined based on daily usage, these expected values are for a sampling time of 30 seconds, so they represent the worst case scenario for each geolocation method.

In the Figure 5.23b, are represented the converted values from Table 5.23a. In the end, this values were translated to the corresponding levels in Table 5.24. One example of this translation is the GNSS value of 8 m in Table 5.23a, which was converted for 3.24 in Table 5.23b and is equivalent to Excellent in the last Table 5.24.



Corresponding Levels	BLE	GNSS	WIFI	LoRa
Device Specifications Acuracy	Excelent	Excelent	Fair	Poor
Measured Response Time	Excelent	Poor	Excelent	Excelent
Mean Indoor Accuracy	Excelent	Poor	Good	Poor
Mean Outdoor Accuracy	Poor	Excelent	Good	Fair
Expected Battery Life	Excelent	Fair	Good	Excelent

Figure 5.24: Geolocation Results Corresponding Levels

The next Figure 5.25 uses a radar diagram to provide a visual and easier interpretation of the results from the Converted Values table. It is used for better comparing the different geolocation techniques with each other.

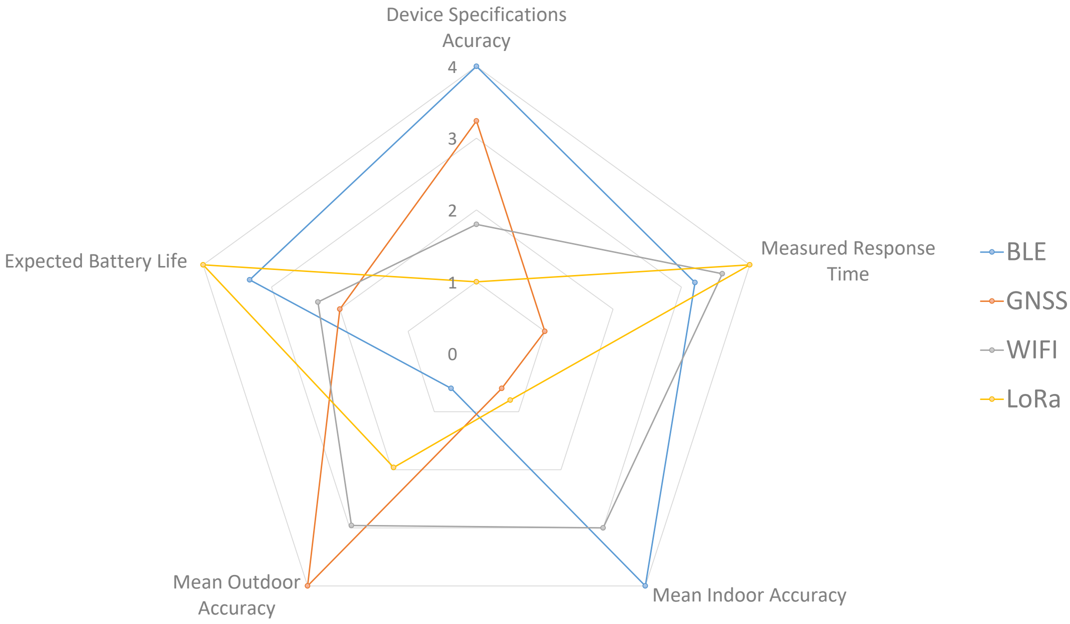


Figure 5.25: Geolocation Results Comparison

From the previous figure, the next graphic 5.26 was done. In this graphic is possible to observe the best technique to use in every different situation. It is possible to observe that there is not a technique better than the others in every situation. Next to the red arrow represented in the graphic is explained the situation where WIFI is the best suited for a mixed indoor/outdoor accuracy, although it is not the best neither Indoor (BLE), nor Outdoor (GNSS). With a closer observation of this graphic, it can be concluded that the best solution is a profile-based decision system, capable of choosing at any moment the best technique to use.

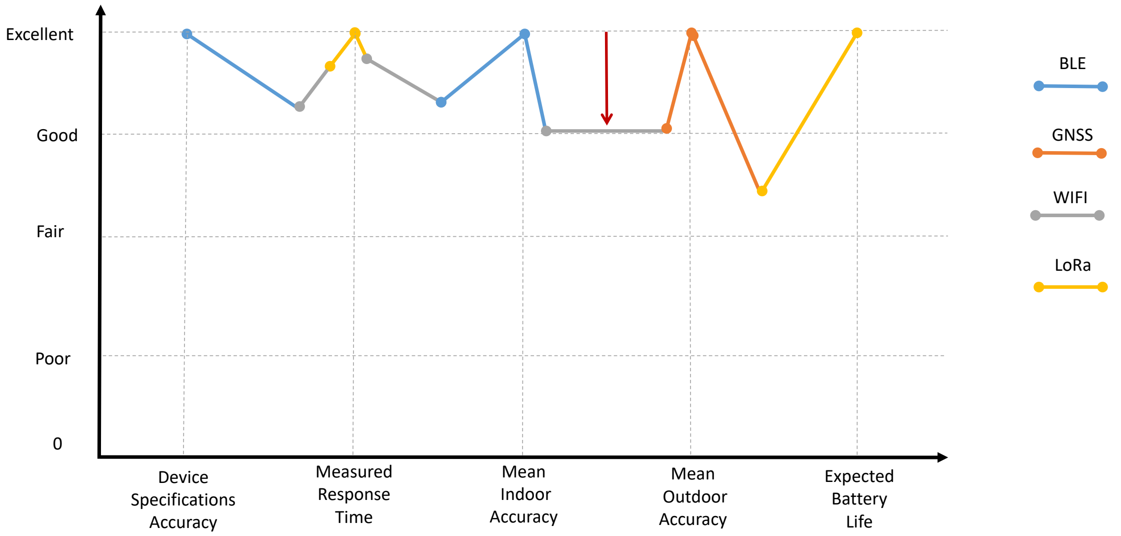


Figure 5.26: Geolocation Results Comparison line

In the following list will be described one example of use for every geolocation technique. This list is done based on the information from the previous figures, and could serve as example of usage for the profile-decision system.

- BLE
  For BLE the optimal scenario will be an indoor short-range location, for example, to know if a person is inside a room, or, in a building, to know in which floor the person is currently at.
- GNSS
  The GNSS is the classic approach for solving location problems and is the best fitted for outdoor accuracy.
- WiFi
  The best scenario for this assisted location will be urban outdoor, where the accuracy for the results can be close to GNSS.
- LoRa
  The optimal scenario for the analyse of the LoRa Metadata is outdoor, but can also work indoor. The best fit for this technique is when the remaining battery is to low.

5.5 Discussion

One initial objective was to determine what kind of impact had LPWAN enabled solutions in wearable devices, in particular on those focused on people with dementia.
It was hypothesized that people dealing with dementia would benefit from LPWAN enabled solutions, since they would help in perform the location of the person, while saving battery, reducing the overall cost, and comply with the technical constraints of wearable devices. In addition, they could also bear the possibility to dynamically choose the best location available in that moment.

The results accuracy matched those mentioned in previous studies of geolocation technologies and techniques and that are represented in Related Works.
There are several possible explanations for the results, since the used hardware, to the environmental conditions (for example, if the tests were taken outdoor in an urban area or in a rural location).

With these results, in the system was possible to observe that, with the change of communication and the type of received message, the system was capable of returning the same output, and perform the adaptive geolocation.
These results provide support for the hypothesis that LPWAN can be used for determining the location of people with dementia, while accounting for the constraints of wearable IoT devices.

CHAPTER 6

CONCLUSIONS

In this last chapter, the ultimate conclusions will be drawn from this work and a retrospective will be made on this thesis, concluding with the proposed future work.

In this thesis, the author presented the work needed to create an adaptive geolocation model, composed by three functional stages (“Hierarchical”, “Smart”, “Advanced”) that were presented in Chapter 3 and are dedicated for wearable IoT devices. The main aim was to prove that Adaptive Geolocation of IoT devices can provide a viable approach to the location of these devices, especially those dedicated to people with dementia.

The implementation of this model was integrated into the Carelink [6] platform and is capable of dynamically choose the best geolocation technology in each situation. To achieve this, different operation modes were used that, ultimately, made possible to improve accuracy and energy consumption. In this way, the proposed hypothesis was used to solve the research question.

Initial location tests were done to evaluate the accuracy results of different geolocation techniques. The first technique with better average accuracy, was GNSS, with 10 meters. The second test was for Wi-Fi assisted location, this method scans the radio environment looking for Wi-Fi access points and, based on the know location of such access points returns the approximate location for the device (here, the average result was 30 meters). The last one, with the similar working principle as Wi-Fi, was LoRa, with the average accuracy of 300 meters.

The results for the tests were performed using as a microcontroller, the FiPy [50] with the Pytrack [66] localization shield. This shield has a GNSS, that is the Quectel L76-L [67], and the antenna used for the GNSS was the internal one. Different hardware configurations would have shown different results.

The system was capable of hierarchically choosing the best location and responding to information from different communication methods and data sources, concluding the implementation of the “Hierarchical” functional stage.

This initial implementation can be checked in the research paper done for the International

Conference on Information Society and Technology (ICIST), with the help of professor João Sarraipa and the Researcher Jorge Calado, found in A and also at [69]. This paper was accepted and will be published in ICIST online repository and in the book of proceedings of this conference.

The tests for the implementation of the "advanced" stage were undertaken, testing the battery consumption and communication resilience. This tests occurred in a laboratory scenario, and this implementation stage was not validated within real life trials, being this last step needed to be considered completed.

The results for the "Advanced" stage, identified by the author, prove a battery duration of up to 12 hours when using the LoRa assisted location method, with a 30 seconds period between transmissions. For the communication resilience, was concluded that the device was capable of dynamically change the communication method and the system was able to handle the data transition. The system was also capable of forcing the transition, based on the remaining battery of the device.

The identified drawbacks for the adaptive geolocation model, were the maximum number of messages processed per second, and the fact that the Wi-Fi and LoRa locations were depended from third-party providers.

The main use case of this model, was the Carelink project, where the model was utilised for knowing the location of wearable devices, used by people who suffer from dementia, as it was tested in real-life trials.

Finally, by analysing the developed system (composed by the model and the device) and its results, it is possible to verify that the approach is promising and the defined model has provided an appropriate starting point for further research and in the field of assisted living location.

6.1 Future Work

Future work should focus on performing more tests, to discover faults or bottlenecks related to the capacity of the system to process received messages. For the device's future tests, it is required to assess the location performance and evaluate power consumption of the geolocation technologies, both GPS, GPS-free, and assisted location in indoor environments.

Further studies of the solution should be conducted to implement the "Smart" stage proposed in the Chapter 3. This functional stage comprises the development and testing of more operation modes, to have a profile based decision, taking into account variables such as if the PwD is at home or outside, if the person is accompanied or is alone and the current time of day. In the end, this stage should be validated with actual people in real-life trials.

Bibliography

[1] *PORDATA - Indicadores de envelhecimento*. URL: https://www.pordata.pt/Portugal/Indicadores+de+envelhecimento-526 (visited on 12/16/2019).

[2] *PORDATA - Índice de envelhecimento*. URL: https://www.pordata.pt/Europa/{\'{I}}ndice+de+envelhecimento-1609 (visited on 12/16/2019).

