Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2016

Rúben Pedrosa Oliveira

Sensor Networks with Multiple Technologies: Short and Long Range

Redes de Sensores com Múltiplas Tecnologias: Curto e Longo Alcance

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2016

Rúben Pedrosa Oliveira

Sensor Networks with Multiple Technologies: Short and Long Range

Redes de Sensores com Múltiplas Tecnologias: Curto e Longo Alcance

"A creative man is motivated by the desire to achieve, not by the desire to beat others."
- Ayn Rand

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2016

Rúben Pedrosa Oliveira

Sensor Networks with Multiple Technologies: Short and Long Range

Redes de Sensores com Múltiplas Tecnologias: Curto e Longo Alcance

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica da Professora Doutora Susana Sargento, Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientacão científica do Doutor Lucas Guardalben, Investigador do Instituto de Telecomunicações de Aveiro.

o júri / the jury

presidente / president

Professor Doutor André Ventura da Cruz Marnoto Zúquete
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Ana Cristina Costa Aguiar
Professora Auxiliar Convidada do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto (Arguente)

Professora Doutora Susana Isabel Barreto de Miranda Sargento
Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (Orientadora)

agradecimentos / aknowledgments

A entrega desta dissertação marca o final de um percurso académico de 6 anos de altos e baixos, momentos bons e momentos menos bons, mas que no fim se revela bastante enriquecedor e de sucesso. Para tal contribuíram várias pessoas que desde o inicio, meio ou somente no final estiveram presentes.

Em primeiro lugar, queria agradecer aos meus Pais e Avós por todo o apoio e incentivo que me foi dado ao longo de todos estes anos, sem vocês não teria chegado aqui. Muito obrigado!

Quero agradecer à minha namorada, Joana, por todo o apoio e dedicação que me deu, não só durante esta dissertação mas também ao longo de todo o meu percurso académico. Obrigado por teres estado ao meu lado nos melhores e piores momentos, eu sei que nem sempre foi fácil, mas tu foste sempre capaz de te superar e continuar lá quando mais precisei. Obrigado por todo o carinho, motivação, reclamações, puxões de orelha e sobretudo por estares presente. A conclusão deste trabalho deve-se muito a ti. Obrigado, amo-te.

Um agradecimento especial ao Francisco e ao Pedro, por toda a ajuda e motivação dada ao longo deste último ano, revelaram-se verdadeiros companheiros e apoios. Um grande obrigado! Quero também agradecer ao André Barros, André Martins, Tiago e Marco por todo o companheirismo prestado. Obrigado malta!

Deixo também uma palavra de apresso a todo o grupo de investigação do NAP, em especial ao Carlos Ferreira pelo importante apoio no inicio do desenvolvimento desta dissertação e ao Nelson Capela pela boa disposição.

Agradeço à Professora Doutora Susana Sargento e ao Doutor Lucas Guardalben por me terem sugerido este tema e por toda a orientação e apoio dados que o levaram à conclusão deste trabalho. Foi um prazer trabalhar convosco.

Palavras-chave

Redes de Sensores, LoRa, Low-Power Wide Area Networks, Comunicações de longo alcance, WiFi, Internet of Things, Aquisição de Dados

