**People's Democratic Republic of Algeria**
**Ministry of Higher Education and Scientific Research**

**University M'Hamed BOUGARA – Boumerdes**


**Institute of Electrical and Electronic Engineering**
**Department of Electronics**

Project Report Presented in Partial Fulfilment of

The Requirements of the Degree of

## 'Master'

## In Computer Engineering

Title:

# Design and implementation of a LoRa mesh network

Presented By:

**ALLAL CHERIF Zakaria**

**HAMMI Saad Eddine**

Supervisor:

**Dr. BELAIDI Hadjira**

Registration Number: 2020/2021

# *Acknowledgement*

*In the name of **Allah,** the **Most Beneficent** and the **Most Merciful***

*We thank Allah for all His blessing and strength that He gave us in completing this modest project.*

*We would like to thank our supervisor **Dr. Belaidi Hadjira** and our co-supervisor **Mr. Zakaria Rabiai** for all helps, advices and for the opportunity he gave us to research under his guidance and supervision.*

*We received motivation, encouragement and support from him during the completion of our work.*

*We would like to express our gratitude to all IGEE teachers and staff for their kindness and great help they provide to us to finish this work.*

*We would like to thank also our friends for their encouragement throughout this work.*

*Last but not least, we would like to express our deepest gratitude to our families for their endless love, unconditional support, and encouragement throughout our life.*

*Zakaria and Saad Eddine*

# *Dedication*

*I, Zakaria ALLAL CHERIF, dedicate this humble work*
*to my father and mother who are my stars and the beacons of my hope.*
*to my sister and brothers who never fail to show their utmost support and always watch over me.*

*To my childhood friend and all my friends from INELEC, who have stood by my side when I most needed help and supported me. I am blessed to have you involved in my life and hope for your continuous patronage.*

*Thank you very much.*

**“Zakaria”**

*Every challenging work needs self-efforts as well as guidance and orientation of those who are very close to our heart. I dedicate this humble work*

*To the bright memory of my aunt “HAMMI DJAMILA”, who loved and raised me like a mother, and to her two daughters who are continuing to support me like real sisters.*

*To my parents, the reason of what I become today and who have always supported me and oriented me towards the good.*

*To my brothers and sisters for their help to follow my studies, and their encouragement and sacrifices.*

***“Saad Eddine”***

## Abstract

With the widespread use of intelligent sensors, the need for a more reliable sensor network has emerged. LoRaWan is suitable for this kind of application and has an advantage over cellular networks due to its long range, low power capabilities and doesn't require paid plans for data transfer.

The purpose of this work is to design and implement a LoRaWan mesh network and test its capabilities in text based communication with the use of a smartphone application connected to lora nodes through low energy Bluetooth, allowing messages exchange between any two users connected to the network.

Such application of mesh network technology could be of use in case of disaster scenarios where cellular towers could be down; enabling a secure and reliable way of alternate communication.



**Keywords :** LoRaWan, mesh network, disaster scenarios, long range, low power.

## Table of contents

## Chapter 2 : Design and Architecture of LoRa Mesh Network

## Chapter 3 : Implementation and Results

## List of figures

## List of tables

## List of abbreviations

| | |
|---|---|
| **LoRa** | Long Range |
| **IoT** | Internet of Things |
| **GMS** | Global Mobil System |
| **LPWAN** | Low-Power Wide-Area Network |
| **Zigbee** | Zonal Intercommunication Global-standard |
| **ADAS** | Advanced driver-assistance systems |
| **GPRS** | General Packet Radio Service |
| **LTE** | Long Term Evolution |
| **CSS** | Chirp Spread Spectrum |
| **SF** | Spreading Factor |
| **MAC** | Media Access Control |
| **RF** | Radio Frequency |
| **BTS** | Base Transceiver Station |
| **WMN** | Wireless mesh network |
| **DSR** | Dynamic Source Routing |
| **CSMA** | Carrier-Sense Multiple Access |
| **EIGRP** | Enhanced Interior Gateway Routing Protocol |
| **DSDV** | Destination-Sequenced Distance-Vector |
| **TLV** | Type-Length-Value |
| **IS-IS** | Intermediate System to Intermediate System |
| **OSPF** | Open Shortest Path First |
| **CRC** | Cyclic Redundancy Check |
| **TCP** | Transmission Control Protocol |
| **IP** | Internet Protocol |

| | |
|---|---|
| **OLED** | Organic Light-Emitting Diode |
| **GPIO** | General-Purpose Input/Output |
| **DAC** | Digital to Analog Converter |
| **ADC** | Analog to Digital Converter |
| **ESP** | Espressif modules |
| **SRAM** | Static Random-Access Memory |
| **SoC** | System on Chip |
| **UDP** | User Datagram Protocol |
| **BLE** | Bluetooth Low Energy |
| **ToA** | Time on Air |
| **PDR** | Packet Delivery Rate |

# Introduction

Nowadays, almost every person has access to a mobile phone or a computer, this is due to the increasing number of the facilities and services they offer, since they became like a personal “hub”. In most cities and areas where a good coverage is guaranteed, consumers had the chance to unlock the full potential of these devices and utilized a variety of functions depending on their needs. They used the phone to connect with their loved ones, access information more easily, plan and have entertainment.

In fact, not everyone lives in these good coverage areas, some people are in a situation where either the communication infrastructure is not available or only at a high cost, e.g., in remote areas, in the agricultural sector, as a result of disasters, or due to political restriction. In addition to that, in the countries with less developed infrastructures, e.g., due to low population densities or due to economic reasons, cellular networks often cannot be used at all or cannot be established in an economically practical manner.

The priority for the people living in these areas is to be able to rely on a minimal level of connectivity, where they can reliably send and receive phone calls and text messages. Being able to communicate important messages at the time they need should be the goal in mind when designing a solution in these areas.

In this Work, the use of LoRa wireless technology is proposed as a communication facilitator in these situations. LoRa (Long-Range) is a long range and low power network protocol designed for the Internet of Things (IoT) to support low data rate applications.

using LoRa to transmit data, there are many topologies that can be used to have the best reliability and performance. The Mesh topology is chosen due to major advantages using it over other topologies, the main one being practicality.

The main objective of this project is to design a messaging architecture system that makes use of cheap and flexible devices and the LoRa technology to establish a link that can span wide areas with an easy to use interface.

The goal of the implemented solution is providing messaging services for the isolated areas that have no GMS signals coverage. Supplying low cost and low power messaging services to rural areas without affordable access to an existing communication network is definitely a crucial application.

Furthermore, in case of disasters, the LoRa network can be exploited to exchange emergency messages from long distances even if there was a disruption in the cellular infrastructure.

The report is divided into three chapters. The first chapter is the introduction to the world of internet of things, its architectures and different protocols, particulary LPWAN, which leads to talking about LoRa technology and its different features.

The second chapter then explains how lora mesh networks work, as well as demonstrating the logic behind conducting this project and the different parts and materials used.

The last chapter deals with the actual implementation, the performance and operation of our LoRa mesh network explaining all of its parts and the layered model as well as offering some experimental results and discussions. finally the reports wraps up with a conclusion about the work conducted.

# CHAPTER 1:

# Introduction to LoRa Technology

In this chapter, we are going to introduce the internet of things, which is the new hot technology, its four mostly used architectures, and the communication protocols that are built on top of it. After that, we are going to explore one of these protocols, which is low power wide area (LPWAN). Then, we dive into the heart of our project that is LoRa; we begin by introducing this new technology and all its features and advantages. Moreover, we introduce the idea of implementing LoRa in a typical mesh network. Finally, to sum up this chapter, we give a sample LoRa mesh network architecture.

## 1.1 Internet of Things definition

The Internet of Things (IoT) is the notion of connecting any device (can be switched on/off) to other devices via a network or the Internet. The IoT is a massive network of communicating "things" and people collecting and sharing data about the way of their functioning and their environment [1].

The IoT can be also defined as an extension of the internet and other network connections to different sensors and devices — or "things" — affording even simple objects, such as light bulbs, locks, and vents, a higher degree of computing and analytical capabilities [2].

Interoperability is one of the key aspects of the IoT that contribute to its growing popularity. Smart connected devices (things) have the ability to gather and share data from their environments with other devices and networks. Through the analysis and processing of the data, devices can perform their functions with little or no need for human interaction [2].

As the number of connected devices increase, the IoT continues its path of evolution, adding different layers to the data that is already being shared and processed, and giving rise to sophisticated algorithms that result in improved levels of automation [3]. And because of the variety of "things" that can be connected to it, the IoT has enabled diverse applications for individual users and entire industries alike as indicated in Figure 1.1.

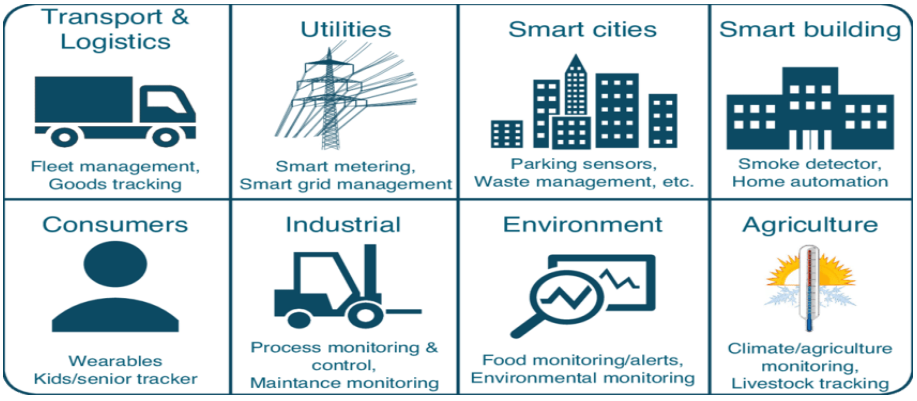



**Figure 1.1: Illustration of different industries that uses IoT.**

## 1.2 The communication protocols used by IoT:

The Internet Architecture Board released a guiding document that outlines the four communication channels used by the IoT. The four models also demonstrate how the connectivity of IoT devices helps extend the value of each device and adds quality to the overall user experience [3]:

**1.2.1 Device-to-Cloud:** Many IoT devices connect to the cloud, often with the use of wired Ethernet or Wi-Fi. Connecting to the cloud allows users and related applications to access the devices, making it possible to course through commands remotely as well as push necessary updates to the device software. Through this connection, the devices can also collect user data for the improvement of their service providers.

**1.2.2 Device-to-Gateway:** Before connecting to the cloud, IoT devices can communicate first with an intermediary gateway device. The gateway can translate protocols and add an additional layer of security for the entire IoT system. In the case of a smart home, for example, all smart devices can be connected to a hub (the gateway) that helps the different devices to work together despite having different connection protocols.

**1.2.3 Back-End Data-Sharing:** An extension of the device-to-cloud model, this model allows users to gain access to and analyze a collection of data from different smart devices. A company, for instance, can use this model to access information from all of the devices working inside the company building as organized together in the cloud. This model also helps lessen issues with data portability.

**1.2.4 Device-to-Device:** This model represents how two or more devices connect and communicate directly with one another. Communication between devices is usually achieved through protocols such as Bluetooth, Zigbee, and the latest technology which will be the subject of our project "**LoRa**". This model is often found in in wearables and in home automation devices, where small packets of data are communicated from one device to another, as with a door lock to a lightbulb.

## 1.3 Internet of things communication protocols:

The Internet of Things (IoT) starts with connectivity, but since IoT is a widely diverse and multifaceted realm, there exist several IoT communication protocols as shown

in Figure 1.2, each has its strengths and weaknesses as indicated in Figure 1.3. The following sited protocols are the six most common types of IoT wireless technologies [4]:

### 1.3.1 Cellular (3G/4G):

Well-established in the consumer mobile market, cellular networks offer reliable broadband communication supporting various voice calls and video streaming applications. On the downside, they impose very high operational costs and power requirements.