[3] A. Hammoud, M. Deriaz, and D. Konstantas. "Wandering Behaviors Detection for Dementia Patients: A Survey." In: *2018 3rd International Conference on Smart and Sustainable Technologies, SpliTech 2018* (2018).

[4] "Care Needed: Improving the Lives of People with Dementia." In: (2018). DOI: 10.1787/9789264085107-en. URL: https://doi.org/10.1787/9789264085107-en.

[5] OECD. *Health at a Glance 2019*. Health at a Glance. OECD, Nov. 2019. ISBN: 9789264942752. DOI: 10.1787/4dd50c09-en. URL: https://www.oecd-ilibrary.org/social-issues-migration-health/health-at-a-glance-2019{_}4dd50c09-en.

[6] *Carelink Homepage - Carelink*. URL: http://carelink-aal.org/ (visited on 12/16/2019).

[7] T. Hadwen, V. Smallbon, Q. Zhang, and M. D'Souza. "Energy efficient LoRa GPS tracker for dementia patients." In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2017), pp. 771–774. ISSN: 1557170X. DOI: 10.1109/EMBC.2017.8036938.

[8] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso. "Low-power wide area network technologies for internet-of-things: A comparative review." In: *IEEE Internet of Things Journal* 6.2 (Apr. 2019), pp. 2225–2240. ISSN: 23274662. DOI: 10.1109/JIOT.2018.2883728.

[9] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer. "A comparative study of LPWAN technologies for large-scale IoT deployment." In: *ICT Express* 5.1 (Mar. 2019), pp. 1–7. ISSN: 24059595. DOI: 10.1016/j.icte.2017.12.005. URL: http://www.sciencedirect.com/science/article/pii/S2405959517302953.

[10] *A Comprehensive Look At LPWAN For IoT Engineers & Decision Makers*. Tech. rep. URL: https://www.link-labs.com/lpwan (visited on 10/15/2019).

[11] A. Zourmand, A. L. Kun Hing, C. Wai Hung, and M. Abdulrehman. "Internet of Things (IoT) using LoRa technology." In: *2019 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2019 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., June 2019, pp. 324–330. ISBN: 9781728107844. DOI: `10.1109/I2CACIS.2019.8825008`.

[12] *A technical overview of LoRa ® and LoRaWAN What is it?* Tech. rep. 2015.

[13] LoRa Alliance. "LoRaWAN 1.0.3 specification." In: *Lora-Alliance.Org* 1 [Online], Accessible: https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf (2018), pp. 1 –72. URL: `https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf`.

[14] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler. "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things." In: *IEEE Communications Magazine* 56.2 (Feb. 2018), pp. 60–67. ISSN: 01636804. DOI: `10.1109/MCOM.2018.1700625`.

[15] R. S. Sinha, Y. Wei, and S. H. Hwang. "A survey on LPWA technology: LoRa and NB-IoT." In: *ICT Express* 3.1 (Mar. 2017), pp. 14–21. ISSN: 24059595. DOI: `10.1016/j.icte.2017.03.004`.

[16] J. Pacheco. "Performance Evaluation of Class A LoRa Communications." Master's thesis. 2019.

[17] *Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)*. URL: `https://www.sigfox.com/en` (visited on 12/23/2019).

[18] *Sigfox Technical Overview*. Tech. rep. 2017.

[19] M. Anteur, V. Deslandes, N. Thomas, and A.-L. Beylot. "Ultra Narrow Band Technique for Low Power Wide Area Communications." In: Institute of Electrical and Electronics Engineers (IEEE), Mar. 2016, pp. 1–6. DOI: `10.1109/glocom.2015.7417420`. URL: `https://ieeexplore.ieee.org/document/7417420` (visited on 12/26/2019).

[20] J. Gozalvez. "New 3GPP Standard for IoT [Mobile Radio]." In: *IEEE Vehicular Technology Magazine* 11.1 (2016), pp. 14–20. ISSN: 15566072. DOI: `10.1109/MVT.2015.2512358`.

[21] *NB-IoT: a sustainable technology for connecting billions of devices - Ericsson Technology Review - Ericsson*. URL: `https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/nb-iot-a-sustainable-technology-for-connecting-billions-of-devices` (visited on 01/06/2020).

[22] M. Chen, Y. Miao, Y. Hao, and K. Hwang. "Narrow Band Internet of Things." In: *IEEE Access* 5 (2017), pp. 20557–20577. ISSN: 21693536. DOI: `10.1109/ACCESS.2017.2751586`.

[23] *LTE-M DEPLOYMENT GUIDE to Basic Feature set Requirements*. Tech. rep. 2018. URL: www.gsma.com/IoT.

[24] Y. Hwang. *Cellular IoT Explained - NB-IoT vs. LTE-M vs. 5G*. 2018. URL: https://medium.com/iotforall/cellular-iot-explained-nb-iot-vs-lte-m-vs-5g-and-more-8f26496df5d4https://www.iotforall.com/cellular-iot-explained-nb-iot-vs-lte-m/ (visited on 12/16/2019).

[25] *GSMA| Deployment Map| Internet of Things*. URL: https://www.gsma.com/iot/deployment-map/#deployments (visited on 01/14/2020).

[26] W. Webb. *Understanding Weightless: Technology, Equipment, and Network Deployment for M2M Communications in White Space*. Understanding Weightless. Cambridge University Press, 2012. ISBN: 9781107027077. URL: https://books.google.pt/books?id=eOJ4fZYG2WAC.

[27] U. Raza, P. Kulkarni, and M. Sooriyabandara. "Low Power Wide Area Networks: An Overview." In: *IEEE Communications Surveys and Tutorials* 19.2 (Apr. 2017), pp. 855–873. ISSN: 1553877X. DOI: 10.1109/COMST.2017.2652320. arXiv: 1606.07360.

[28] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver. "IEEE 802.11AH: The WiFi approach for M2M communications." In: *IEEE Wireless Communications*. Vol. 21. 6. Institute of Electrical and Electronics Engineers Inc., Dec. 2014, pp. 144–152. DOI: 10.1109/MWC.2014.7000982. arXiv: 1402.4675.

[29] *How RPMA Works - Ingenu*. URL: https://www.ingenu.com/technology/rpma/how-rpma-works/ (visited on 01/21/2020).

[30] T.Figueiredo. "LoRaWAN Performance evaluation." Master's thesis. 2019.

[31] J. Peña Queralta, T. N. Gia, Z. Zou, H. Tenhunen, and T. Westerlund. "Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: Beyond Lora and Lorawan." In: *Procedia Computer Science*. Vol. 155. Elsevier B.V., 2019, pp. 343–350. DOI: 10.1016/j.procs.2019.08.049.

[32] *MIOTY - The Wireless IoT Platform*. URL: https://www.iis.fraunhofer.de/en/ff/lv/net/telemetrie.html (visited on 01/20/2020).

[33] J. Danebjer. "A Hybrid Approach to GPS-Free Geolocation over LoRa." Master's thesis. 2018.

[34] K. J. Markoulidakis C. DesiniotisK. "Method for improving the CGI++ mobile location technique by exploiting past measurements." In: (Jan. 2005). URL: https://www.researchgate.net/publication/228887009_Method_for_improving_the_CGI_mobile_location_technique_by_exploiting_past_measurements.

[35] F. Thomas and L. Ros. "Revisiting Trilateration for Robot Localization." In: *IEEE TRANSACTIONS ON ROBOTICS* 21.1 (2005). DOI: 10.1109/TRO.2004.833793.

[36] B. C. Fargas and M. N. Petersen. "GPS-free geolocation using LoRa in low-power WANs." In: *GIoTS 2017 - Global Internet of Things Summit, Proceedings*. Institute of Electrical and Electronics Engineers Inc., Aug. 2017. ISBN: 9781509058730. DOI: `10.1109/GIOTS.2017.8016251`.

[37] *Mlat Service Area*. URL: `https://commons.wikimedia.org/wiki/File:2D_Mlat_Service_Area.jpg` (visited on 03/09/2020).

[38] N. Ivanov and V. Salischev. "The glonass System An Overview." In: *Journal of Navigation* 45.2 (1992), pp. 175–182. ISSN: 14697785. DOI: `10.1017/S0373463300010675`.

[39] *Galileo is the European global satellite-based navigation system | European Global Navigation Satellite Systems Agency*. URL: `https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system` (visited on 01/06/2020).

[40] F. Briilwdis, T. Friedman, M. D. De Amorim, and S. Fdida. "GPS-free-free positioning system for wireless sensor networks." In: *2005 International Conference on Wireless and Optical Communications Networks*. 2005, pp. 541–545. ISBN: 0780390199. DOI: `10.1109/wocn.2005.1436085`.

[41] G. Zhu, Q. Li, P. Quan, and J. Ye. "A GPS-free localization scheme for wireless sensor networks." In: *International Conference on Communication Technology Proceedings, ICCT*. 2010, pp. 401–404. ISBN: 9781424468690. DOI: `10.1109/ICCT.2010.5688823`.

[42] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn. "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas." In: *Data* 3.2 (2018), pp. 1–15. ISSN: 23065729. DOI: `10.3390/data3020013`.

[43] N. Podevijn, D. Plets, J. Trogh, L. Martens, P. Suanet, K. Hendrikse, and W. Joseph. "TDoA-Based Outdoor Positioning with Tracking Algorithm in a Public LoRa Network." In: *Wireless Communications and Mobile Computing* 2018 (2018). ISSN: 15308677. DOI: `10.1155/2018/1864209`.

[44] LoRa-Alliance. "Geolocation Whitepaper." In: January (2018), p. 15.

[45] S. Hu, A. Berg, X. Li, and F. Rusek. "Improving the Performance of OTDOA based Positioning in NB-IoT Systems." In: *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings* 2018-January (Apr. 2017), pp. 1–7. arXiv: `1704.05350`. URL: `http://arxiv.org/abs/1704.05350`.

[46] *MicroPython & #WiPy 2.0/3.0: Geolocation using WLAN - LeMaRiva|tech*. URL: `https://lemariva.com/blog/2017/11/micropython-wipy2-0-geolocalization-using-wlan` (visited on 01/05/2020).

[47] M. D'souza, M. Ros, and M. Karunanithi. "An indoor localisation and motion monitoring system to determine behavioural activity in dementia afflicted patients in aged care." In: *Electronic Journal of Health Informatics* 7.2 (2012), p. 14. ISSN: 14464381. URL: `www.eJHI.net`.

[48] N. E. Tabbakha, W. H. Tan, and C. P. Ooi. "Indoor location and motion tracking system for elderly assisted living home." In: *Proceeding of 2017 International Conference on Robotics, Automation and Sciences, ICORAS 2017* 2018-March.September 2019 (2018), pp. 1–4. DOI: `10.1109/ICORAS.2017.8308073`.

[49] Pycom. *FiPy datasheet*. Tech. rep. 2017. URL: `https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_FiPy_v2.pdf`.

[50] *FiPy - Pycom - Five Network Development Board with LTE-M, LoRa, Sigfox, WiFi and Bluetooth*. URL: `https://pycom.io/product/fipy/` (visited on 04/29/2020).