Resumo

Low-Power Wide Area Networks (LPWANs) são um conjunto de tecnologias em crescimento na área da *Internet of Things (IoT)*. Devido às suas capacidades de comunicar a longo alcance e de baixo consumo energético, as *LPWANs* apresentam-se como a tecnologia ideal para o envio ocasional de pequenas porções de dados. Ao possuírem características únicas, as *LPWANs* podem ser usadas em diversas aplicações e em diferentes ambientes, sejam eles urbanos, rurais ou interiores. O trabalho desenvolvido nesta dissertação apresenta um estudo acerca da tecnologia *Long Range (LoRa)*, uma *LPWAN*, testando e avaliando o seu alcance, a qualidade do sinal e o desempenho na entrega de dados. Para isso, três cenários distintos são propostos e testados. A inclusão de LoRa numa plataforma de aquisição de dados com múltiplas tecnologias é um dos objectivos chave desta dissertação. Para isso, são propostas: (1) uma organização baseada em *clusters* de sensores; (2) um protocolo de controlo de acesso ao meio (MAC) para permitir que as comunicações através de LoRa sejam eficientes; e finalmente, (3) um gestor de conectividade com capacidade de gerir as diferentes tecnologias disponíveis nos sensores e que seja capaz de agir consoante o tipo de dados adquiridos. Os testes efectuados têm como objectivo perceber que tipo de parâmetros podem influenciar o desempenho global da solução proposta, bem como as vantagens de usar uma abordagem baseada em múltiplas tecnologias numa plataforma de aquisição de dados.

Keywords Sensor Networks, LoRa, Low-Power Wide Area Networks, Long Range Communications, WiFi, Internet of Things, Data Gathering

Abstract Low-Power Wide Area Networks (LPWANs) are one set of technologies that are growing in the field of the Internet of Things (IoT). Due to the long range capabilities and low energy consumption, Low-Power Wide Area Networks (LPWANs) are the ideal technologies to send small data occasionally. With their unique characteristics, LPWANs can be used in many applications and in different environments such as urban, rural and even indoor.

The work developed in this dissertation presents a study on the LPWAN LoRa technology, by testing and evaluate its range, signal quality properties and its performance in delivering data. For this, three distinct scenarios are proposed and tested.

The inclusion of LoRa in a multi-technology data gathering platform is the key objective of this dissertation. For this it is proposed: (1) an organization based in clusters of sensor nodes; (2) a Media Access Control (MAC) protocol to provide efficient communications through the LoRa technology; and finally, (3) a Connection Manager that is capable of managing the different available technologies in the sensor nodes and that is able to adapt its actions according to the acquired data type is proposed.

The performed tests aim to perceive which type of parameters can influence the performance of the overall proposed solution, as well as the advantages of a multi-technology approach in a data gathering platform.

Contents

List of Figures

List of Tables

List of Equations

Acronyms

3GPP	Third Generation Partnership Project
ACK	Acknoledgment
AP	Access Point
API	Application Programming Interface
BLAST	Bursty Light Asynchronous Stealth Transitive
BW	Bandwidth
CDMA	Code Division Multiple Access
CF	Carrier Frequency
CH	Cluster-Head
CR	Coding Rate
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSS	Chirp Spread Spectrum
CTS	Clear to Send
dB	decibel
dBm	decibel miliwatt
DBPSK	Differential Binary Phase Shift Keying
DCU	Data Collecting Unit
EC-GSM	Extended Coverage GSM

FEC	Forward Error Correction
GMSK	Gaussian Minimum Shift Keying
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, scientific and medical
IT	Instituto de Telecomunicações
LEACH	Low Energy Adaptive Clustering Hierarchy
LoRa	Long Range
LoS	Line of Sight
LPWAN	Low-Power Wide Area Network
LTE	Long Term Evolution
LTE-M	LTE for Machines
M2M	Machine-to-Machine
MAC	Media Access Control
MACA	Multiple Access with Collision Avoidance
MACAW	Multiple Access with Collision Avoidance for Wireless
NB-IoT	Narrow Band IoT
NLoS	Non Line of Sight
OSI	Open Systems Interconnection
PER	Packet Error Rate
PELR	Packet Error Loss Rate
PHY	Physical
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency

RPMA	Random Phase Multiple Access
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SF	Spreading Factor
SIFS	Short Interframe Space
SIG	Special Interest Group
SN	Sequence Number
SNR	Signal-to-noise Ratio
SSID	Service Set Identifier
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
ToA	Time on Air
UNB	Ultra Narrow Band
USB	Universal Serial Bus
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
WTS	Wait To Send

Chapter 1

Introduction

1.1 Motivation

The fundamental difference between *the Internet* and the *Internet of Things (IoT)* can be described by *less of everything* for the IoT. This means that, the available resources in a given devices or network are much more limited in terms of memory, processing power, bandwidth and available energy. This is either due to the fact that *things* are battery powered and their lifetime is a key priority, or due to the expected exponential growth of connected things that are predicted to be dozens of billion devices in the incoming years.