While cellular networks are not viable for the majority of IoT applications powered by battery-operated sensor networks, they fit well in specific use cases such as connected cars or fleet management in transportation and logistics. For example, in-car infotainment, traffic routing, advanced driver assistance systems (ADAS) alongside fleet telematics and tracking services can all rely on the ubiquitous and high bandwidth cellular connectivity.

### 1.3.2 Wi-Fi:

There is virtually no need to explain Wi-Fi, given its critical role in providing high-throughput data transfer for both enterprise and home environments. However, in the IoT space, its major limitations in coverage, scalability and power consumption make the technology much less prevalent.

Imposing high energy requirements, Wi-Fi is often not a feasible solution for large networks of battery-operated IoT sensors, especially in industrial IoT and smart building scenarios. Instead, it more pertains to connecting devices that can be conveniently connected to a power outlet like smart home gadgets and appliances, digital signages or security cameras.

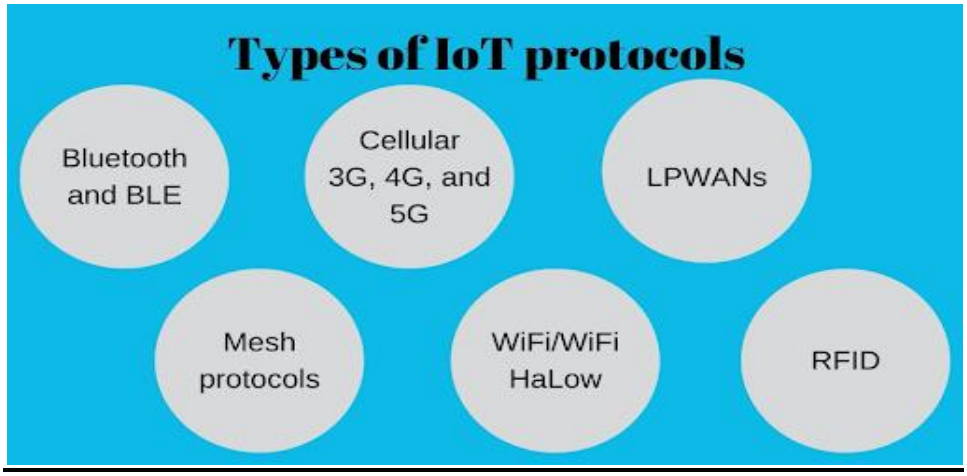



**Figure 1.2: Types of IoT protocols.**

### 1.3.3 Bluetooth:

Defined in the category of Wireless Personal Area Networks, Bluetooth is a short-range communication technology well-positioned in the consumer marketplace. Bluetooth Classic was originally intended for point-to-point or point-to-multipoint (up to seven slave nodes) data exchange among consumer devices. Optimized for power consumption, Bluetooth Low-Energy was later introduced to address small-scale Consumer IoT applications.

Bluetooth -enabled devices are mostly used in conjunction with electronic devices, typically smartphones that serve as a hub for transferring data to the cloud. Nowadays, Bluetooth is widely integrated into fitness and medical wearables (e.g. smartwatches, glucose meters, pulse oximeters, etc.) as well as Smart Home devices (e.g. door locks) – whereby data is conveniently communicated to and visualized on smartphones.

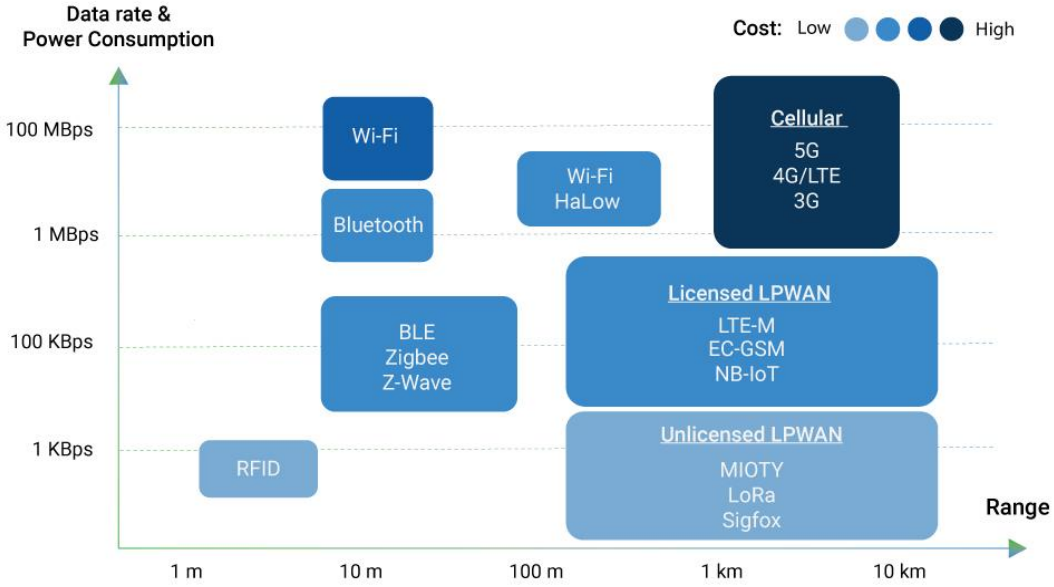



**Figure 1.3: Strength and weaknesses of each IoT solution [4].**

### 1.3.4 LPWANs:

Low Power Wide Area Networks (LPWANs) are the new phenomenon in IoT. By providing long-range communication on small, inexpensive batteries that last for years, this family of technologies is purpose-built to support large-scale IoT networks sprawling over vast industrial and commercial campuses.

LPWANs can literally connect all types of IoT sensors – facilitating numerous applications from asset tracking, environmental monitoring and facility management to occupancy detection and consumables monitoring. Nevertheless, LPWANs can only send small blocks of data at a low rate, and therefore are better suited for use cases that don't require high bandwidth and are not time-sensitive.

## 1.4 Low-Power Wide-Area Networks (LPWANs)

We are entering a world in which WiFi and Bluetooth may no longer be the best communication technologies for Internet of Things (IoT) applications. The IoT is gaining more ground each year. Experts predict there will be *45* billion connected devices by 2025 [5].

To support this incredible demand for bandwidth, new players have entered the IoT arena. For many industries, including supply chain, agriculture, healthcare, energy, and urban planning, Low-Power Wide-Area Networks (LPWANs) are a much better fit.

Low-Power Wide-Area Networks, also known as Low-Power Networks or LPWANs, are wireless telecommunication networks. Compared to WiFi and Bluetooth, an LPWAN is known for its ability to transmit small data packets over incredible, long-range distances using the unlicensed spectrum. While not ideal for things like video calls and playing video games, LPWANs shine for larger applications involving low-power sensors sending small amounts of data over long distances. Examples include automatic watering machines in the agricultural sector, smart lighting in industrial buildings, and sensing technologies associated with smart energy meters.

One of the most popular new LPWAN communication technologies is LoRa (short for "Long Range"), which runs on top of the LoRaWAN network protocol.

There are two reasons LoRaWAN has the attention of the market: First, it is currently the strongest technology available for LPWAN applications. Second, it is widely available, with more than 100 operators in over 100 countries.

## 1.5 LoRa technology:

Most communication radios like Zigbee, BLE, WiFi among others are of short range and others like, 3G and LTE, are power hungry and the span of their coverage areas cannot be guaranteed especially in developing countries. While these protocols and communication modes work for certain projects, it brings an extensive limitation like: difficulties in deploying IoT solutions in areas without cellular (GPRS, EDGE, 3G, LTE/4G) coverage and gross reduction in battery life of devices. Thus, envisaging the future of IoT and the connection of all kind of "things", located in all kind of places, there was a need for a communication medium tailor-made for IoT which supports its requirements of specifically low power, significantly long range, cheap, secure, and easy to deploy [6]. This is where **LoRa** comes in.

## 1.6 LoRa definition

LoRa (short for Long Range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. LoRa devices and wireless radio frequency technology is a long range, low power wireless platform that has become the de facto technology for Internet of Things (IoT) networks worldwide. LoRa devices and the open LoRaWAN protocol enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, disaster prevention, and more. According to Semtech,the major provider of LoRa devices and services a single LoRa base station can connect to sensors more than 15-30 miles away in rural areas [7].

## 1.7 LoRa features

### 1.7.1 Spread spectrum modulation

The behavior of a LoRa chirp is controlled by both the spreading factor, *SF*, and the bandwidth parameter, *BW*. The spreading factor is an integer value, typically ranging from 6 to 12, while the specified bandwidth can be chosen from values in the range of 7.8 to 500 kHz. Each chirp (or symbol) is encoded with *SF* bits, which means that there are $M = 2SF$ possible symbol values, where *M* is the modulation order. The instantaneous frequency of a chirp linearly increases or decreases across the bandwidth specified by *BW* over the symbol duration. The tradeoff between the range capabilities and the nominal bit rate depends on *SF* and *BW*. For instance, high *SF* and low *BW* allow for higher receiver sensitivity, but at a

lower bit rate, whereas low *SF* and high *BW* lead to reduced receiver sensitivity, but a higher bit rate [8]. The formula of SF is given by:

$$SF = \log_2(Rs/Rc)$$

Where Rs is the symbol rate, and Rc is the chirp rate which is equal to the bandwidth (one chirp per second per Hertz of bandwidth).

The unlicensed 868 MHz band in Europe and 900 MHz in the Americas opens new possibility of communication that can combat interference by leveraging the LoRa modulation, in which transmission speed is traded by range, in a classical application of Shannon's channel capacity formula:

$$C = B \log_2(1 + S/N)$$

were C is capacity or throughput in b/s, B is bandwidth in Hz, S is the signal power in W and N is the noise power in W, the formula for the bandwidth B is given by:

$$B = br / \log 2M$$

Where br is the Bit Rate and M is the modulation order. It is intuitive that the same capacity can be achieved by using a narrow bandwidth and a high S/N or a wide bandwidth and low S/N. LoRa offers a versatile method for lowering the throughput in order to allow the data transfer to occur in very low S/N instances, even with signal power of 1 percent of the noise power. This is accomplished by controlling the data rate by means of modifications of a so called spreading factor (SF), which determines the degree of expansion of the information rate (bit rate) to the transmission data rate. The SF ranges between 7 and 12 in Europe, corresponding to bit rates from 10937 b/s to 292 b/s. When the receiver is close to the transmitter a low SF can be used, which requires a receiver sensitivity of -123 dBm, whereas when the distance is higher or there are obstacles that reduce the received signal a spreading factor of 12 will decode signals as low as -136 dBm, which are much lower than the thermal noise (that at room temperature over a bandwidth of 125 kHz corresponds also to -123 dBm).

### 1.7.2 Frequency

While the LoRa technology is frequency agnostic, Communication between LoRa radios happen via the use of unlicensed sub-GHz radio frequency bands that are available around the world. These frequencies vary from region to region and often also differ between countries as shown in Table1.1. For instance the 868MHz is commonly used for LoRa communications in Europe, while the 915MHz is used in North America. Irrespective of the frequency, LoRa can be used without any major variation in the technology [9].

### 1.7.3 Adaptive Data Rate

LoRa uses a combination of variable bandwidth and spreading factors (SF7-SF12) to adapt the data rate in a trade-off with the range of the transmission as indicated in Table1.2. Higher spreading factor allows longer range at the expense of lower data rate, and vice versa. The combination of bandwidth and spreading factor can be chosen according to the link conditions and the level of data to be transmitted. Thus, a higher spreading factor improves transmission performance and sensitivity for a given bandwidth, but it also increases transmission time as a result of lower data rates. These can vary from as few as 18bps up to 40Kbp.