[51] *Pytrack - Pycom*. URL: `https://pycom.io/product/pytrack/` (visited on 04/29/2020).

[52] *jms-calado/Pycom*. URL: `https://github.com/jms-calado/Pycom` (visited on 06/09/2020).

[53] *The Things Network*. URL: `https://www.thethingsnetwork.org/` (visited on 10/20/2019).

[54] *LORIOT - The LoRaWAN® Network Server Provider*. URL: `https://loriot.io/` (visited on 11/08/2019).

[55] *IoT smart metering solutions and smart meter data visualization with ThingsBoard | ThingsBoard*. URL: `https://thingsboard.io/smart-metering/` (visited on 11/08/2019).

[56] *Introduction - Mbed OS 5 | Mbed OS 5 Documentation*. URL: `https://os.mbed.com/docs/mbed-os/v5.15/introduction/index.html` (visited on 11/09/2019).

[57] *About — Mozilla IoT*. URL: `https://iot.mozilla.org/about/` (visited on 11/09/2019).

[58] *Node-RED*. URL: `https://nodered.org/` (visited on 11/14/2019).

[59] *rafaelrodriguesfct/Thesis*. URL: `https://github.com/rafaelrodriguesfct/Thesis` (visited on 06/15/2020).

[60] *jms-calado/MQTT-Specs*. URL: `https://github.com/jms-calado/MQTT-Specs` (visited on 06/09/2020).

[61] *Specifications | Wi-Fi Alliance*. URL: `https://www.wi-fi.org/discover-wi-fi/specifications` (visited on 06/02/2020).

[62] *Developer Guide | Geolocation API | Google Developers*. URL: `https://developers.google.com/maps/documentation/geolocation/intro` (visited on 11/05/2019).

[63] *Build apps with HERE Maps API and SDK Platform Access | HERE Developer*. URL: `https://developer.here.com/` (visited on 11/08/2019).

[64] *OpenCelliD - Largest Open Database of Cell Towers & Geolocation - by Unwired Labs*. URL: `https://opencellid.org/` (visited on 11/06/2019).

[65] *Semtech LoRa | CLOUD Docs*. URL: `https://www.loracloud.com/documentation/geolocation` (visited on 11/16/2019).

[66] *Pytrack Specsheet*. Tech. rep. URL: https://docs.pycom.io/gitbook/assets/pytrack-specsheet-1.pdf (visited on 05/13/2020).

[67] *Quectel GNSS L76-L Specification*. Tech. rep. URL: https://www.quectel.com/UploadFile/Product/Quectel_L76-L_GNSS_Specification_V1.2.pdf (visited on 05/13/2020).

[68] *Cayenne Features -Developer*. URL: https://developers.mydevices.com/cayenne/features/ (visited on 05/21/2020).

[69] R. Rodrigues, J. Calado, J. Sarraipa, and R. Jardim-Goncalves. *Geolocation Solver of IoT Devices for Active and Assisted Living*. Ed. by Milan Zdravković, Zora Konjović, Miroslav Trajanović. Vol.1. Society for Information Systems and Computer Networks, 2020, pp. 165–169. URL: http://www.eventiotic.com/eventiotic/library/paper/607.

APPENDIX

A

PAPER: GEOLOCATION SOLVER OF IOT DEVICES FOR ACTIVE AND ASSISTED LIVING

Geolocation Solver of IoT Devices for Active and Assisted Living

Rafael Rodrigues[*], Jorge Calado[*], João Sarraipa[*], Ricardo Jardim-Gonçalves[*]
[*] CTS, UNINOVA, DEE, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa,
2829-516 Caparica, Portugal
ror@uninova.pt, jsc@uninova.pt, jfss@uninova.pt, rg@uninova.pt

***Abstract*— Latest enhancements in IoT devices and in communication technologies, has brought new ideas that are capable of providing advanced sensing of the surrounding environment. On the other hand, average life expectancy has grown, resulting in a considerable increase in the number of elderly people. Consequently, there is a constant search for new solutions, to support an Active and Assisted Living (AAL) of these people. This paper aims to propose a solution to help in knowing the location of IoT devices that could be helping these people. The proposed solution takes into consideration the risk factors of the target persons, at any given time and as well as the technical constraints of the device, such as available power and communications. Thus, a profile-based decision is taken autonomously either by the device or its integrated system to ensure the use of the best geolocation technology in each situation.**

Keywords: Internet of Things, Low Power Wide Area Networks, Geolocation, Active and Assisted Living

I. INTRODUCTION

By 2050 the amount of people with dementia will be tripled to 132 million, with societal economic costs accounting for 1% of global GDP [1]. Dementia is characterized by progressive loss of memory, as well as other mental faculties including language, judgment, planning, social interaction, and leads to serious problems in coping with activities of daily living, including orientation and wayfinding simple tasks. One of the most common forms of disruption, for people with this health status is wandering. According to the Alzheimer's Association, 6 in 10 people with dementia will wander [2]. This problem induces a great risk to the safety, well-being and reduces drastically the quality of life of the person, therefore is a critical concern for caregivers, and family having a major impact on their lives. Even though there is still no cure, efforts can be made to help prevent such behaviours, that's why there is a continuous search for solutions to support an Active and Assisted Living of such people, not only the ones with dementia but more extensive to all the elderly population.

Alongside with a fast-growing in the elderly populations across the world, caused by the augmented life expectancy, there is also an expected increasing number of Internet of Things (IoT) devices, these devices can be key components to mitigate some of the problems caused by this ageing. These devices have special requirements and technical constraints, such as low power consumption or low-cost hardware, but they can provide valuable sensor data over long distances, and the ability to retrieve location data, which is especially useful for the Carelink [3] project. This project consists of an innovative personal tracking for people with dementia (PwD), where a wearable device is currently being developed, alongside with an online platform, where it will be possible to observe the actual position of the PwD as well as define safe and unsafe areas. When the PwD crosses outside a safe geo-fence zone, the carer will receive an SMS notification, with the information to locate the PwD, in case of emergency.

With the use of these devices, it is possible to detect, try to predict and prevent risk behaviours, such as falls or wandering events. Additionally, with the ability to geolocate the PwDs, there is the opportunity to discover wandering patterns, helping to prevent such events in the future, thus ensuring a better quality of life for the PwD and for those responsible for them.

II. RELATED WORK

Most studies in the area of geolocation have focused on the use of a single technology, or technique. Then there are the ones that uses multiple technologies and techniques, but seems that do not explore so much the dynamic change of the working technology in functioning mode.

The following four examples, focus only in different individual technologies and techniques. The last two focus either in multiple technologies, or multiple techniques, but as already emphasized, without taking into consideration the use of multiple techniques and technologies simultaneously.

In [4] an IoT tracking system is presented, using LoRa[5] where the geolocation is calculated through a multilateration algorithm, using the gateways timestamp, with an accuracy of around 100m in a stationary test scenario.

In [6] the performance and accuracy of the Google Geolocation API [7], for geolocation using Wi-Fi [8] access points(APs) is evaluated, the results in an urban environment, achieving a maximum accuracy of 20 meters, minimal 187 meters and median 39 meters, for this test a minimum of three Wi-Fi APs is recommended.

In [9], an indoor localization monitoring system is presented and a wearable device was developed, using FleckTM-3 wireless sensor platform, with a position error of 1 to 3.5 meters. The main disadvantage of this solution is that only works indoors, and the technology is similar to ZigBee so only works for short-range applications.

In [10], was developed an indoor position system based on Raspberry Pi and an MPU6050, used has BLE beacon, and with the use of Raspberry Pi as BLE scanner, measuring the RSSI, the results for indoor activity were 99% of accuracy in knowing in which division the patience was.

In [11] a dataset of messages was created from LoRa and Sigfox containing the GPS coordinates, and respective RSSI (Received Signal Strength Indication). The results of the median error in an urban scenario were 514.83 meters for Sigfox and 273.03 meters for LoRa.

In [12], a Hybrid (Time of Flight and RSSI) approach for Geolocation system using LoRa, and the results are similar to the work mentioned in [11], with a median error of 272 meters.

III. RESEARCH QUESTION

One of the objectives of this work is to evaluate the use of technologies as LoRa, Wi-Fi, Global Navigation Satellite System (GNSS) integrated with other sensor data, available from the localization devices such as the remaining battery level and accelerometer parameters. From this the idea is to define different operation modes in relation to different energy consumptions with the objective to improve the overall energy consumption of the device.

The selection of each operation mode is dependent on the following conditions: for LoRa, the number of gateways available; for GNSS, the number of satellites available to fix the location; and for Wi-Fi, the minimum amount of APs in range, to perform the assisted location.

Can the geolocation technologies, dependent on the usage scenario, be managed dynamically to improve the precision of the location, and the energy consumptions of the wearable devices?

IV. METHODOLOGY

To respond to the presented research question, a Geolocation model able to autonomously decide the best location method is proposed. To reach a model able of the mentioned autonomous feature, it must address distinct variables as remaining battery and the availability of communication technologies. Such availability can be analysed through advanced sensing of the surrounding radio signals. This means that the model has to be aware of the environment in which the device is working. Such environment main characteristics relates to indoor or outdoor and rural or urban.

These characteristics have a direct impact in what kind of technology has to be used in each working moment or environment. Thus, the idea is to define a model that responds to all of these situations, and consequently develop a correspondent profile-based decision system able to act accordingly.

The development of the presented solution in this paper focuses only in the urban and rural environment. Thus its testing used geolocation technologies such as GNSS, Wi-Fi and LoRa. Furthermore, these tests will consist in the gathering information from different location coordinates, at different speeds as well as stationary, to evaluate the performance of the different technologies against each other.

Additionally, these tests were integrated in the Carelink project presented in the introduction. It used a geo-fencing polygon, defined by the carer of the PwD. Consequently, the device communicates with the Carelink platform to give its position and receive support on the technologies to use accordingly to the profiles of the PwD user. Thus, at the end, these tests were conducted with real patients in real life trials. To perform these communication between the devices and the Carelink platform it was used LoRa and NB-IoT [13].

V. THE PROPOSED MODEL AND SYSTEM

To develop the solution described, it was defined a model to characterize and formalize the different types of solutions called stages . Later an implementation following the model defined occurred creating the so-called profile-based decision system.

The model intends to define what technologies to use and how the geolocation data from different sources should be processed, so that it can later be used for knowing the actual position of the PwD wearing the device. This model is composed by three different functional stages, called: “hierarchical”, “advanced” and “smart”.

In the first stage, the function relates to the decision of the best location technology using their own approximate accuracy values (specifications), meaning an “hierarchical” choice.

The second stage formalises the location decision function based in two characteristics at the same time: the battery level of the device and the location accuracy for the chosen technology, thus being more “advanced”. In this stage, the geolocation function should be capable of knowing the remaining battery. In case the person is in a dangerous situation, for example, in an area previously assigned as unsafe, do the balance between the highest accuracy location and the more power efficient location.

In the last stage, a more advanced location function is formalised. Its decision capability for choosing the available technologies are based in the surrounding environment, thus including other types of data as physiological or even weather data. For example, if a PwD is not showing any activity by a long period of time, it is night time and he/she is in a safe place, as home it can be deducted that the person is sleeping, and therefore the sampling time for the location can be reduced, thus saving battery.