Figure 1.1: The Internet of Things [1]

To fulfill the specific requirements of the IoT platforms, a new range of protocols and technologies has emerged: Low-Power Wide Area Networks (LPWANs). Colloquially speaking, the LPWANs aim to be to the IoT what Wireless Fidelity (WiFi) is to the consumer networking: with a large area radio coverage provided by the base stations, transmitting power, modulation techniques, usage of unlicensed Industrial, scientific and medical (ISM) bands, they have the objective to allow that the end-devices incur in a very low power consumption when they are connected.

The LPWAN technologies present interesting trade-offs between range, power consumption and cost, which make them a powerful technology for network implementations in diverse types of environments, from the most harsh urban conditions to a rural and agricultural environment.

LoRa is one of most relevant LPWAN technologies due to its unique modulation, which makes it a very versatile technology that can adapt to different type of environments and applications. Also, its use of unlicensed bands makes it an attractive solution for the IoT and Machine-to-Machine (M2M) platforms. Its low power associated to the long range communications pushes LoRa to the top of the LPWAN technologies.

The study and comprehension of the LoRa technology is a key objective of this work, as well as its inclusion in a data gathering solution, either by assuming the main technology role or a complement to the already existing technologies. With the diversity of the wireless technologies and sensing information, it is important to endow the data collection units that integrate several IoT devices with different technologies and the ability to choose the best technology for each type of data, simultaneously transmitting information through the available technologies and networks, in a coordinated and dynamic approach. This is the overall dynamic scenario envisioned in this Dissertation.

1.2 Objectives

The objective of this dissertation aims to provide a solution that is capable of handling different technologies in order to forward environmental sensor data from the sensor sets to a fixed server, as well as snapshots from cameras installed in the data collection units. With the goal defined and the challenges pointed in mind, the present dissertation has the following objectives:

- Study of the rising LPWAN technologies available in order to understand its strengths and weaknesses and how they could be integrated to be part of the solution;
- Perform range and quality tests to the LoRa technology;
- Implement a data gathering protocol that can use different communication technologies, both long and short range;
- Use a cluster based organization approach to organize the nodes that compose the network to turn it more scalable and reliable;
- Create a Media Access Control (MAC) protocol for the LoRa technology;
- Create a Connection Manager that is capable of deciding which is the best technology to forward the gathered data to the server;

- Evaluate the functionality and overall performance of the developed solution in real environments.

1.3 Contributions

The work developed in this dissertation led to the following contributions:

- Conclusions about range and quality of the long range technologies based on different environmental/conditions tests;
- A multi-technology approach that is capable of adapt to different environmental situations;
- A modular solution that includes the cluster formation, the Connection Manager and the MAC layer well defined;
- A feasible data gathering solution that can handle the information to the end-user with low latency and reliability.

This work is targeting two papers in preparation: one that presents the performance of the long range technologies in different scenarios, and one that presents the proposed multi-technology gathering solution.

1.4 Document Organization

This document is organized as follows:

- **Chapter 1** contains the dissertation's motivation, context and objectives;
- **Chapter 2** presents the state of the art about Low-Power Wide Area Networks (LPWANs), in which it is included Long Range (LoRa), among other existent solutions, and also on the related work, which comprises the Media Access Control (MAC) and clustering solutions in Wireless Sensor Networks (WSNs), as well as the existing Smart Cities testbeds. A study on the LoRa range capabilities is also presented;
- **Chapter 3** presents the proposed solution and the overall architecture as well as the modules specification;
- **Chapter 4** provides a more detailed explanation of the modules including technical concepts in this point;
- **Chapter 5** shows the results of the evaluation of the implemented solution;
- **Chapter 6** summarizes the work performed in this dissertation, the main conclusions and suggests possible future improvements to the work done.