**Table 1.1: LoRa frequency bands in different countries [9]**

| | Europe | North America | China | Korea | Japan | India |
|---|---|---|---|---|---|---|
| **Frequency band** | 867-869MHz | 902-928MHz | 470-510MHz | 920-925MHz | 920-925MHz | 865-867MHz |
| **Channels** | 10 | 64 + 8 +8 | | | | |
| **Channel BW Up** | 125/250kHz | 125/500kHz | | | | |
| **Channel BW Dn** | 125kHz | 500kHz | | | | |
| **TX Power Up** | +14dBm | +20dBm typ (+30dBm allowed) | | | | |
| **TX Power Dn** | +14dBm | +27dBm | | | | |
| **SF Up** | 7-12 | 7-10 | | | | |
| **Data rate** | 250bps- 50kbps | 980bps-21.9kpbs | | | | |
| **Link Budget Up** | 155dB | 154dB | | | | |
| **Link Budget Dn** | 155dB | 157dB | | | | |

In definition by Technical Committee

**Table 1.2: LoRa physical bit rate according to different SFs**

| Data Rate | Configuration | | Physical bit rate | Max. MAC Payload size | Max. Frame Payload Size |
|---|---|---|---|---|---|
| | Modulation | Bandwidth | | | |
| 0 | SF12 | 125kHz | 250 | 59 | 51 |
| 1 | SF11 | 125kHz | 440 | 59 | 51 |
| 2 | SF10 | 125kHz | 980 | 59 | 51 |
| 3 | SF09 | 125kHz | 1760 | 123 | 115 |
| 4 | SF08 | 125kHz | 3125 | 230 | 222 |
| 5 | SF07 | 125kHz | 5470 | 230 | 222 |
| 6 | SF07 | 250kHz | 11000 | 230 | 222 |
| 7 | FSK | 50kbps | 50000 | 230 | 222 |
| 8-15 | RFU | | | | |

### 1.7.4 Immunity to Interference

LoRa's combination of multiple frequency channels, different spreading factors and a very low duty cycle have the great advantage of allowing a large number of devices to operate in the same physical area.

With LoRa, packets using different spreading factors are orthogonal, meaning that they are invisible to each other: as mentioned earlier, they simply appear as noise to one another. Therefore, two packets that arrive at the same time on the same receive channel at different spreading factors will not collide and, both will be demodulated by the gateway modem chip. However, two packets with the same spreading factor arriving at the same time on the same channel **might** result in a collision. However if one of the two packets is stronger by six dB, it will survive.

## 1.8 LoRaWAN is Not LoRA

It's important to understand that LoRaWAN and LoRa are not synonymous. LoRaWAN is the media access control (MAC) layer protocol on top of the physical layer that defines the communication protocol and system architecture for a device. It is, in short, the network upon which all LoRa devices operate; this difference is also illustrated in Figure 1.4.

The LoRaWAN link-layer spec is the same across all regions. Each region may, however, differ in the attributes of the physical layer of the device. These connected devices are not gateway-specific. Instead, consumers can grab any certified LoRaWAN device and trust it will perform necessary tasks like filtering out redundant messages and maintaining low

power consumption. LoRaWAN, of course, works with LoRa technology, but leverages the LoRa Alliance member ecosystem to provide a wide range of supplier options, from chips to the cloud. There are also many LoRaWAN devices already on the market and ready to work with the cloud.

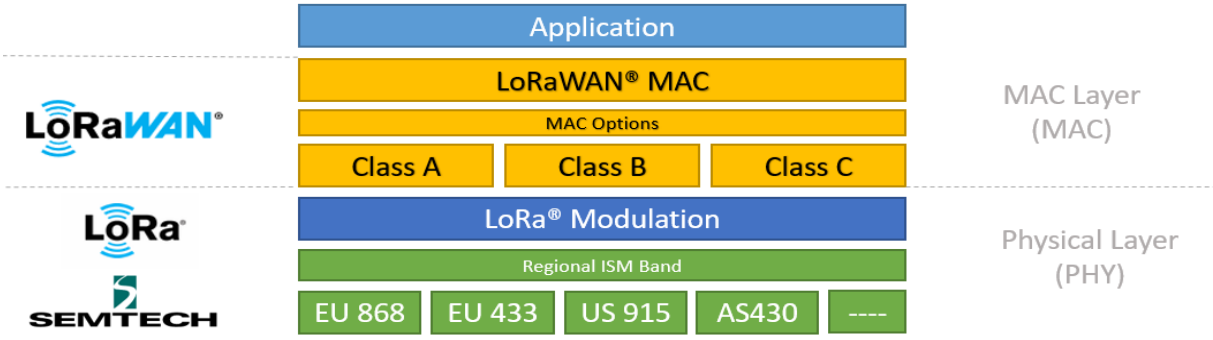



**Figure 1.4: The communication protocol and system architecture of LoRaWAN**

## 1.8 Advantages of LoRa network:

LoRa technology offers great advantages, which makes it an attractive network solution to many applications; these advantages are summed in Figure 1.5, and are explored in depth in the following:

### 1.8.1 Range:

With the RF power configurable up to 25 milliwatts, LoRa lets you send small data packages up to 30 kilometers (19 miles) depending on the environment. You can exceed this distance with more powerful chipsets and antennas. So far, the record LoRa transmission distance is 702 kilometers (436 miles).

### 1.8.2 Message capacity:

When it comes to capacity, a LoRaWAN network can support millions of messages. However, the number of messages supported in any given deployment depends upon the number of gateways that are installed. A single eight-channel gateway can support a few hundred thousand messages over the course of a 24-hour period. If each end device sends 10 messages a day, such a gateway can support about 10,000 devices[1]. If the network includes 10 such gateways, the network can support roughly 100,000

devices and one million messages. If more capacity is required, all that is needed is to add additional gateways to the network.

### 1.8.3 Battery life:

As far as battery life goes, the energy required to transmit a data packet is quite minimal given that the data packets are very small and only transmitted a few times a day. Furthermore, when the end devices are asleep, the power consumption is measured in milliwatts (mW), which allows for a long battery life as we will show in a later chapter (CH 3).

### 1.8.4 Cost:

When it comes to cost, given the capabilities of LoRa-based end nodes and gateways, only a few gateways – configured in a star network – are required to serve a multitude of end nodes. This means that capital and operational expenses can be kept relatively low. Also, when the cost-effective LoRa RF modules that are embedded in inexpensive end nodes are used in conjunction with the open LoRaWAN standard, the return on investment can be considerable.

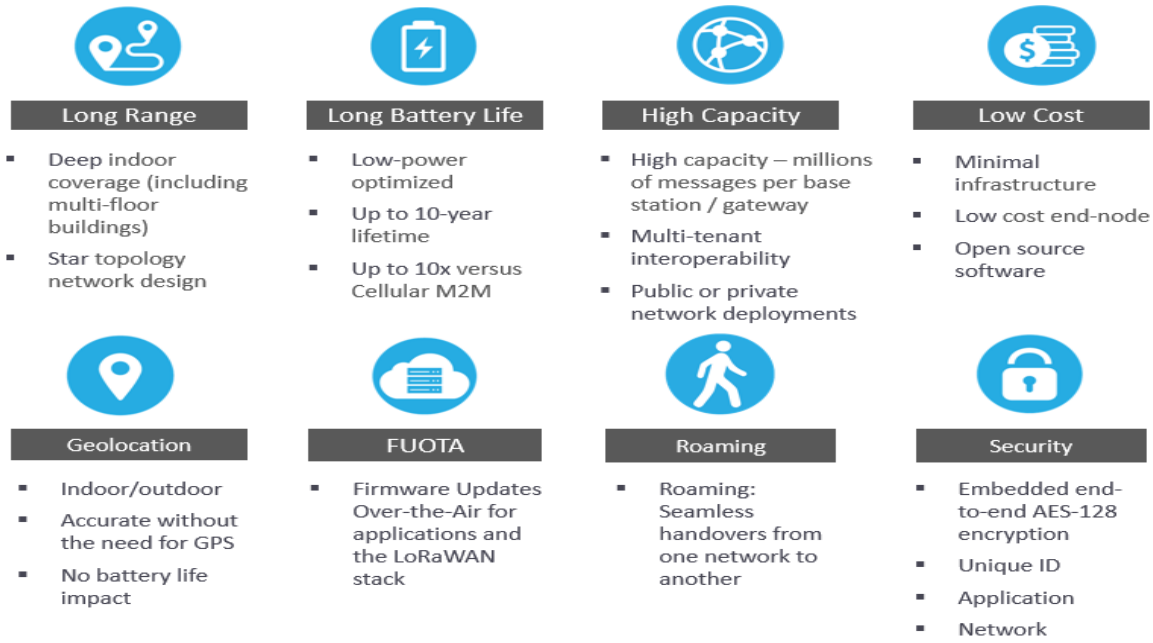



**Figure 1.5: Some LoRa advantages**

## 1.9 The LoRaWAN Network Architecture:

A typical LoRa network uses a **mesh topology** in which an end node can send messages to multiple gateways that communicate with the network server. Since an end node does not belong to a specific gateway, more than one gateway can receive a message sent by an end device. LoRa radio access technology is used in communications between an end device and the gateways. The gateways and network server are connected via standard IP connections.

- **End Device**: A LoRa End device is used to send small amounts of data at low frequencies over long distances. It can be utilized in various applications such as smart city, smart building, factory automation, farm automation, and logistics.
- **LoRa Gateway**: A LoRa gateway is a LoRa BTS receives packets from the end node via a radio link and then forwards them to the network server through the IP backhaul or 3G/4G broadband connections.
- **Network Server**: Network server manages the entire network. When it receives packets, it removes the redundancy of packets and performs a security check and then determines the most suitable gateway to send back an acknowledgement message.
- **Application Server**: An Application Server is the end server where all data sent by End Device is post processes and necessary action being taken.

## 1.10 Lora mesh network

### 1.10.1 Mesh network

A mesh network is a type of network in which the individual devices (called nodes) within the infrastructure connect directly, dynamically and non-hierarchically to as many other nodes as possible and cooperate with one another to create one virtual network that can efficiently route data between clients. These have become enormously popular among consumers in the past few years, particularly as more people opt for agile or remote working environments.

Fundamentally, a mesh node is a small radio transmitter that operates in a manner similar to a standard wireless router, using common 802.11 Wi-Fi standards to interface with devices and other nodes. Nodes within a network are programmed with specific software (routing protocol) that tells them how to interact with the larger network and handle information. For example, a mesh network that utilizes dynamic routing will automatically

select the quickest and most secure node path for data packets that are travelling through the system.

There are more than 70 routing protocols that can be used in wireless mesh networks. It can be said that more than 70 protocols have been tried in wireless mesh networks already and researchers are finding the new techniques to improve the performance of communication network.

Some of the names of those routing protocols are Associativity Based Routing, Ad-hoc on Demand Distance Vector, Better Approach to Mobile Ad-hoc Networking, Destination Sequenced Distance Vector Routing, Dynamic State Routing, Hazy Sighted Link State, Hybrid Wireless Mesh Protocol, Optimized Link State Routing Protocol, Order One Routing Protocol, Open Shortest Path First Routing, Predictive Wireless Routing Protocol, Zone Routing Protocol, Temporarily Ordered Routing Protocol and BABEL Protocol. Like all other routing protocols, protocols of WMN also specify how routers can communicate and exchange information with each other.

All routing protocols are made up from different algorithms that determine the right choice of the respective route. In this project we try to develop and test our own routing algorithm with respect to out LoRa mesh network.

Considering the LoRaWAN mesh network shown in Figure1.6, it is clearly shown that Sensor 1 is an intermediary between Sensor 4 and the network server. If Sensor 1 breaks, the intermediary role is assigned to Sensor 5, and the connection between Sensor 4 and the server is re-established.