Another situation is in case of by combining additional sensor data, like an accelerometer that can detect if the person has fallen, priority can be given to the method with the best precision, because of this dangerous situation.

In conclusion, this last stage combine a set of different factors to categorize and do a profile-based decision. This relates to a previous work reported in [14], which addresses the problem of creating adaptable power profiles for wearable localization devices also establishing a relation to different levels of dementia of the PwD users.

Based on the presented model the authors started to develop the profile-based decision system. In this phase it only includes the “hierarchical” functional stage. Its first architecture version is illustrated in Figure 1. Although some tests related to the second stage were performed and are also described later in this paper.

The first step in the proposed system is to collect sensor data from the PwD wearing the device. Afterwards, a specific transmission method is selected, of which the options may vary from LoRa to NB-IoT.

The second step is to combine the previous information in the proposed Geolocation solver, in order to get the best location possible. The aforementioned Carelink platform is responsible for managing the device, and ensure the high availability needed, in order to always know the location of the PwD, especially when he/she is lost.

The geolocation information is then passed to the tracking service, and then the final step is sending the response from the tracking service to the GUI.

In the GUI, the user responsible for the PwD can be alerted, when a geo-fencing alert is raised.

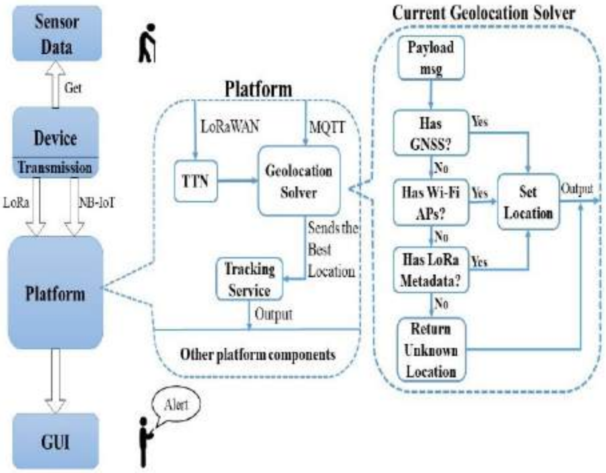


Figure 1. Profile-based decision System Architecture – Hierarchical version

The right dotted square represented in Figure 1 shows the workflow behind the “Geolocation Solver” block. An input is received containing a JSON object, which contains the status of the device, in the format shown in Figure 2.

```
{
    "timestamp": "YYYY-MM-DDThh:mm:ssZ",
    "location": {
        "lat": float,
        "lon": float,
        "alt": float,
        "hdop": float,
        "vdop": float,
        "pdop": float
    },
    "batteryLevel": integer,
    "accompanied": boolean,
    "sensor": {
        "accelerometer": {
            "x": float,
            "y": float,
            "z": float
        }
    },
    "wifiAPs": {
        "mac_1": string,
        "rssi_1": integer,
        "mac_2": string,
        "rssi_2": integer,
        "mac_3": string,
        "rssi_3": integer
    }
}
```

Figure 2. JSON Status payload

In the “Has GNSS?” block (Figure 1), the location field (Figure 2) is analysed, the coordinates provided are validated in case the latitude and longitude fields are different from zero, which is the default value for no GNSS location data, being then passed to the “Set Location” block. In case this field is empty, or the coordinates are not valid, the following block is “Has Wi-Fi APs?”.

In this step the Wi-Fi APs field, from the above-mentioned JSON is checked. If it is different from null, this payload is utilized, for performing the assisted location based on the Wi-Fi data. This data will then be sent to three different APIs, using a load balancer based in the sequential Round Robin to obtain the best information for the location. The round robin solution is used to give capability to the system to perform multiple location at the same time and chose the best one. The three used APIs to accomplish this were: Google Geolocation API [7], Here Position API [15] and OpenCelliD Cellular Geolocation API [16].

If the communication method in use is LoRa, then the LoRa metadata will also be used to apply a different geolocation algorithm, such as multilateration, based on received signal strength or the time of difference of arrival. The result is the approximate device location. For this last method to work, with reliability, a minimum of three LoRa gateways in range is needed.

The location data, passed by either one of the three previous blocks (“Has GNSS?”, “Has Wi-Fi APs”, “Has LoRa Metadata”) to the “Set Location” block, is then analysed according to the rules defined by the corresponding functional stage (“hierarchical”, “smart”, “advanced”).

The functional stage used was “hierarchical” that can perform a hierarchical location, based in the information received earlier. Being the hierarchical steps, GNSS, Wi-Fi and LoRa, and this operation is based on the average accuracy of location, for each method.

Therefore, the location with higher priority is the one from GNSS, and the one with less priority is the LoRa Metadata. In the scenario where all of the three previous blocks return an invalid location, it is returned location unknown. After this operation, the information is then sent to the “Tracking Service” on the platform.

VI. RESULTS

The work presented by the authors intends to combine different location technologies and techniques. The result is an overall location solution, that provides enhancements in terms of precision and performance, meeting the dynamic changes of the utilization scenarios with optimal device configurations. It also integrates energy profiles to control and reduce the energy consumptions of the devices, thus answering the proposed research question.

The tests executed consist in connecting the wearable device to the Carelink [3] platform, using the “hierarchical” geolocation function to provide the best location to the platform. The wearable device in use, was based in the FiPy [17] development board, and it was used in the form of belt box. Three tests were used to determine the location accuracy of the previous methods, obtaining the following results (see Table 1):

a) The location results for the GNSS, after using the device in a clear day, and perform a 4.3 km walk in a rural outdoor environment, at an average walking speed of 5 km/h, gathering data from 100 location points, had an accuracy between 5m to 20m, with an average of 10m.

b) The Wi-Fi assisted location test was done outdoor, in an urban environment, at an average walking speed of 5 km/h, with a distance of 2.8 km, collecting about 80 location points. The final accuracy for this method was a maximum of 15m, and a minimum of 200m, with an average of 30m.

c) The last method LoRa had the lowest accuracy, has expected by the authors, with a maximum of 30m and minimum of 3000m being the average 300m. The test occurred in an urban outdoor environment, at an average walking speed of 5 km/h, with a distance of 2 km, collecting around 50 location points.

As a result of this observation it can be concluded that in a specific case the Wi-Fi could be more precise than the GNSS. This means that sometimes the better expected technologies are not the best in some specific cases. This case relates to examples when people/users are entering in their homes where Wi-Fi sources become more precise than the GNSS. This can be verified in Table 1 (red values), where the min accuracy of GNSS is higher than the max accuracy of Wi-Fi.

Table I
Geolocation Results Comparison

	Max Accuracy	Min Accuracy	Average Accuracy	Location Points
GNSS	5 m	20 m	10 m	100
Wi-Fi	15 m	200 m	30 m	80
LoRa	30 m	3000 m	300 m	50

The authors, after the tests for the location accuracy of the Geolocation Solver system. Also performed tests of the communication resilience, and the battery consumption to prepare the next phase of the implementation, the "Advanced" functional stage.

The communication tests were conducted to the two different methods: LoRa and NB-IoT. In order to test the communication resilience, the communication fallback capabilities were evaluated. If the device loses the NB-IoT link it should fallback to using the LoRa stack. In case the communication network changes, then the Geolocation Solver service should be able to handle the data transition and continue the processing of information. Additionally, it should also be possible to force the change of the communication technology based on the energy efficiency settings.

The first test consisted in using the device outdoor, with good NB-IoT signal strength, and then enter an underground parking lot, which was previously tested and had no NB-IoT coverage. A LoRa gateway was set inside the parking lot, in order to provide LoRa coverage. The test was conducted 5 times, and the device always changed to LoRa after a predefined connection time-out of the NB-IoT.

With this expected result, in the system side was possible to observe that with the change of communication, and the type of received message the system was capable of returning the same output.

The second test consisted in sending a downlink message in a JSON format, to the device with the communication technology and a boolean value, such as: "{"ltenb": "False", "lora": "True"}". In the device side, the response is similar to the previous test. The device receives and decodes the JSON information, and starts the connection to the LoRa network, if the LoRa connection is not successful, the device reboots and starts transmitting again with NB-IoT.

The initial battery tests used LoRa as communication, because it is possible to use as geolocation method, and used a LiPo 3.7V 800mAh battery.

The first test revealed a duration of 8 hours and 30 minutes, for this operation mode. This mode consists in having activated the GNSS, Sensors (accelerometer) and perform the scan for Wi-Fi APs. This first mode is the one with better precision, but in the other hand is the one which requires more power.

In the second operation mode, only the GNSS and the sensors are activated, thus the duration was 9 hours, an increase of 5.88% in battery duration, without using Wi-Fi APs assisted location method.

The last operation mode, only LoRa communication was used, with a fixed payload of 110 Bytes, the same length as the previous test. The configurations in use were the following: for the sampling time 30 seconds, the antenna was an external one (Molex ISM 105262, omnidirectional with 0.4 dBi Peak Gain at 868 MHZ), spreading factor 7, bandwidth 125 KHz, the first three channels in 868.1 MHz, the coding rate 4/5. For the LoRa class was chosen class C with a transmission power of 14 dBm. Obtaining the total duration of 12 hours, the best of all the test, but at the same time is the least accurate, according to table 1.

The previous work done by [14], shows it is possible to have a pre-defined set of power consumption profiles, all of this in the wearable localization device. The goal of the work presented in this paper is to have the Geolocation Solver dynamically managing the operation modes, that combined with power consumption profiles, will create the energy profiles, that consist in a set of rules to improve the battery duration or the results precision.

After this phase of testing, the model needed to be validated. The first results for this model validation, only for the first functional stage, obtained during the initial phase of trials, were a total of over 50000 received messages from 10/02/2020 until 10/03/2020. The difference between the input received messages and the output sent messages was less than 1%. The implemented system was able to support spikes of simultaneously received messages from different devices, and the version in the test was able to process 1 message per 2 seconds. These first tests were conducted using 4 devices, used by 4 PwDs in real life trials. To conclude the model was able to dynamically choose the best location available.

VII. CONCLUSION

In this research, the authors presented the work to create a geolocation solver model, with three functional stages ("hierarchical", "smart", "advanced"), for wearable IoT devices. The implementation of this model, is capable of dynamically choose the best geolocation technology in each situation, for this to happen different operation modes were used. With these operation modes it is possible to improve accuracy and energy consumption.

Initial location tests were undertaken, to evaluate the accuracy results, of different geolocation techniques. The first with better average accuracy was GNSS with 10 meters. The second test was for Wi-Fi assisted location, this method scans the radio environment looking for Wi-Fi access points and based on the know location of such access points returns the approximate location for the device, the average result was 30 meters. The last one with the similar working principle as Wi-Fi was LoRa, the average accuracy was 300 meters.

The results for the tests, were preformed using as microcontroller, the FiPy [17] with the Pytrack [18] localization shield. This shield has a GNSS that is the Quectel L76-L [19] , and the antenna used for the GNSS was the internal one. Different hardware configurations will have different results.

The system was capable of hierarchically choose the best location and respond to information from different communication methods and data sources. Concluding the implementation of the "hierarchical" functional stage.

The initial tests for the implementation of the "advanced" stage, were undertaken, testing the battery consumption and communication resilience.

The results identified by the authors, prove a battery duration of up to 12 hours, when using the LoRa assisted location method, with a 30 seconds period between transmissions. For the communication resilience was conclude that the device was capable of dynamically change the communication method, and the system was able to handle the data transition.

The identified drawbacks of this work, were the maximum number of messages processed per second, and the fact that the Wi-Fi and LoRa locations were depended from third party providers.

The use case of this Geolocation solver, is for knowing the location of wearable devices, used by people who suffer from dementia, as it was tested in real life trials.

Finally, by analysing the developed system and its results, it is possible to verify that the approach is promising, and the defined model has provided an appropriate starting point for further research and applications in the field of assisted living location.

A. Future Work

Future work will focus on performing more tests , in order to discover possible faults or bottlenecks related to the ability of the system to process messages.

For the device future tests are required, to assess the location performance, of the different methods in indoor environments.

Further studies of the Geolocation Solver should be realised, in order to implement the "advanced" stage proposed in the solution. This comprises the development and testing of more operation modes, in order to have a profile based decision, taking into account variables such as if the PwD is at home or outside, if the person is accompanied or is alone, and the time of day.

ACKNOWLEDGMENT

The work has also been promoted under the project CARELINK, AAL-CALL-2016- 049 funded by AAL JP, and co-funded by the European Commission and National Funding Authorities FCT from Portugal and the national institutions from Ireland, Belgium and Switzerland.

REFERENCES

[1] A. Martin Prince *et al.*, "World Alzheimer Report 2015 The Global Impact of Dementia An AnAlysIs of prevAlence, IncIDence, cosT AnD TrenDs."

[2] "Alzheimer's Association | Alzheimer's Disease & Dementia Help." [Online]. Available: https://alz.org/. [Accessed: 15-Dec-2019].

[3] "Carelink Homepage - Carelink." [Online]. Available: http://carelink-aal.org/. [Accessed: 16-Dec-2019].

[4] B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANs," *GIoTS 2017 - Glob. Internet Things Summit, Proc.*, no. June 2017, 2017.

[5] LoRa Alliance, "LoRaWAN 1.0.3 specification," *Lora-Alliance.Org*, no. 1 [Online], Accessible: https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf, pp. 1–72, 2018.

[6] "MicroPython & #WiPy 2.0/3.0: Geolocation using WLAN - LeMaRiva|tech." [Online]. Available: https://lemariva.com/blog/2017/11/micropython-wipy2-0-geolocalization-using-wlan. [Accessed: 05-Jan-2020].

[7] "Developer Guide | Geolocation API | Google Developers." [Online]. Available: https://developers.google.com/maps/documentation/geolocation/intro. [Accessed: 05-Nov-2019].

[8] "Specifications | Wi-Fi Alliance." [Online]. Available: https://www.wi-fi.org/discover-wi-fi/specifications. [Accessed: 02-Jun-2020].

[9] M. D'souza, M. Ros, and M. Karunanithi, "An indoor localisation and motion monitoring system to determine behavioural activity in dementia afflicted patients in aged care," *Electron. J. Heal. Informatics*, vol. 7, no. 2, p. 14, 2012.

[10] N. E. Tabbakha, W. H. Tan, and C. P. Ooi, "Indoor location and motion tracking system for elderly assisted living home," *Proceeding 2017 Int. Conf. Robot. Autom. Sci. ICORAS 2017*, vol. 2018-March, no. September 2019, pp. 1–4, 2018.

[11] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, pp. 1–15, 2018.

[12] J. Danebjer, "A Hybrid Approach to GPS-Free Geolocation over LoRa," 2018.

[13] "Release 13." [Online]. Available: https://www.3gpp.org/release-13. [Accessed: 02-Jun-2020].

[14] M. Faustino, J. Calado, J. Sarraipa, and R. Jardim-gonçalves, "Adaptable power consumption profiles for wearable localization devices," 2019.

[15] "HERE Maps API | HERE Developer." [Online]. Available: https://developer.here.com/. [Accessed: 08-Nov-2019].

[16] "OpenCelliD - Largest Open Database of Cell Towers & Geolocation - by Unwired Labs." [Online]. Available: https://opencellid.org/#zoom=16&lat=37.77889&lon=-122.41942. [Accessed: 06-Nov-2019].

[17] Microcontroller and W. / Bluetooth, "FiPy datasheet," 2017.

[18] "Pytrack - Pycom." [Online]. Available: https://pycom.io/product/pytrack/. [Accessed: 29-Apr-2020].

[19] "Quectel GNSS L76-L." [Online]. Available: https://www.quectel.com/product/l76l.htm. [Accessed: 13-May-2020].

APPENDIX B

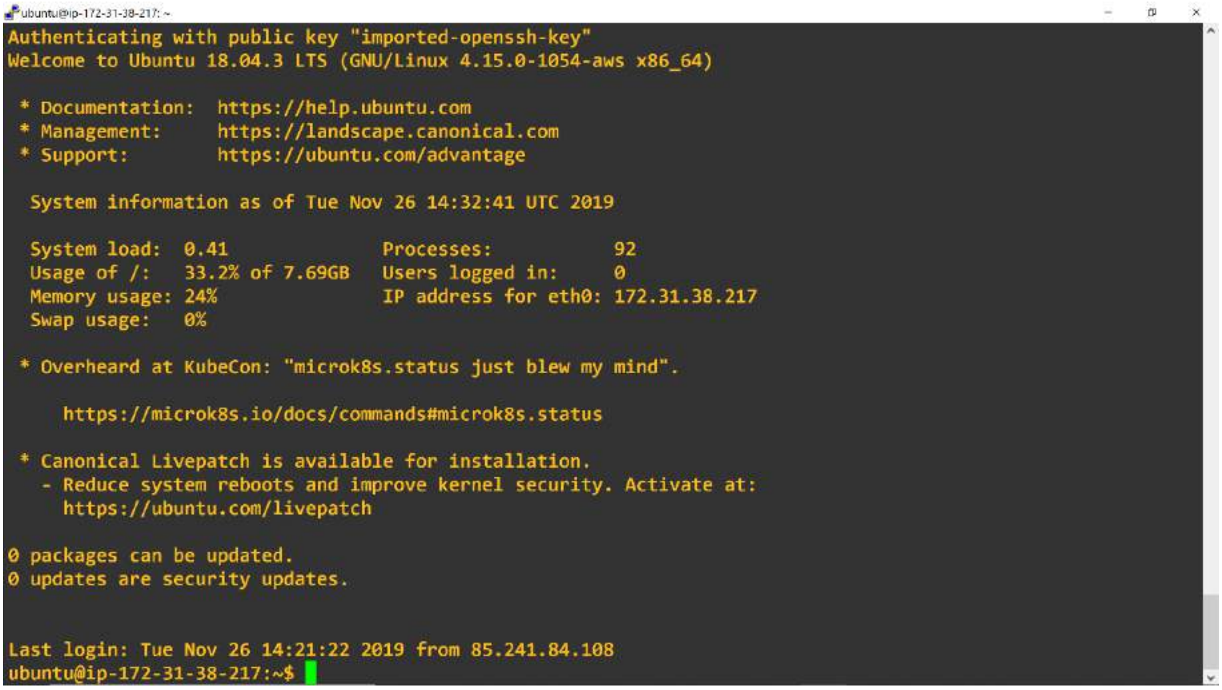
PART OF NODE RED INSTRUCTIONS

Node-RED

This document will contain the steps required for the installation, and the configuration of Node-RED. Node-RED will be used as backend allowing to do data processing of the packets received by The Things Network (TTN), the purposed idea is having devices communicating through LoRa with a gateway. This gateway will forward the packet to TTN, TTN will then communicate by MQTT to Node-RED. Node-RED will communicate also by MQTT to the Carelink platform, the reverse path is also possible. Node-RED is also used to process the " statusWifiAPs " messages (that are published in the MQTT broker, with a field containing the Wi-Fi access points that are near the device), which will make use of 3 Wi-Fi location API's to determine an "assisted" location, complementary to the GPS. This will enable the Pycom devices to have one more alternative to the GPS, in case it fails. Additionally, a third localization technique is also available when using the LoRa protocol.

1. Installation

SSH connection:



```
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1054-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Nov 26 14:32:41 UTC 2019

  System load:  0.41              Processes:           92
  Usage of /:   33.2% of 7.69GB   Users logged in:     0
  Memory usage: 24%               IP address for eth0: 172.31.38.217
  Swap usage:   0%

 * Overheard at KubeCon: "microk8s.status just blew my mind".

     https://microk8s.io/docs/commands#microk8s.status

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.


Last login: Tue Nov 26 14:21:22 2019 from 85.241.84.108
ubuntu@ip-172-31-38-217:~$
```

Figure 1 Ubuntu initial screen

4. Importing the Carelink flows

In order to replicate the work done for connecting TTN to the Carelink platform import the provided JSON. For security reasons some nodes need passwords. These passwords will be in a separate file.

After importing the JSON file, five different flows will be placed in the Node-Red as in the figures 17-22,

The first flow is the one responsible for receiving the uplink messages from TTN, and check if there’s a location in the status message If the location from the GPS is valid, it is kept. If the location is none, the field is replaced by the location provided by the LoRa Cloud API. After this, the message is published to the "status" topic in the Carelink platform MQTT broker