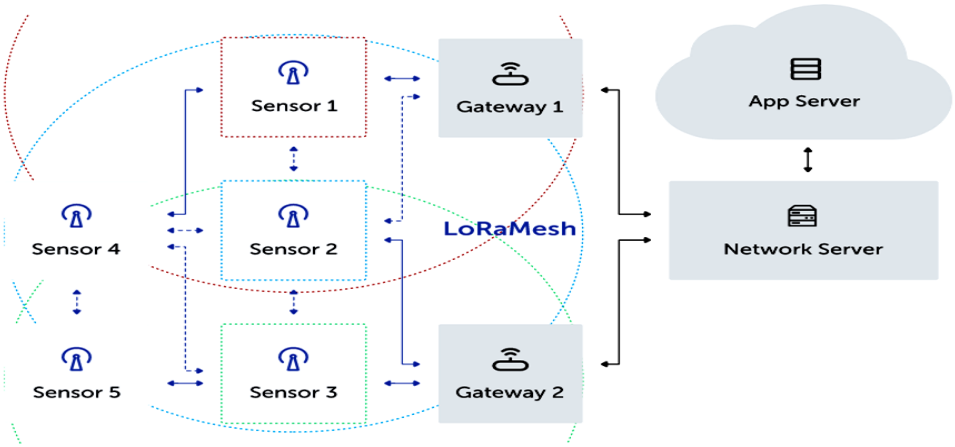



**Figure 1.6: Atypical LoRa mesh network.**

## 1.11. Conclusion

This chapter was an introduction to internet of things (IoT), and LoRa technology. We have explored LoRa and its advantages which motivated us to implement it in this project, our proposed implementation and the used technologies are explored in more depth in the next chapter.

# Chapter 2:
# Design and Architecture of LoRa Mesh Network

The purpose of this chapter is to describe the methodology followed to conduct this master project. Hence, this chapter describes the way the LoRa mesh network is built and the included devices and technologies. Firstly, the design goals of a generic LoRa modem firmware and routing protocol are presented. Secondly, the methodology of using LoRa on smartphones is covered. Finally, the hardware used in the design of our LoRa mesh networks is introduced.

## 2.1 Network structure

The structure of our LoRa mesh network consists of four devices (or nodes) with a single radio transceiver interface that transmits messages Omni-directionally; such that, under ideal conditions, all nodes should receive every message transmitted by a neighboring node, as illustrated in Figure 2.1.

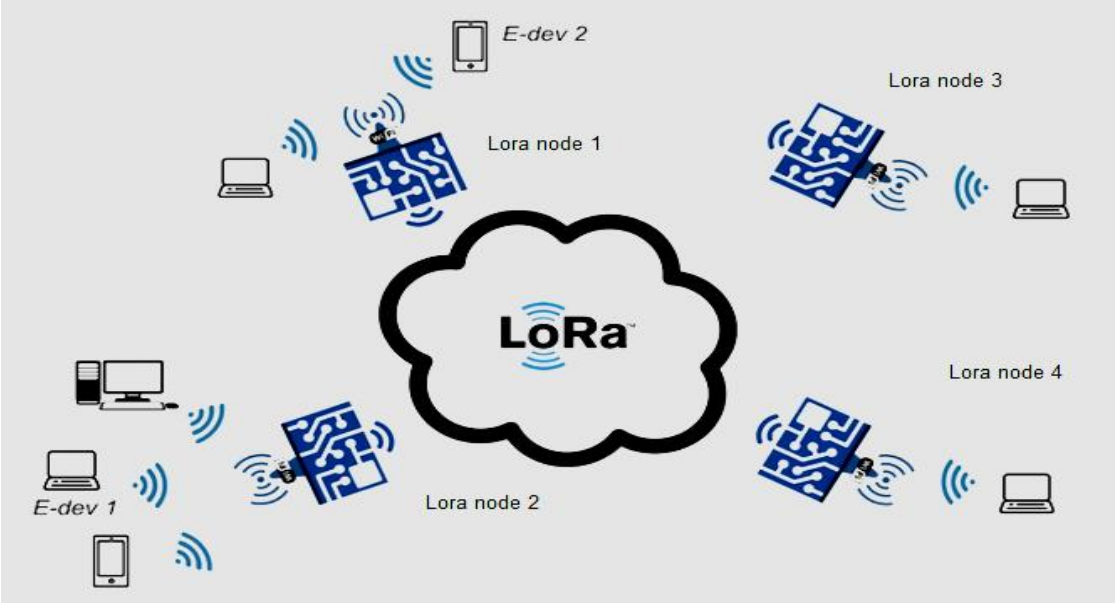



**Figure 2.1: Illustration of our LoRa mesh network**

This LoRa network is a heterogeneous wireless mesh network that utilizes the LoRa modulation scheme for **point-to-point** or **point-to-multipoint** connections between nodes. The LoRa routing protocol developed for the use of LoRa mesh networks is described. The «LoRaLayer2» protocol is a minimal distance vector routing protocol that incorporates elements of Dynamic Source Routing (DSR).

## 2.2 Why not LoRaWAN ?

So, after talking extensively about the LoRaWAN protocol and all its advantages, the protocol has been discarded and not used in this implementation of mesh network, and another protocol is selected instead (LoRaLayer2). Thus the question arises, why LoRaWAN has not been used for this project? And was talking about it devious and dishonest?

Well, in the LoRaWAN topology, direct communication is not supported between the LoRa nodes. Any node-to-node communication must be through the network server via two gateway transmissions. In LoRaWAN, there is no multiple access scheme to prevent the collision between the packets coming from the end-nodes, whereas carrier-sense multiple access (CSMA) is employed in conventional wireless networks, such as 802.11 (WiFi), to mitigate the collision rate. The link capacity of LoRaWAN decreases when a significant number of node transmissions occur simultaneously.

So, to be fair, LoRaWAN is the goto protocol when implementing a star topology; but in this project, we have deviated from the conventional LoRa network architecture to a full mesh topology, where every node can communicate with every node directly. There are two problems when LoRaWAN is used to serve applications in a poorly cell-planned private network:

1) Presence of end-nodes that are disconnected from the network (node to gateway).
2) A high collision rate because of the interference among the end-nodes.

To address the problems associated with standard LoRaWAN when employed in mesh private networks, we started looking for a modified LoRa protocol stack that supports mesh networking and time-division multiple-access for our LoRa mesh network. As there is no hierarchy in a mesh network, every node can relay a packet and cooperate with other nodes to efficiently route a packet to the gateways. Mesh networks dynamically connect end-nodes together and self-configure the routing paths. So, our search has led us to the LoRaLayer2 protocol which is based on Babel routing protocol.

## 2.3 Babel routing protocol

A routing algorithm developed for mesh networks is Babel, a network layer distance-vector routing protocol based on a distributed version of the Bellman-Ford algorithm. Babel uses a mechanism originally developed for the Enhanced Interior Gateway Routing Protocol

(EIGRP), known as “feasibility” that avoids routing loops and, therefore, makes counting to infinity impossible [10]. The drawback of the “feasibility” mechanism is that it can happen that a route is rejected also if it is loop-free. To avoid this problematic behavior, the “feasibility” method is combined together with another mechanism developed in the Destination Sequenced Distance Vector Routing (DSDV), known as “sequenced routes”. This mechanism avoids the “starvation” of a route happening when a router rejects all routes to a given destination, even those that are loop-free. However, in DSDV implementation, the sequenced routes algorithm is slow to react to a starvation episode. In Babel, instead, starvation recovery is accelerated by using explicit requests (known as “SeqNo requests” in the protocol) that signal a starvation episode and cause new sequenced routes to be propagated in a periodic way [11].

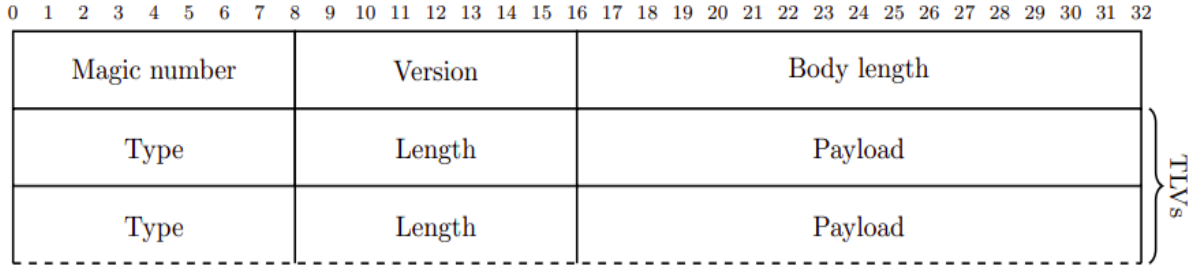
Babel relies on the computation, made by every pair of neighboring nodes *A* and *B*, of the link cost among *A* and *B*. Given a route between any two nodes, the route metric corresponds to the sum of the costs of all the links connecting *A* to *B*. However, Babel does not specify any algorithm for computing metrics and existing implementations are based on packet-loss metric in wireless links and a simple hop-count metric in all other types of links. The goal of the Babel algorithm is to compute, for each source node *S*, the routes of lowest cost to *S*. In detail, a Babel node periodically broadcasts *hello* messages, embedded in UDP datagram, to all of its neighbors, together with an *I Heard You* (IHU) message to every neighbor from which it has recently heard a *hello*. From the information retrieved from *hello* and IHU messages received from *B*, node *A* computes the cost *C* (*A*, *B*) of the link from *A* to *B*. A key feature of Babel is its fast reaction to network changes; this is obtained through triggered updates, triggered retractions and explicit requests, in which a Babel node requests an action to be done from another node or a set of nodes.

The main Babel packet, shown in Figure 2.2a, contains the following fields: (i) magic number, set to the arbitrary value 42 and identifying the Babel packet; (ii) version, identifying the adopted version of the Babel routing protocol; and (iii) body length, containing the length (in bytes) of the body following the packet header. The body is basically composed by a sequence of Type-Length-Value (TLV) Tuples, each of them composed by the following fields: (i) type of TLV (e.g., *hello* pr IHU messages); (ii) length of the body, considering only the Type and Length fields; and (iii) the TLV’s payload, consisting of a body and, potentially, a list of sub-TLVs. The structure is, thus, somehow modular and the packet can be extended

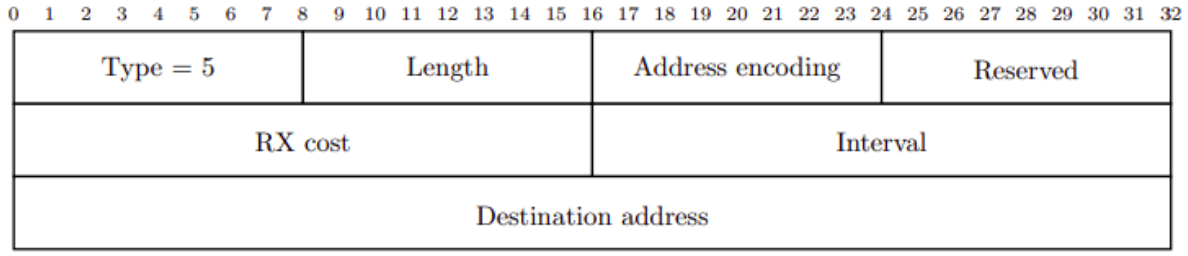
easily in order to support new functionalities. Among the available TLV types, the two most relevant, being responsible for route creation, are the *hello* and IHU packets, shown in Figure 2.2b and Figure 2.2c, respectively.

The *hello* TLV is used for neighbor discovery and for calculating the cost of a link and is, in turn, composed by the following fields: (i) type—corresponding to the value 4 in the case of *hello* TLV; (ii) length of the body; (iii) flags to handle the *hello* TLV, e.g., to determine a unicast or a multicast transmission; (iv) sequence number—which is increased every time an *hello* packet is sent; and (v) time window, after which the originator node will send a new *hello* packet. In an IHU packet, the type field is set to 5 and the remaining fields are shown in Figure 2.2c. Among the fields, the most relevant is the "Rx Cost", since it carries information about the cost of the link between two nodes.