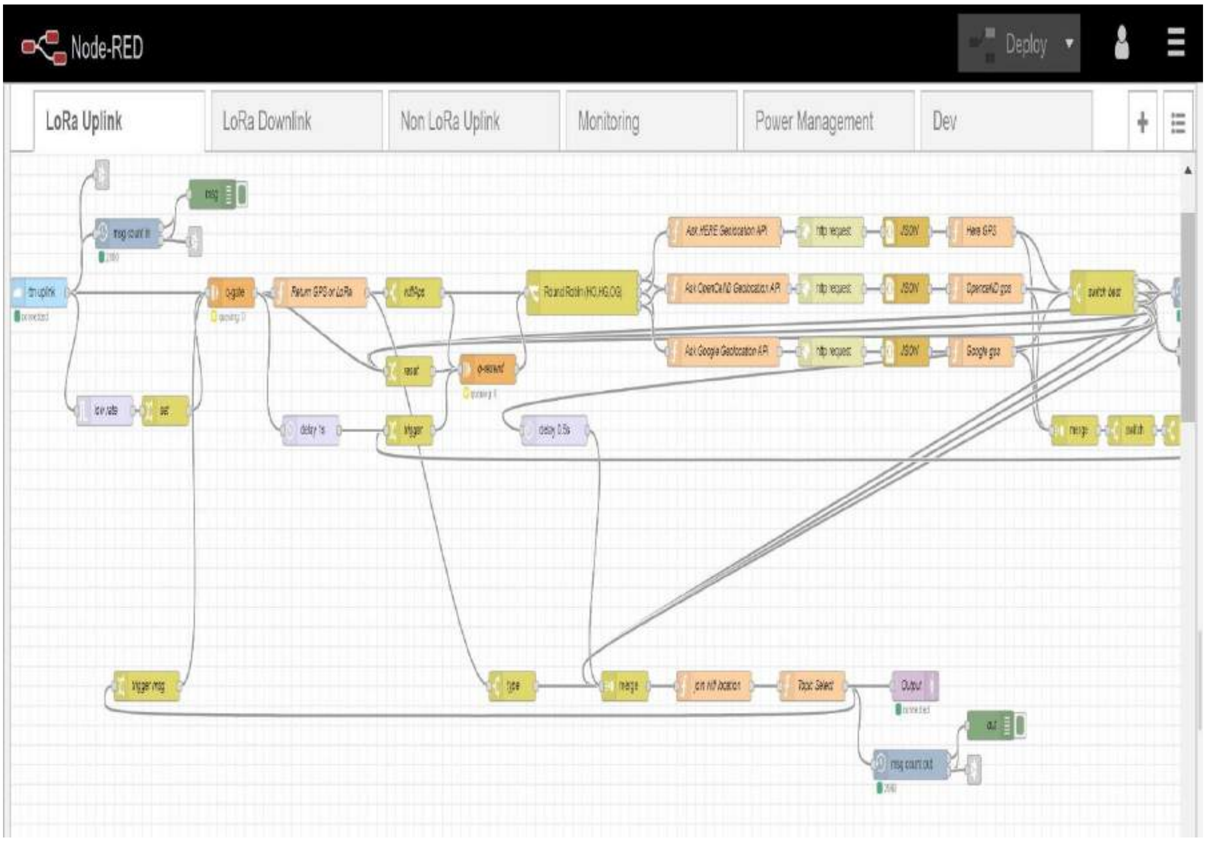
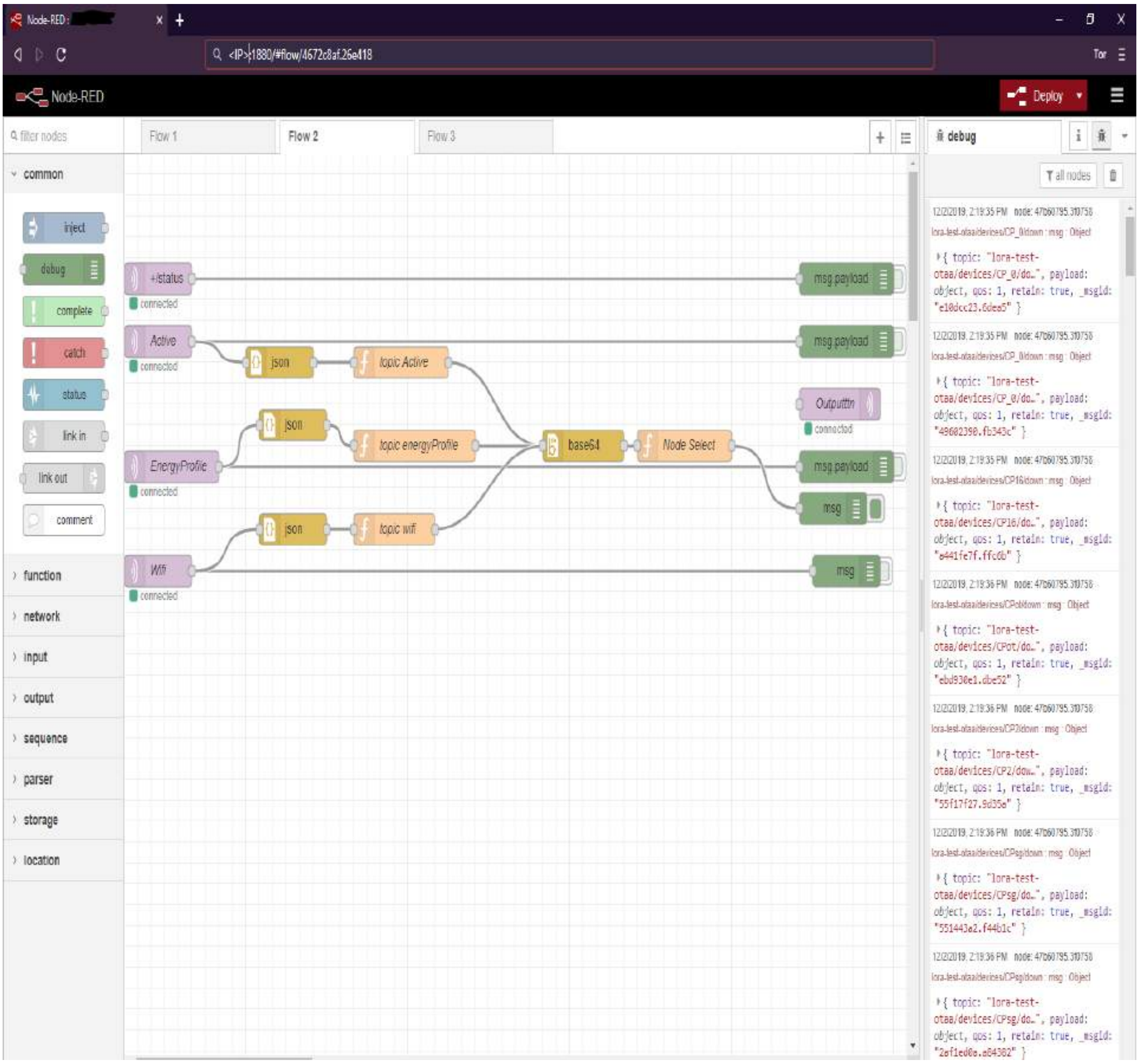


Figure 17 Node-RED flow wifi Assisted location for LoRa communication, TTN Uplink Status MQTT Downlink

The second flow is responsible for the LoRa downlink messages from the subscription of the different MQTT topics. The subscribed messages are converted to the right format so that they can be sent as a downlink for TTN.



```
12/2/2019, 2:19:35 PM  node: 47b60795.310758
lora-test-otaa/devices/CP_0/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CP_0/do…", payload: object, qos: 1, retain: true, _msgid: "e10dcc23.6dea5" }

12/2/2019, 2:19:35 PM  node: 47b60795.310758
lora-test-otaa/devices/CP_0/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CP_0/do…", payload: object, qos: 1, retain: true, _msgid: "49602390.fb343c" }

12/2/2019, 2:19:35 PM  node: 47b60795.310758
lora-test-otaa/devices/CP16/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CP16/do…", payload: object, qos: 1, retain: true, _msgid: "a441fe7f.ffc6b" }

12/2/2019, 2:19:36 PM  node: 47b60795.310758
lora-test-otaa/devices/CPot/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CPot/do…", payload: object, qos: 1, retain: true, _msgid: "ebd930e1.dbe52" }

12/2/2019, 2:19:36 PM  node: 47b60795.310758
lora-test-otaa/devices/CP2/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CP2/dow…", payload: object, qos: 1, retain: true, _msgid: "55f17f27.9d35a" }

12/2/2019, 2:19:36 PM  node: 47b60795.310758
lora-test-otaa/devices/CPsg/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CPsg/do…", payload: object, qos: 1, retain: true, _msgid: "551443a2.f44b1c" }

12/2/2019, 2:19:36 PM  node: 47b60795.310758
lora-test-otaa/devices/CPsg/down : msg : Object
▸{ topic: "lora-test-otaa/devices/CPsg/do…", payload: object, qos: 1, retain: true, _msgid: "2af1ed0a.a84302" }
```

Figure 18 Node-RED flow Subscribe MQTT to Downlink TTN

The third flow is responsible for the processing of the assisted location using the WiFi information provided by in the "statusWifiAPs" MQTT topic, by the devices, when not using LoRa.

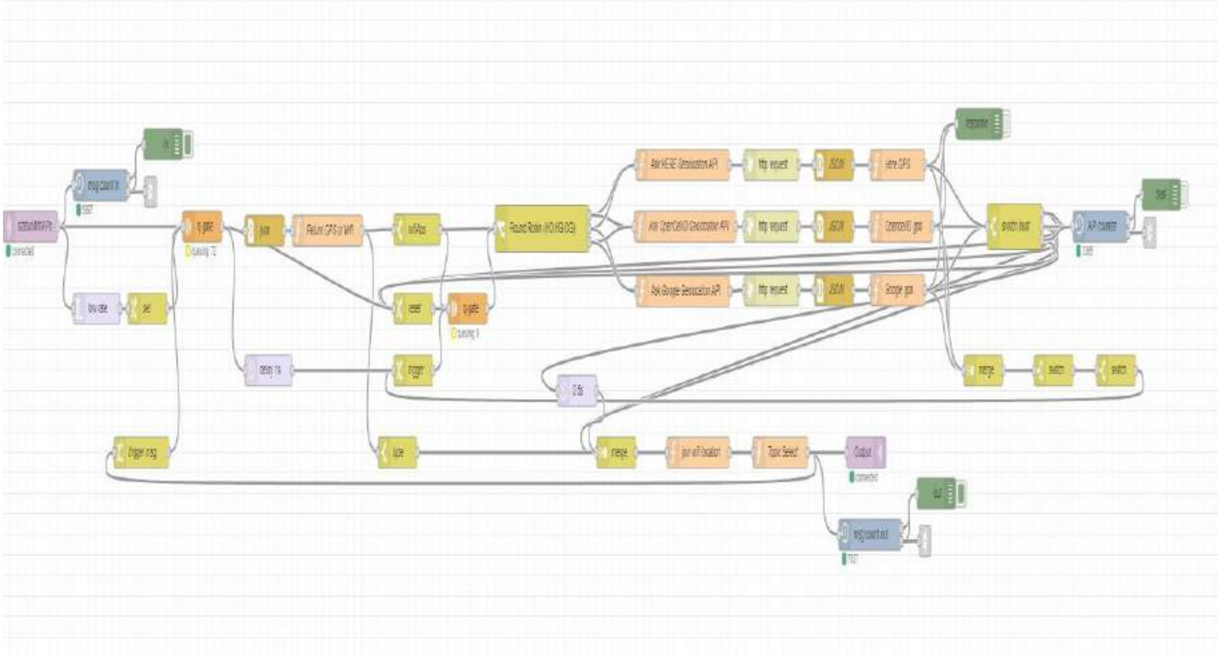


Figure 19 Node-RED flow wifi Assisted location for non-LoRa communication

The fourth flow is responsible for the monitoring and ensuring the correct function of the other flows. This flow uses several counters in different check points, and combines this information to create a report. This report is then sent daily by e-mail. This flow also analyses the different counters, and when a defined threshold is crossed, sends an alert SMS.

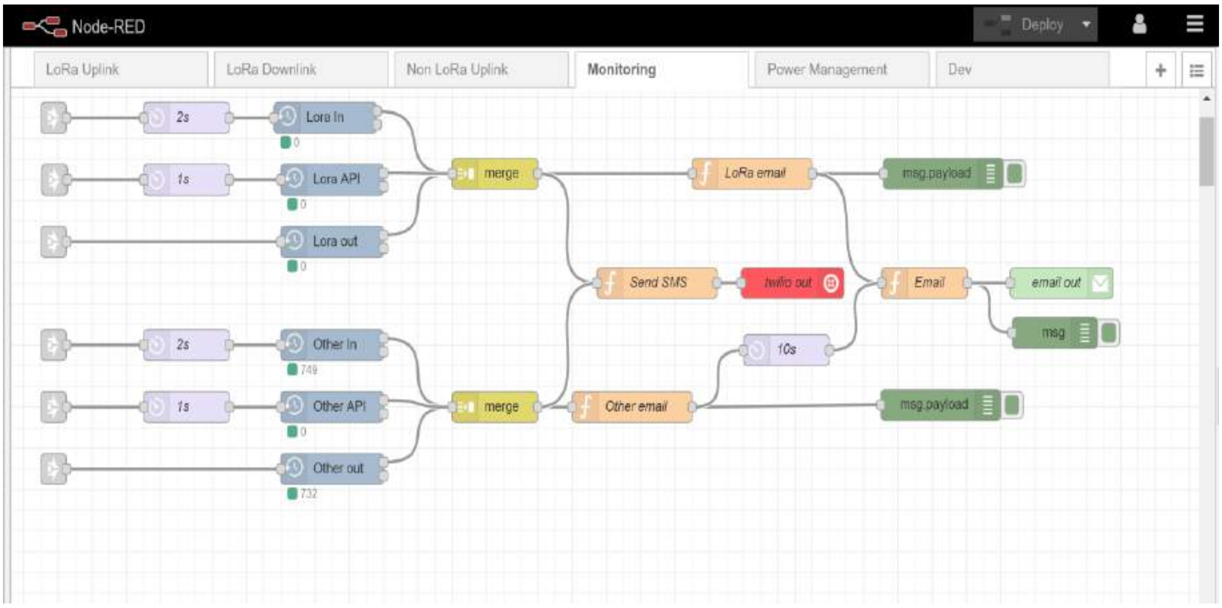


Figure 20 Flow Monitoring e-mail and SMS notification

The last flow is in charge for the power management of the devices. The working principle is the following first on the left side of the figure, the status topic is subscribed. Then the message is converted to JSON, and is filtered only the messages from Pycom devices. The message is analysed and the battery level is checked. If the battery level is valid, the remaining battery is calculated, using a polynomial function, as well as, the percentage of the total capacity, for example, "CP20" 10 hours 90%. Then based on this information the active components and the sample rate of them are adjusted.

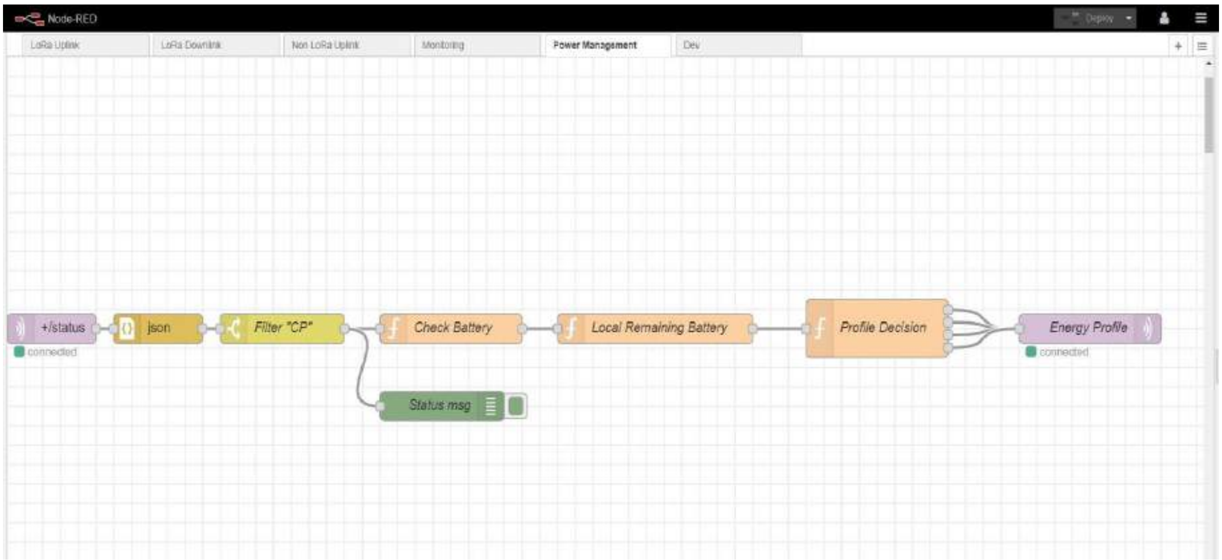


Figure 21 Flow Power Management

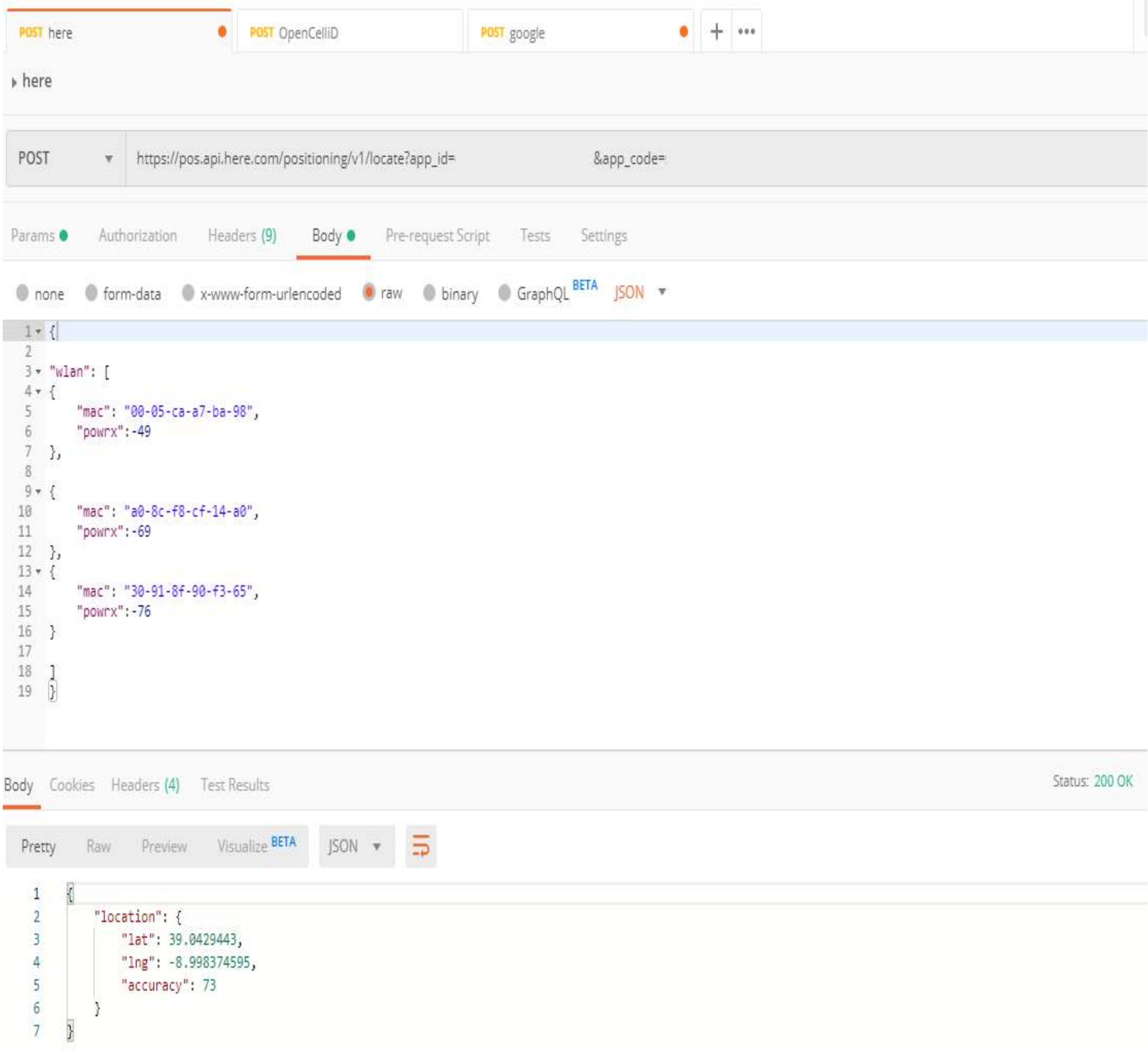
The Dev flow exists but is not represented in this document, since it is not used in a production environment. The development flow aims, to provide a sandbox, where the 3 features can be tested, alongside with the other flows that are in a production environment, but without being in production. The Debugging feature this function is used in case something goes wrong. It is possible to replicate, to find the error. The Updates Testing feature is used when the error is found, to be possible to develop and test an update. After some iterations, the update is finished and ready for production. The Map visualization feature is used to have a place for data visualization in a real world map, to have an idea of what is going to be shown next, in the platform, and for easy interpretation of the data.

5. API’s

For the flows represented by the figures 17 and 19, three different API’s are used. These API´s have the functionality to read the data from the "wifiAPs" field in the JSON, and returning a location based in the mac addresses and RSSI values.

For using the "here" API it is required an "appid" and "appcode", for the "OpenCelliD" API it is required a token and for the "google" API it is required a key.

The next step is testing API. This task is done through postman as shown in the next figures.