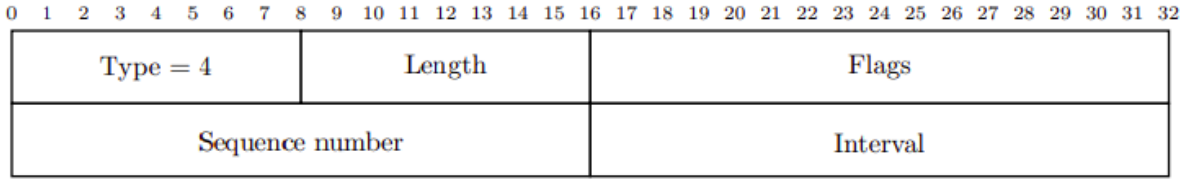
Other TLV types are: Router-Id, Next Hop and Update. The latter is used to force the request for an update of the route. The union of all available TVL packets "attached" in the main Babel packet makes this routing protocol easily extensible and customizable, leading to new solutions such as a new delay-based metric for route selection and Type of Service (ToS)-based routing. Finally, since Babel is based on periodic routing table updates, it is not adequate for: (i) large stable networks—since sending periodic updates even in the absence of topology changes increases the amount of control packets in the network, which can be reduced by using a protocol that relies on a reliable transport (such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS) or EIGRP) or DSR; and (ii) low-power networks, since periodic updates use battery power even when there are no topology changes and no user traffic—this makes Babel wasteful in low-power networks.




**(a)**




**(b)**




**(c)**

**Figure 2.2: Babel packet structure. (a)** Main Babel packet. (b) *hello* packet. (c) IHU packets [12]

## 2.4 Babel applications

IEEE 802.11-based mesh networks have been used for several applications, both alone or in combination with other radio technologies [13]. For example, the use of IEEE 802.11 mesh networks has been investigated for indoor positioning, in combination with Ultra-Wide Band (UWB) technology. In particular, an IEEE 802.11-based mesh backbone is used to combine its high data-rate feature together with high-accuracy UWB-based localization, since it allows an easy and fast deployment in areas without existing infrastructures. Video surveillance scenarios have also been studied in the literature, using IEEE 802.11 mesh networking as enabling technology. Another application is a video surveillance system for a large outdoor area based on an IEEE 802.11s mesh network. More in detail, the user mounts a camera on a drone moving over the area of interest and sending the video to a control center. As the drone is a mobile node and is not always in visibility with the control center, some static IEEE 802.11s nodes are deployed on the ground guaranteeing mesh connectivity. Another application is a wide area surveillance mesh network aimed at preserving forests from fire. The proposed system is based on an IEEE 802.11-based mesh network, which supports communications both inside and outside the area afflicted by the fire, thus enabling

the surveillance of the environment and guaranteeing the communication between the nodes in the area [13]. And another application is this project.

## 2.5 LoRaLayer2 routing protocol

The LoRaLayer2 protocol is heavily reliant on existing LoRa technology and development; it is an open source routing protocol that was inspired by Juliusz Chroboczek's Babel routing protocol [14]. In our implementation, we only modified the existing protocol to suit our own implementation.

The Lora nodes work as standard WiFi access points to provide connectivity to nearby devices. The interface with the messaging application is a web based page. The user can decide whether to send a text message to a specific user or to send a broadcast to all users. Every user need to "register" before interchanging any message. Registration is required to allow the system to localize end-point that can possibly be dynamically present.

When a user sends a message, the local node "learns" that the user is connected through it and creates an entry in a table. The first step is to discover where the destination user is located. To this end the hub sends a broadcast message using the physical layer LoRa protocol to all surrounding devices. The device which has that end user as a registered one, replies to the requesting hub.

### 2.5.1 Packet structure

The packet structure used by this protocol is very simple with a limitation; it should be less than 256 bytes according to Semtech (the company that maintains LoRa). The reason is that a LoRa device has a FIFO buffer size of 256 bytes, and with a low data-rate configuration, the time on air with a 256 byte payload will be very long (several seconds or even longer), which is not good for resisting fading and high interference environments.

The packet is rewritten at every hop in the route, while the datagram inside of the packet remains the same over the course of the route.

Semetech's specifications for LoRa transceivers include a header (set explicit or implicit). When set in explicit mode, a header will be sent that contains the payload's length, coding rate, and the presence of a CRC. In implicit mode, only a preamble is sent to initiate

communication. In this project, we are using an explicit header, which is structured as given in Table 2.1.

**Table 2.1: LoRaLayer2 packet header structure**

| Byte 0 | Byte 1 | Byte 2 - 5 | Byte 6 - 9 | Byte 10 | Byte 11 - 14 | Byte 15 | Byte 16 | Byte 17 - 255 |
|---|---|---|---|---|---|---|---|---|
| ttl | totalLength | sender | receiver | sequence | source | hopCount | metric | datagram |

where,

**ttl** is the "time to live", i.e. the number of hops allowed before a message is discarded by a node.

**totalLength** is the entire length of the packet (header length + datagram length).

**sender** is the 4 byte address of the node that is transmitting the message.

**receiver** is the 4 byte address of the intended receiver of the packet.

**sequence** is the global message counter of messages transmitted by sender, determines packet loss rate.

**source** is the 4 byte address of the node that generated the datagram.

**hopCount** is the number of hops that the sender is from the source, this is incremented by the receiver.

**metric** is the quality of the link between the sender and the source.

**datagram** is the content of a packet to be interpreted by the application.

### 2.5.2 Addressing

There are two possible ways of addressing on a LoRa mesh.

1. single node addressing
2. multicast addressing

Single node addressing implies that messages are sent with a single destination address. Every node has a unique 4 byte address. Initially, we are using the last 4 bytes of MAC

address of the node's WiFi interface as its unique address. This should be unique across our four nodes network.

Multicast addressing implies that messages are sent with multiple destinations, these destinations have a shared 4 byte multicast address. This would enable nodes to subscribe to a channel by accepting messages for a specific multicast address.

### 2.5.3 Node states and convergence

The node can be said to be in a constant state of simultaneous learning and forwarding all the time. When a node joins a LoRa mesh network it can begin sending messages to its neighbors immediately, but it has to wait to hear messages sent to nodes more than 1 hop away before it can begin routing outside of its immediate neighbors.

Similar to the Babel, a TCP/IP mesh networking protocol, a node in a LoRa mesh network can be said to know the address of every node in the network and the nextHop to reach that address, but it does not know the entire topology of the network.

Given the physical constraints of LoRa modulation, a LoRa mesh network converges relatively slowly (approx. on order of 1 minute for every 6 hops and perhaps longer under non-ideal conditions). But for our four nodes network it has takes close to 40s for the nodes to completely converge.

## 2.6 Hardware setup

The proposed system is based on TTGO ESP32 Lora with OLED, ESP32 microprocessor. This board is quite simple in that it includes a nice OLED screen and Bluetooth radio, the board used is shown in Figure 2.3.


**Figure 2.3: TTGO T-Beam board**

The following are the system specifications:

- TTGO T-Beam node is built around the ESP32 chip.
- It has 4MB of SPI flash.
- It operates at 433 MHz, 868 MHz and 915 MHz.
- TTGO T-Beam node includes an antenna, to send and receive LoRa packets.
- It uses a LoRa chip from the HopeRF RFM9X family. The node has a total of 26 pins with GPIO, ADC, VP/VN, DAC,
- It can be fed by batteries.
- Contains a slot for an external SD card.

### 2.6.1 ESP32 Microprocessor

The system is based on the ESP32 microprocessor, which is a robust and adaptive module with much built-in functionality. This module was assembled with different hardware devices. It acted as a bridge between the user application and the LoRa module at the sender end through Wi-Fi connection, between different sensors and controlling switches at the receiver end through wired connection.

### 2.6.2 LoRa Module

The LoRa transceiver module provides fanatical long-range spectrum for establishing intercommunication and reliable interference security with low power consumption. This made the automation system more robust. An antenna is integrated with the ESP32 board for long range communications. LoRa has a built-in SemTech SX1276 engine and 127 dB dynamic range RSSI that enables controlling different appliances from 3 km to 12 km [15].

With this board we can choose whether to connect to our smart devices via UDB OTG cable, Bluetooth or WiFi .We simply flashed the board with the appropriate firmware image downloaded from the disaster.radio Github repository. Then, some changes were made to the firmware image to suit our own implementation; moreover, we developed our own web server and web application with React.Js.

In a typical LoRa network, the end devices directly communicate to LoRa via gateways (front end) and the data from the end devices are sent to a network or cloud server via a gateway (back end). ESP32 module is chosen for the automated system developed in this work to control the total appliances of the home or the institute due to its more powerful processing capability, and because it supports both Wi-Fi and Bluetooth connectivity without any gateway.

The power consumption of the ESP32 module is lower than other devices such as Raspberry Pi. The system architecture presented in this work is different from other reported architectures. Here, a one-way Wi-Fi connection (at the sender side) connected the ESP32 module to the Android phone, and the signal went to the LoRa module from the phone. Furthermore, a LoRa module connection with ESP32 (at the receiver side) eliminated the requirement of Wi-Fi connection. The combination of simple modular design, cheaper hardware and controlling it by a smart phone allow making changes in the system by adding or removing certain features in order to expand its functional capability.

## 2.7 Enabling LoRa on smart devices

LoRaLayer2 is an open source routing protocol for LoRa. In this project, we have adopted this routing protocol for our own implementation, and our contribution to the open source project is a web application built using React.Js.

## 2.8 The Web application

The firmware opens up a websocket using the ESPAsyncWebServer Javascript library. Through this, client-side javascript can transmit and receive messages over the LoRa transceiver. For our web app, we have written a websocket client that sends and receives messages in the format expected by the firmware as follow:

<msgID><msgType>|<msg>

Where :

<msgID> is a two-byte binary unsigned integer representing an arbitrary sequence number, this is sent back to the websocket client with an ! appended to act as an acknowledgment and could be used for error-checking,

<msgType> is a single binary utf8 encoded character representing the application for which the message is intended, such as 'c' for chat, 'm' for maps, or 'e' for events…

<msg> is a binary utf8 encoded string of characters limited to 236 bytes, this can be treated as the body of the message and may be used to handle client-side concerns, such as intended recipient or requested map tile.

An example message may appear as follows, 0100c|<noffle>@Hello world !

Each person that joins the conversation in our group must be allocated a unique numerical ID, between 1 and 254.

## 2.9 Conclusion

In this chapter, our approach to implement a LoRa mesh network was explained. Our network architecture and all used components are explored without going into much details of how each component interacts with others. However, our network operation, and our devices interactions will be discussed in the next chapter.

# Chapter 3: Implementation and Results

This chapter deals with the implementation of our LoRa mesh network. After the explanation of our design and architecture in the last chapter, this chapter dives into the performance and operation of our network. It is divided into four parts. First, the performance of the used hardware is tested; after that, the firmware operation is explained through a layered model; then, the web app functionality is explained details. Finally, some experimental results and discussions are offered.

## 3.1 The hardware

As shown in the block diagram illustrated in Figure 3.1, our system is mainly made of a number of nodes connected to each other forming a mesh network.