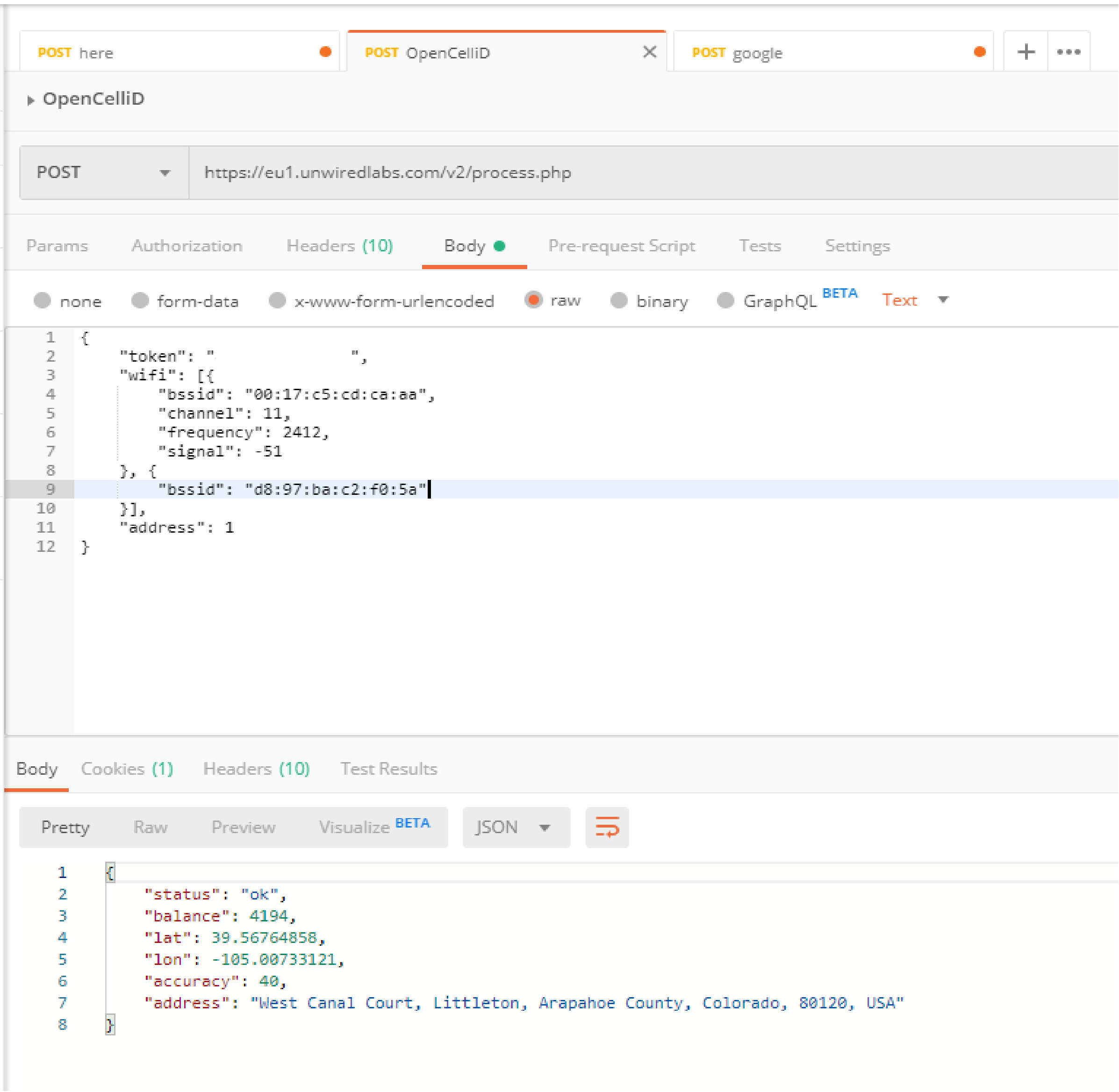
POST https://pos.api.here.com/positioning/v1/locate?app_id= &app_code=

```
{

"wlan": [
{
    "mac": "00-05-ca-a7-ba-98",
    "powrx":-49
},

{
    "mac": "a0-8c-f8-cf-14-a0",
    "powrx":-69
},
{
    "mac": "30-91-8f-90-f3-65",
    "powrx":-76
}

]
}
```

Status: 200 OK

```
{
    "location": {
        "lat": 39.0429443,
        "lng": -8.998374595,
        "accuracy": 73
    }
}
```

Figure 23 Postman here API test



POST https://eu1.unwiredlabs.com/v2/process.php

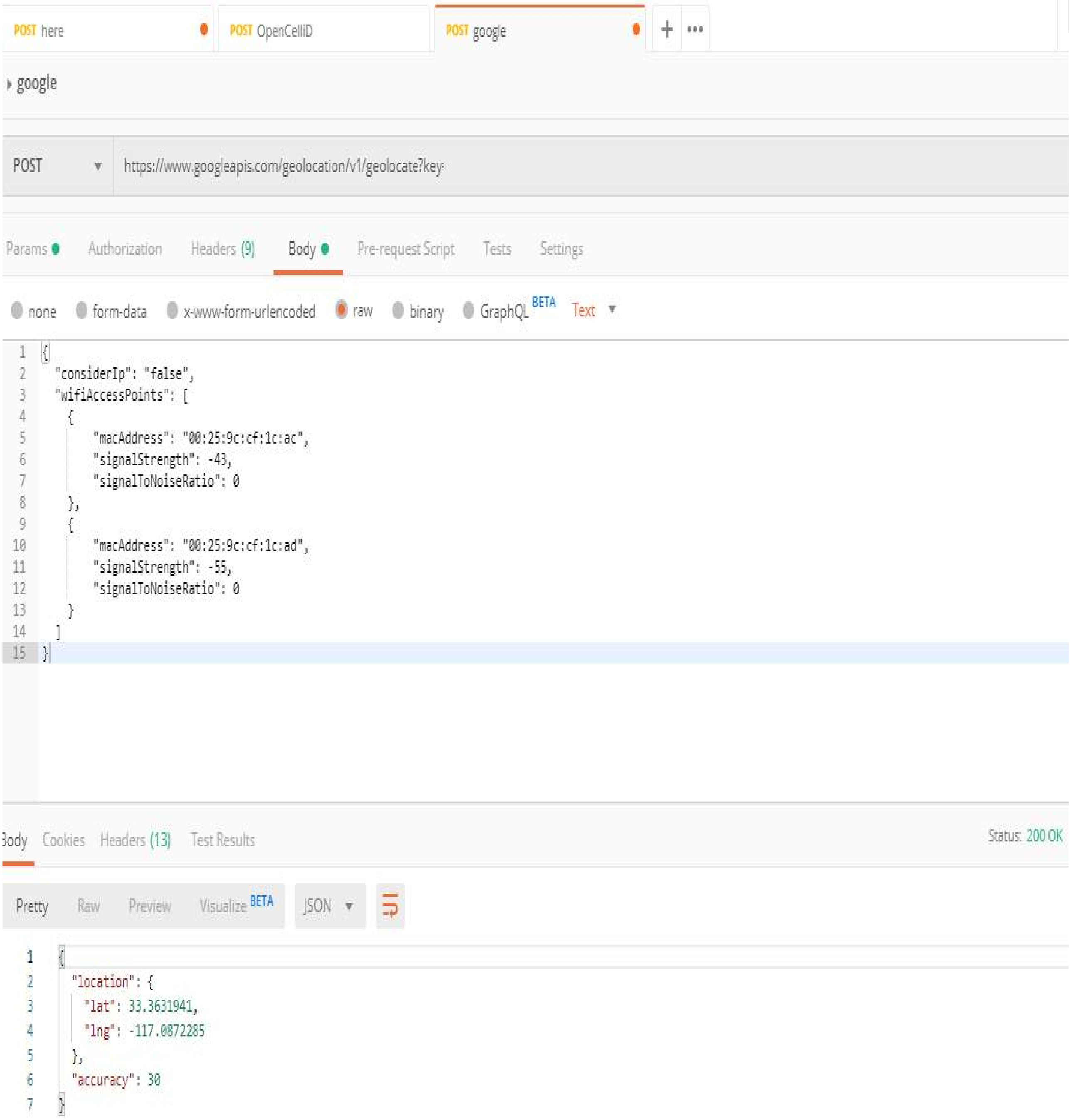
Body: raw, Text

```
{
    "token": "               ",
    "wifi": [{
        "bssid": "00:17:c5:cd:ca:aa",
        "channel": 11,
        "frequency": 2412,
        "signal": -51
    }, {
        "bssid": "d8:97:ba:c2:f0:5a"
    }],
    "address": 1
}
```

Body: Pretty, JSON

```
{
    "status": "ok",
    "balance": 4194,
    "lat": 39.56764858,
    "lon": -105.00733121,
    "accuracy": 40,
    "address": "West Canal Court, Littleton, Arapahoe County, Colorado, 80120, USA"
}
```

Figure 24 Postman OpenCelliD API test



POST https://www.googleapis.com/geolocation/v1/geolocate?key=

Body: raw, Text

```
{
  "considerIp": "false",
  "wifiAccessPoints": [
    {
        "macAddress": "00:25:9c:cf:1c:ac",
        "signalStrength": -43,
        "signalToNoiseRatio": 0
    },
    {
        "macAddress": "00:25:9c:cf:1c:ad",
        "signalStrength": -55,
        "signalToNoiseRatio": 0
    }
  ]
}
```

Status: 200 OK

JSON

```
{
  "location": {
    "lat": 33.3631941,
    "lng": -117.0872285
  },
  "accuracy": 30
}
```

Figure 25 Postman Google Geolocation API test

APPENDIX C

LIST OF CODE: TTN DECODER

Listing C.1: TTN Decoder

```
var hexChar=["0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"];
function byteToHex(b) {
 return hexChar[(b >> 4) & 0x0f] + hexChar[b & 0x0f];
}
function hexToInt(hex) {
 var num=hex;
 if (num>0x7F) {
   num=num-0x100;
 }
 return num;
}
function Decoder(bytes) {
if(String.fromCharCode(hexToInt(bytes[4]))==’-’){
var a;
var c = 0;
var time= "";
for (a = 0; String.fromCharCode(hexToInt(bytes[a])) != ’,’ ; a++) {
   time+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var lat= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’ ; a++) {
   lat+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var lon= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   lon+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
```

```
}
var alt= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   alt+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var hdop= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’ ; a++) {
   hdop+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var vdop= "";
for (a = ++c;String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   vdop+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var pdop= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   pdop+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var bl= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   bl+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var accx= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   accx+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var accy= "";
for (a = ++c; String.fromCharCode(hexToInt(bytes[a])) != ’,’  ; a++) {
   accy+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
var accz= "";
for (a = ++c; a <bytes.length ; a++) {
   accz+= String.fromCharCode(hexToInt(bytes[a]));
   c++;
}
return {
   wifiAPs:null ,
   timestamp:time,
   location:{ lat:lat, lon:lon, alt:alt, hdop:hdop, vdop:vdop, pdop:pdop,},
  batteryLevel:bl,
  sensor:{
    accelerometer:{x:accx, y:accy, z:accz }
    }
  };
```

```
}else{
 var mac1="";
 var i;
 for (i = 0; i < 6; i++) {
   mac1 += byteToHex(bytes[i]);
   if (i<5) { mac1+=’:’;}
}
var rssi1=-hexToInt(bytes[6]);
 var mac2="";
 for (i = 0; i < 6; i++) {
   mac2 += byteToHex(bytes[i+7]);
   if (i<5) { mac2+=’:’;}
}
var rssi2=-hexToInt(bytes[13]);
 var mac3="";
 for (i = 0; i < 6; i++) {
   mac3 += byteToHex(bytes[i+14]);
   if (i<5) { mac3+=’:’;}
}
var rssi3=-hexToInt(bytes[20]);
var time1= "";
var b;
var d=22;
for (b = d; String.fromCharCode(hexToInt(bytes[b])) != ’,’ ; b++) {
   time1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var lat1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’ ; b++) {
   lat1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var lon1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   lon1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var alt1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   alt1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var hdop1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’ ; b++) {
   hdop1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var vdop1= "";
for (b = ++d;String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   vdop1+= String.fromCharCode(hexToInt(bytes[b]));
```

```
   d++;
}
var pdop1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   pdop1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var bl1= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   bl1+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
   }
var acc1x= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   acc1x+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var acc1y= "";
for (b = ++d; String.fromCharCode(hexToInt(bytes[b])) != ’,’  ; b++) {
   acc1y+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
var acc1z= "";
for (b = ++d; b <bytes.length ; b++) {
   acc1z+= String.fromCharCode(hexToInt(bytes[b]));
   d++;
}
return {
   wifiAPs: { mac_1: mac1, rssi_1:rssi1, mac_2: mac2, rssi_2:rssi2,
   mac_3: mac3, rssi_3:rssi3 },
   timestamp:time1,
   location:{lat:lat1, lon:lon1, alt:alt1, hdop:hdop1, vdop:vdop1, pdop:pdop1},
  batteryLevel:bl1,
  sensor:{
    accelerometer:{ x:acc1x, y:acc1y, z:acc1z }
  }
};
}
}
```