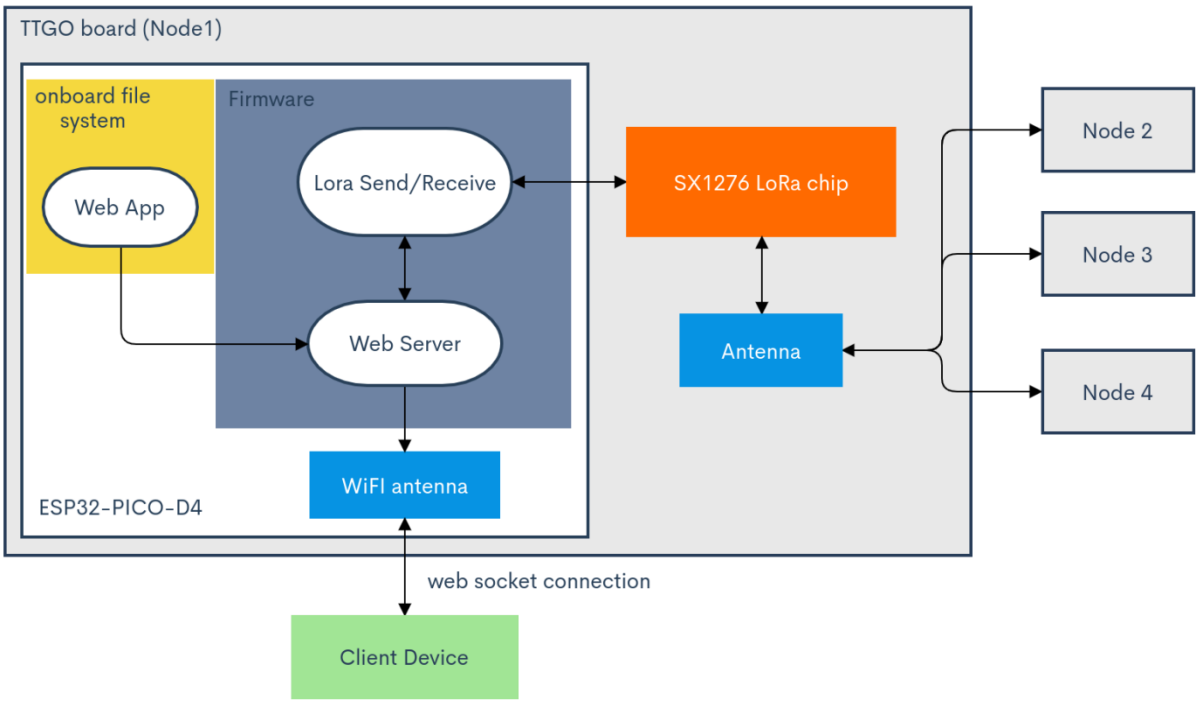



**Figure 3.1: Overall System's Block Diagram.**

Each node is a TTGO T3 LORA 32 868MHz board which consists of: ESP32-PICO-D4 chip (240 MHz dual core processor), SRAM: 520KB, Flash memory: 4MB, Built-in Wi-Fi and SX1276 LoRa chip.

The ESP32-PICO-D4 chip is a system-on-chip (SOC) microcontroller with built-in Wi-Fi capabilities. This provides Wi-Fi access in the node's immediate vicinity, allowing most modern devices to connect to the mesh network. The nodes communicate with each other using a long-range, low-bandwidth radio transceiver, the LoRa SX1276 chip and its chirp-like modulation scheme provide the node-to-node links for the network.

The values shown in Table 3.1 are used to estimate the average power consumption of each node and verify whether theoretical consumption is compatible with a 1000 mAh battery, which is a standard capacity for IoT devices.

**Table 3.1: Current consumption of transceiver SX1276 depending on the radio mode.**

| Radio Mode | Description | Current |
|---|---|---|
| *OFF* | The transceiver is connected to a power source but it has not yet warmed up the components | 0.2 µA |
| *SLEEP* | The transceiver is in energy saving mode and can switch to another mode rapidly | 0.17 µA |
| *RX* | The transceiver is receiving a signal | 14 mA |
| *TX* | The transceiver is transmitting a frame | 46~100 mA |

Furthermore, simulation experiments have been configured to use the 868 MHz frequency and 125 kHz of bandwidth. It has also been decided to use SF 7, and an 8-bit preamble. That way, it resembles a deployment in a European city which may be used for comparison to the equivalent LoRaWAN.

After the current measurement, we had to check with the datasheet published by the manufacturer, as shown in Figure 3.2.

| Symbol | Description | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| IDDSL | Supply current in Sleep mode | | - | 0.2 | 1 | uA |
| IDDIDLE | Supply current in Idle mode | RC oscillator enabled | - | 1.5 | - | uA |
| IDDST | Supply current in Standby mode | Crystal oscillator enabled | - | 1.6 | 1.8 | mA |
| IDDFS | Supply current in Synthesizer mode | FSRx | - | 5.8 | - | mA |
| IDDR | Supply current in Receive mode | *LnaBoost* Off, band 1<br>*LnaBoost* On, band 1<br>Bands 2&3 | -<br>-<br>- | 10.8<br>11.5<br>12.0 | -<br>-<br>- | mA |
| IDDT | Supply current in Transmit mode with impedance matching | RFOP = +20 dBm, on PA_BOOST<br>RFOP = +17 dBm, on PA_BOOST<br>RFOP = +13 dBm, on RFO_LF/HF pin<br>RFOP = + 7 dBm, on RFO_LF/HF pin | -<br>-<br>-<br>- | 120<br>87<br>29<br>20 | -<br>-<br>-<br>- | mA<br>mA<br>mA<br>mA |

**Figure 3.2: SX1276 LoRa chip datasheet**

The battery capacity consumed during a day by a device is estimated assuming 100 events detected and 10 frame transmissions performed by the device. The result is

222.66µAh, which corresponds to a lifetime of 21.5 months for a 150 mAh button cell battery, which is more than enough for a typical IoT battery.

## 3.2 The Firmware

The open source firmware of **disaster.radio** has been flashed into the boards after modifying it to suit our implementation, the firmware is written in C/C++ and the used platform is PlatformIO, shown in figure 3.3.

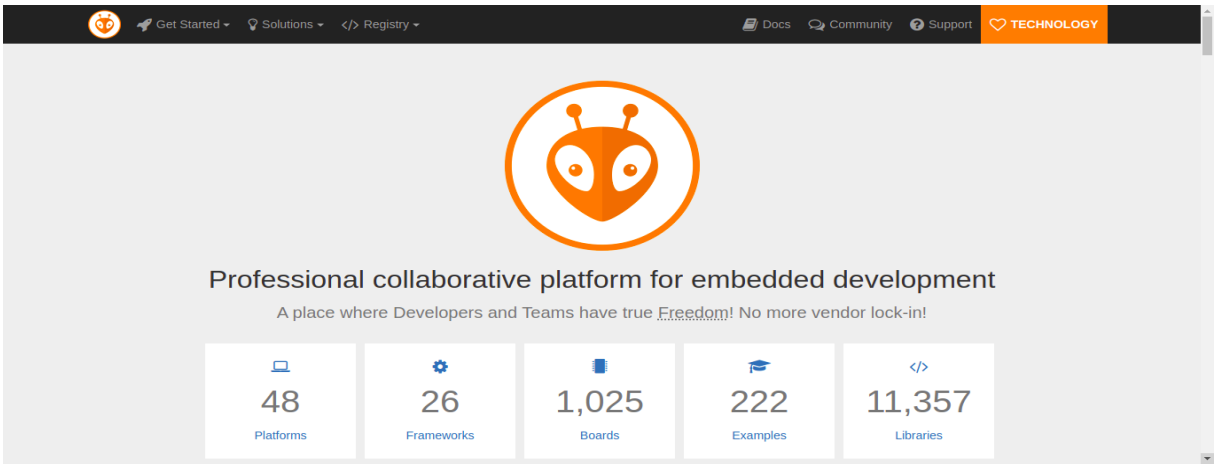



Figure 3.3: PlatformIO

The LoRa chip (SX1276) communicates with the ESP32 microcontroller via SPI. This protocol requires only four general purpose I/O (GPIO) pins, leaving the rest free for the addition of other sensors or devices. The ESP32 acts as a full web server that serves HTML web pages with CSS and JavaScript embedded over Wi-Fi. It also functions as a web sockets server that asynchronously transfers data between the microcontroller and the client web page.

To conceptualize the firmware used in our mesh network, a layered model based largely on the OSI model is used as shown in Figure 3.4.

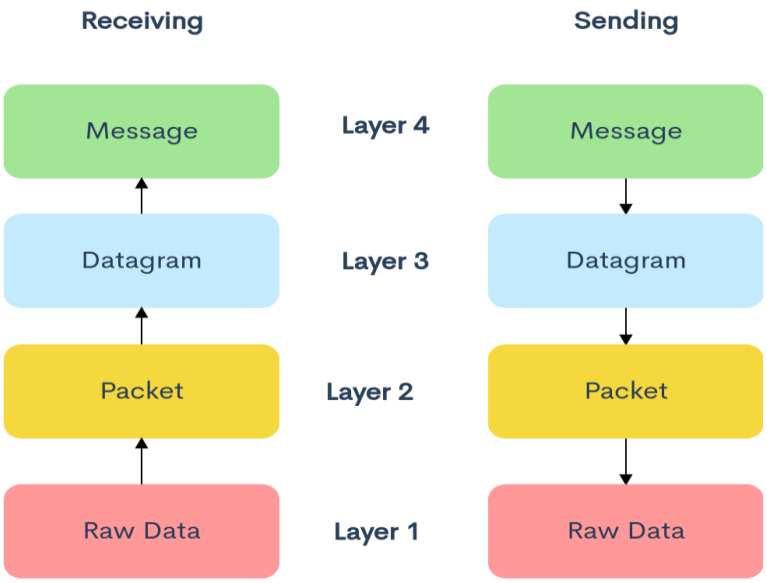



**Figure 3.4: Layered model for the firmware.**

### 3.2.1 Layer1

It is the LoRa physical layer, this is interfaced by the LoRa transceiver developed by Semtech, which handles the physical modulation and encoding of radio waves.

Layer 1 receives raw data from the LoRa transceiver and sends packets to Layer2 to be interpreted; this layer also receives packets from Layer2 and sends raw data to the LoRa transceiver.

### 3.2.2 Layer2

It is the routing layer; this is the LoRaLayer2 code which creates an ad-hoc packet switching network of nodes:

```
struct Packet {
    uint8_t ttl;
    uint8_t totalLength;
    uint8_t sender[ADDR_LENGTH];
    uint8_t receiver[ADDR_LENGTH];
    uint8_t sequence; // message count of packets tx by sender
    uint8_t source[ADDR_LENGTH];
    uint8_t hopCount; // start 0, incremented with each retransmit
    uint8_t metric; // of source-receiver link
    uint8_t datagram[239];
};
```

Layer 2 receives packets from Layer 1, interprets the header; then, either forwards the packet to Layer 3 to be consumed by a service/application or sends the packet back down to Layer 1 depending on whether or not it is the intended receipient.

Layer 2 also receives datagrams from Layer 3 along with information about the destination address, appends a header to the front, creates a packet and sends that packet to Layer 1.

### 3.2.3 Layer3

It is the transport layer; it has some similarities to UDP. It consists of what we have typically referred to as the "firmware". That is, an arduino sketch that joins any number of clients/services and allows them to communicate, e.g. it allows a telnet session to send a message over the LoRa transceiver.

Layer 3 receives datagrams from Layer 2, interprets the datagram header; and then sends the datagram to Layer 4. Layer 3 receives also messages from Layer 4 along with information about the destination address and type, using destination information to create a datagram that is sent to Layer 2, the structure of this datagram is as follows:

```
struct Datagram {
   uint8_t destination[ADDR_LENGTH]; // all zeros for loopback, all Fs for broadcast
   uint8_t type; // similar function to UDP port number
   uint8_t message[234];
};
```

### 3.2.4 Layer4

It is the application layer. This layer can be anything from a Serial console to a WebSocket application or a BLE-connected Android application. The message contained in the datagram (or the entire datagram itself) must be interpreted by the application. Any format can be used for the remaining 233 bytes as long as the intended recipient understands it (e.g. a type 'c' message intended for the chat app may be formatted differently from a type 'm' message intended mapping app). An application could choose to "listen" for specific types of datagram and ignore any datagram that is not relevant or cannot be interpreted.

### 3.2.5 Our contribution to the firmware

The web application offered by the firmware provides the sending and receiving of the messages in a general channel, where the users can see all the messages sent by every user in the network, to add the ability to send private messaging, we have modified the firmware, in layer4, the message that originally has 234 bytes will be segmented in the following way:

- The first 2 bytes: 1 byte for the senderID, and 1 byte for the senderUsername (unique to each client).

- The next 2 bytes: 1 byte for the ReceiverID, and 1 byte for the receiverUsername (unique to each client).

  The id of the clients is generated by the node to which they are connected to, with the following structure: **node mac address + unique integer.**

- The rest of the 230 bytes will be left for the actual message being sent, the new message is illustrated in Figure 3.5:

| 2 bytes | 2 bytes | 230 bytes |
|---|---|---|
| receiver ID & username | sender ID & username | Message |

**Figure 3.5: modified LoRa message format**

## 3.3 The web application

To match the new structure of the datagram, a new web application has been built to provide a private messaging functionality; this application acts as an interface between the end user and the Lora radio, and the flowchart indicated in Figure 3.6 summarizes its functionality.

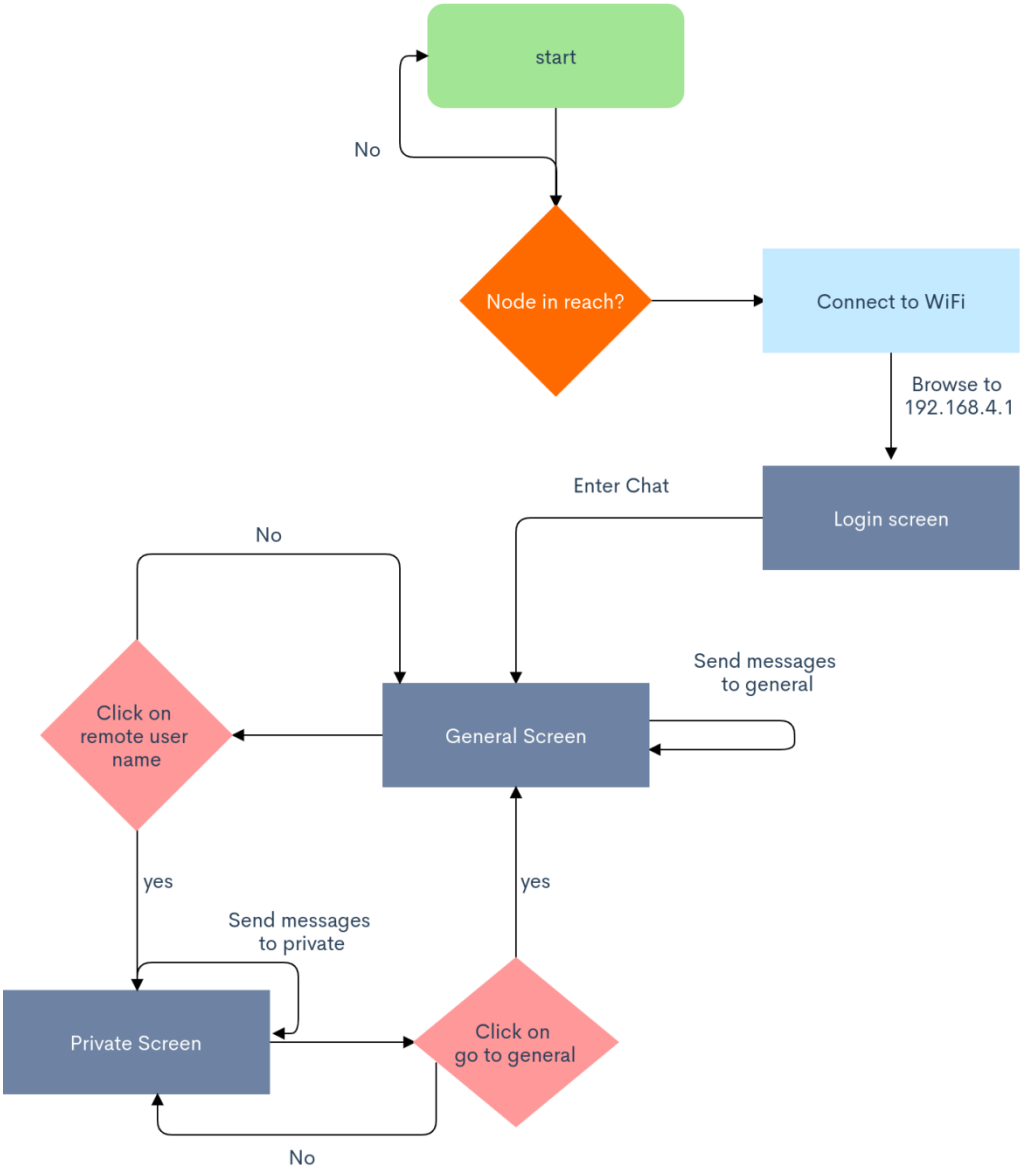



**Figure 3.6: Web application flowchart**

The web application is served over Wi-Fi, and can be accessed via the IP address 192.168.4.1, it currently consists of 3 different screens:

### 3.3.1 Login screen

The first is the login screen, which prompts the user for a username; this screen is shown in Figure 3.7:

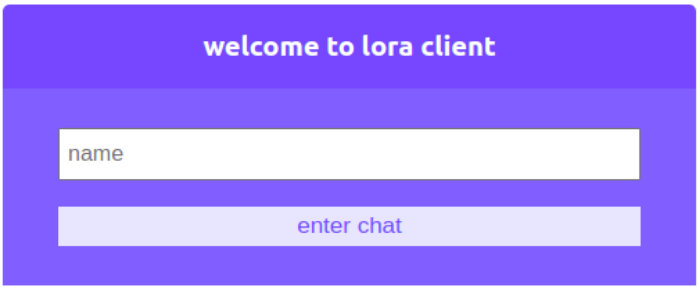



**Figure 3.7: The Login screen.**

Even without logging in, the web socket session is set up between the server run by the node and the user browser; which means that the server now is listening to any type of message. For instance, there are two types of messages, each identified by what is called a namespace. A message is formatted as follows:

**message = namespace + "|" + msg**

After the user enters his name and clicks on “enter chat”, a message is sent from the client to the server with a type of “join” message. The namespace for the “join” message is **j** sending, and the message will be as follows:

**message = “j” + “|” + user.name + “joined the channel”**

### 3.3.2 General chat screen

The second is the general chat, in which all users can participate in a conversation, with every message is seen by all users, this screen is shown in Figure 3.8 below:


**Figure 3.8: The general channel screen.**

The messages can be sent with type “chat”, this type has a namespace of **c,** for example:

**message = “c” + “|” + “hello there”**

The user messages which are referred to as “self messages” will appear at the right of the screen, while messages from other users referred to as “remote messages” will appear at the left of the screen with the name of the user sending the message.

### 3.3.3 Private chat screen

The third screen is the private chat, which is a conversation between two users, and only these two users can send and receive messages, this screen is shown in Figure 3.9.

By clicking on the name of the user sending the remote message, the user will be directed to the private screen where messages with the corresponding remote user can be exchanged privately. However, by clicking on the “go to general”, the user will be redirected back to the general channel.

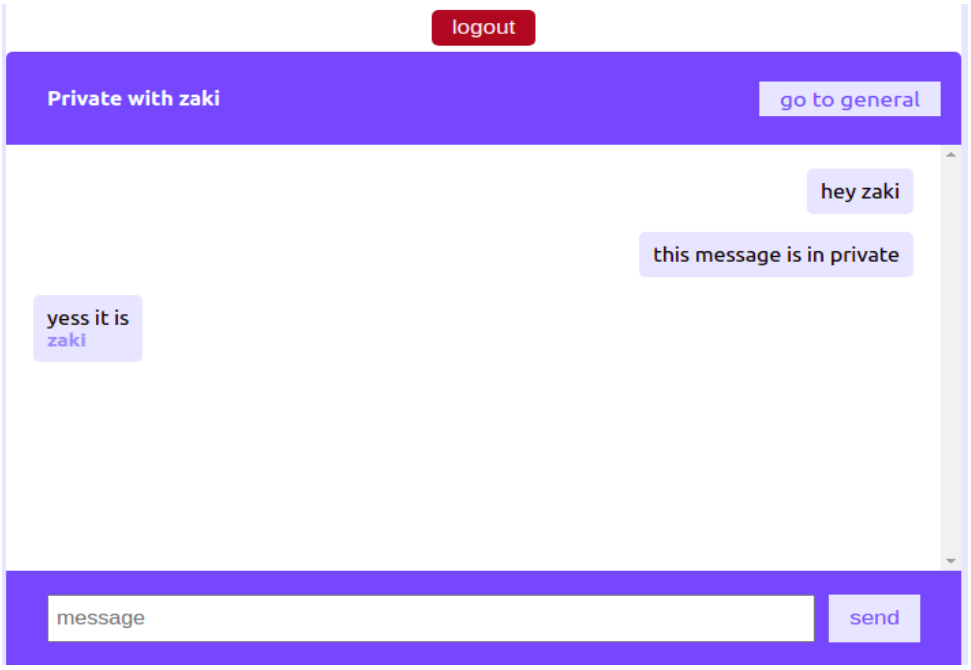



**Figure 3.9: The Private channel screen.**

The application is responsible for attaching the senderID and the receiverID to the message. The general room has been assigned a default ID of 0, which means when a user wants to send a message to the general channel, the receiverID will be 0 and everyone will receive the message.

## 3.4 Experimental Results and Analysis

In this part, we describe the experimental results and analysis. The experimental measurement is divided into three groups: the first and the second ones are the Time On Air (ToA) (delay) measurement experiment; the third one is the Packet Delivery Rate (PDR) measurement experiment.

### 3.4.1 Time On Air vs payload length

In this experiment, we tested the time that takes a packet to travel two hops away from the sender, who is located in Corso residence on building B, and the receiver is located in the same residence but in building G. The distance is estimated to be 150m, with concrete walls and obstacles along the way, as indicated in Figure 3.10, the spreading factor is fixed at 7. For this experiment we used four different Payload Lengths (PL) namely: PL= 10B, PL= 20B, PL= 40, PL= 80, the results of the experiment are shown in Table 3.2:

**Table 3.2: ToA for different payload lengths**

| Payload Length (Bytes) | Time On Air(ms) |
|---|---|
| PL= 10 | 61.7 |
| PL= 20 | 71.9 |
| PL= 40 | 102.7 |
| PL= 80 | 164.1 |


**Figure 3.10: Measured distance between Block B and Block G.**

It can be clearly seen that the time delay increases with the increase of PL length. The length of each packet should be reasonably set to obtain the optimal transmission results, as in a real world scenario, transmission time can be critical. In general, ToA is the time that the sender experiences from the time moment when the data are sent to the moment when it receives feedback packet from the receiver. Measurement software called Postman is used to record the sending and receiving time points and the ToA can be obtained by receiving time points minus sending time points.

### 3.4.1 Time On Air vs spreading factor

In the second experiment, we also tested the time delay it takes a packet of fixed size to travel from the sender to the receiver, who are in the same positions as the first experiment. The Spreading Factor (SF) varies from 7 to 12, and the payload length is set to PL= 50Bytes, the results of the experiment are shown in Table 3.3:

**Table 3.3: ToA for different spreading factors**

| Spreading Factor | Time On Air(ms) |
|---|---|
| 7 | 118.0 |
| 8 | 215.6 |
| 9 | 390.1 |
| 10 | 698.4 |
| 11 | 1,478.7 |
| 12 | 2,793.5 |

From this experiment, we notice that a small SF results in lower Time On air and a larger SF increases the time on air. This is because higher spreading factor increases energy consumption and reduces the data rate which is the speed at which data is transferred between the nodes; but, lower spreading factor decreases energy consumption, and increases the data rate.

### 3.4.1 PDR vs spreading factor:

This experiment aims to test the network coverage of LoRa. Tests were conducted in Boumerdes, two nodes were used in this experiment. The first node was set up in the Signals and Systems Laboratory located on the fourth floor of building B in IGEE. For the second node, two different test points were chosen: the first is located in Corso residence on building G, and the second is located in the science faculty of UMBB (INIM), inside the library. The distances are 3.22Km and 770m for the two experiments respectively, as shown in Figure 3.11 and Figure 3.12.

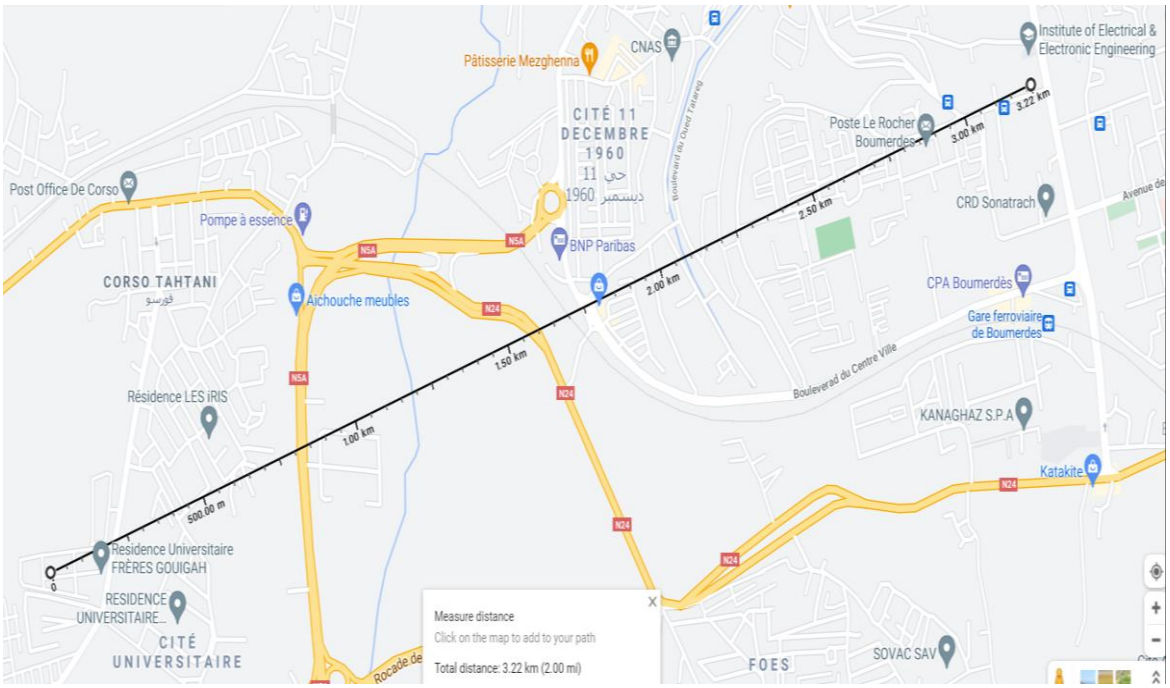



**Figure 3.11: Measured distance between IGEE and Block G of Corso residence**

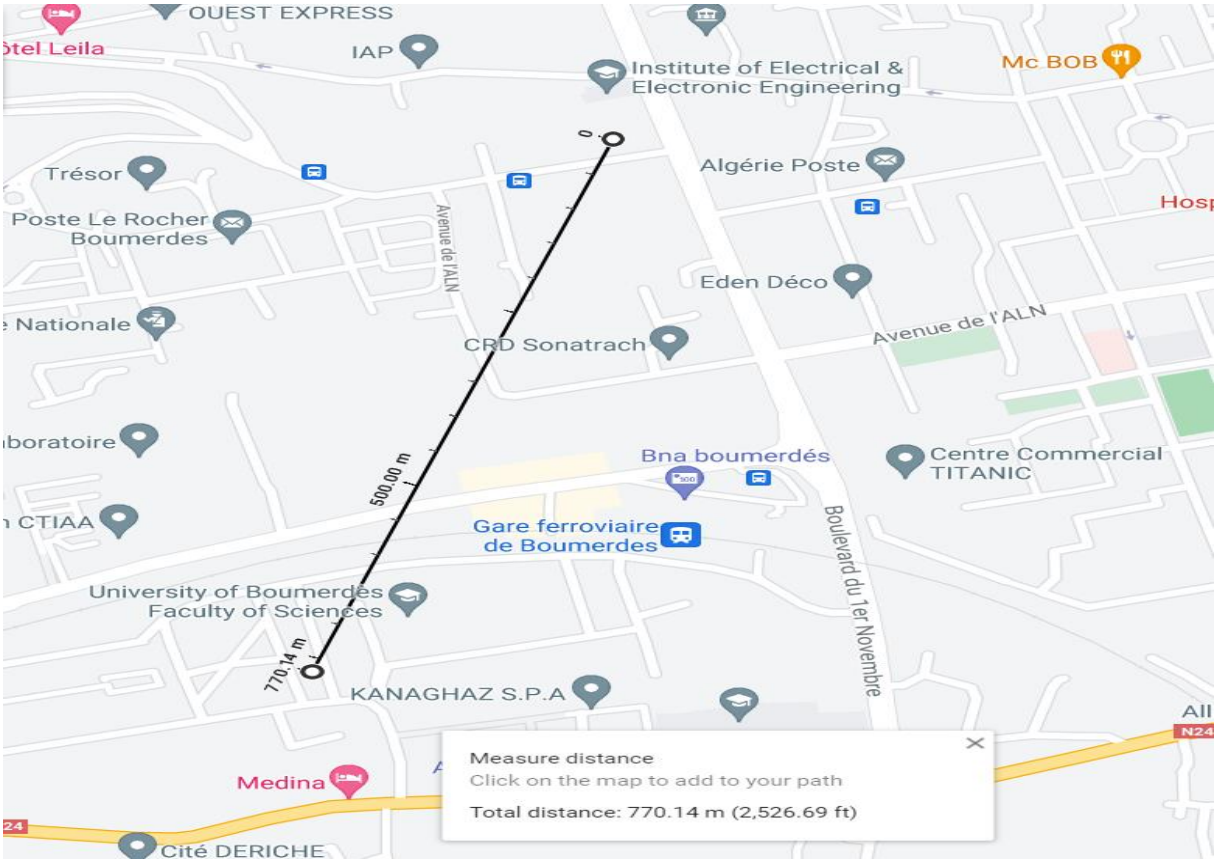



**Figure 3.12: Measured distance between 4th floor block B at IEEE and INIM library.**

The transmission power of each node was left unchanged, as the default value as specified by the manufacturer (14 dBm). To test the performance of different spreading factors (SF), the packet acknowledgment and retransmission was turned off. Spreading factors of 7, 9 and 12 were chosen for the tests, and after performing 10 tests, we were able to summarize the results in Table 3.4.

**Table 3.4: packet delivery ratio for different SF and distances**

| Distance | SF= 7 | SF= 9 | SF=12 |
|---|---|---|---|
| 770m | 80% | 90% | 100% |
| 3.22Km | 00% | 10% | 40% |

We notice that: the higher the spreading factors the better the coverage. For a spreading factor of 12, about 40% of packets were received at (3.22Km), while no packet was received when using a spreading factor of 7 or 9 for the same distance.

We can also observe that for SF = 12, the average PDR obtained becomes considerably higher than the two other spreading factors for both the long and short distances. Our hypothesis is that the capture effect (collision) is occurring and that it allows more packets to be decoded in case of collisions for SF 12 due to its robustness. However, the capture effect may be more limited when the power of the colliding packets increases.

The high delivery ratio using the high spreading factor has the cost of much lower bit rate; on the other hand, the network coverage with low spreading factors is much lower. If the SF parameter is increased the communication range is increased; meanwhile the data rate is decreased. So, the LoRa technology offers a compromise between data rate and communication range.

## 3.5 Conclusion

This chapter offered a detailed explanation of the operation of our LoRa mesh network. We have explored the operation of its three major components: the hardware, the firmware and the web application. Furthermore, we have conducted three experiments to evaluate the performance of our network, and provided a detailed explanation and discussion of the results.

# CH 4: Conclusions and future Work

Clearly, the work we have done in this master project opens a research opportunities with a lot of potential to study. In this final section, we summarize the conclusions and provide a high level overview of the future work.

It is important to note that this master project brings a comprehensive overview of previous works related to LoRa mesh networking, which may be considered as a very valuable resource.

This project discussed how LoRa devices can be used for chat messaging in disaster scenarios. The goal was to design a mesh network that allows users to send and receive text messages in areas that does not have cell coverage.

In this work, we have integrated the wireless mesh network topology with an IoT network where a set of LoRa nodes were made to route data between each other; the data is acquired from smart devices through a web application. The IoT system was developed around LoRa SX 1276 which is a high performance low power low cost computational chip integrated with ESP 32 Wi-Fi SoC. Each node was uploaded with an open source firmware, after modifying it to suite our implementation. Here the implementation was carried out on two nodes only. However, the number could increase to a significant level using this mesh network topology; as here, the nodes are mutually responsible for relaying each other's transmissions and these interconnected nodes result in a much larger coverage area.

After we discussed our network architecture, implimentation and operation, we have conducted some experiments to test the performance of our design. Two vital metrics, Time on Air (ToA) and Packet Delivery Rate (PDR), are considered for use in evaluating the reliability of LoRa signal propagation. ToA is an important parameter to measure the real-time performance of control system, and PDR can quantify the reliability of wireless communication system effectively. The experiments were conducted with two LoRa parameters which are the payload length (PL) and the spreading factor (SF). The experimental results of this study are helpful to the practical application of LoRa wireless mesh communication technology in disaster scenarios.

As an improvement to this work, and to ensure continuous evolution in the field of networking and IoT, more experiments can be performed to estimate the maximum capacity of the network. One option is to allow nodes to generate data more often and study when the network becomes saturated and how it thereby performs to set a maximum time to generate

application data. Another concern that has not been addressed is security. Due to lack of time and complexity of a good design, all security aspects have been avoided in this work, but it must be considered as a critical priority.

Finally, we can conclude that this work is not the end for our research, but rather the beginning of a very ambitious project which could end up manufacturing an IoT product for both the communication of IoT smart devices and people that live in rural areas.

## List of references

[1] “Mobile coverage: Qualitative research”, www.jigsaw-research.co.uk , september 2017

[2] “What Is LoRa?” https://www.semtech.com/lora/what-is-lora

[3] Internet of Things (IoT), Trend Micro Incorporated, 2021, https://www.trendmicro.com/vinfo/ph/security/definition/internet-of-things.

[4] BehrTech Blog, “6 Leading Types of IoT Wireless Tech and Their Best Use Cases”, 2020, https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/.

[5] Richard Dobbs, James Manyika and Jonathan Woetze, “The Internet Of Things: Mapping The Value Beyond The Hype”, June 2015, McKinsey Global Institute.

[6] Emmanuel Odunlade, “Introduction to LoRa and LoRaWAN: What is LoRa and How Does It Work? ”, CircuitDigest, May 2019.

[7] “What is LoRa?”, Semtech, 2021, https://www.semtech.com/lora/what-is-lora

[8] Shania Stewart et al., “Reducing the Cost of Implementing Filters in LoRa Devices”, Sensors 2019, 19, 4037.

[9] 3GLTEinfo, “LoRaWAN Frequency Bands”, 3GPP Internet of Things, 2021, https://www.3glteinfo.com/lora/lorawan-frequency-bands/.

[10] Juliusz Chroboczek, “Applicability of the Babel Routing Protocol”, https://dt-test.rjsparks.org/doc/rfc8965/, february 2016

[11] Antonio Cilfone,“Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies”, https://www.mdpi.com/1999-5903/11/4/99/htm, april 2019

[12] Antonio Cilfone, Luca Davoli , Laura Belli and Gianluigi Ferrari,“Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies”, https://www.mdpi.com/1999-5903/11/4/99/htm, april 2019

[13] Antonio Cilfone, Luca Davoli, Laura Belli and Gianluigi Ferrari, “Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies”, Future Internet 2019, 11, 99, mdpi.

[14] Juliusz Chroboczek, “The Babel Routing Protocol”, 2011. ⟨hal-00696369⟩.

[15] Nur-A-Alam et al., “Smart Monitoring and Controlling of Appliances Using LoRa Based IoT System”, Designs 2021, 5, 17, mdpi.