Chapter 2

State of the art

2.1 Introduction

This chapter is focused on providing the reader an overview of fundamental concepts required to understand the work presented in this dissertation and also to present related work on the main topics.

Section 2.2 presents an overview on the existing LPWAN technologies, including its features, limitations and advantages. It is given especial focus on the LoRa technology, since it is the adopted technology to develop the work in this dissertation.

Section 2.3 presents the related work in the topics that contributed to develop the final proposed solution.

2.2 Low-Power Wide Area Networks (LPWANs)

2.2.1 Overview

The Internet of Things (IoT) [2] paradigm covers the possibility of any device that possesses any kind of radio link and with the capability of transferring data, to be interconnected or connected to the Internet. It proposes to revolutionize the way we live and work, by improving the efficiency of the natural, human and energy management, and it can also optimize the production processes. As a consequence, multiple independent researches have forecasted a unbridled growth in revenue and volume of IoT and M2M industry in the upcoming years.

It is expected that the number of connected consumer electronics and M2M devices will exceed the number of human subscribers that use mobile phones, personal computers, tablets and laptops by the year of 2020 [3]. This leads to an expected total device connections to be between 26 and 28 billions by 2020 [3] [4] [5].

There are many factors that need to be considered for IoT (Figure 2.1); however, no single technology will be able to solve all factors simultaneously [6].

Traditional technologies, such as short-range wireless networks e.g., ZigBee, Bluetooth,

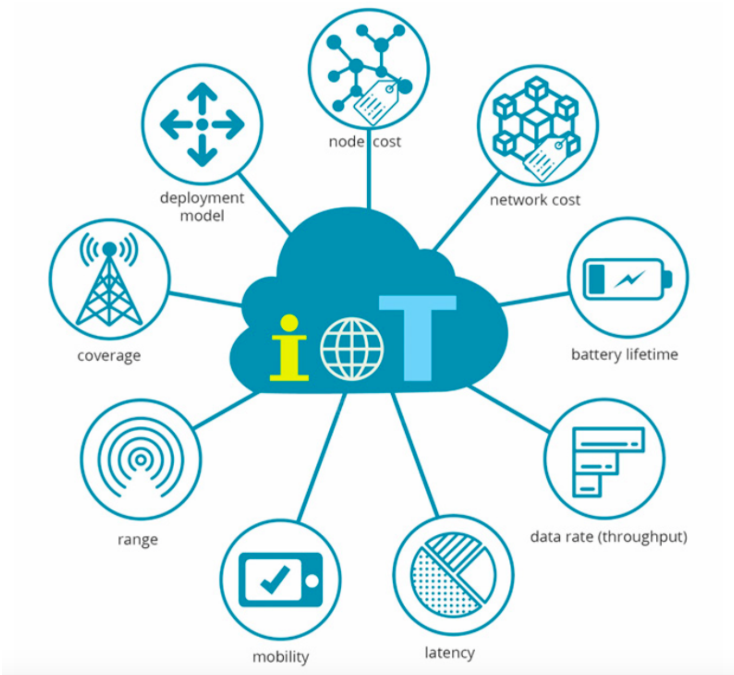


Figure 2.1: IoT Application Considerations, based on [6]

Z-Wave, Wireless Local Area Networks (WLANs) e.g., WiFi [7], and cellular networks e.g. Global System for Mobile Communications (GSM) and Long Term Evolution (LTE) have been the prevalent technologies in IoT. Even though they allow the wireless communication of the IoT devices in the network, they present their weaknesses in one or many points presented in Figure 2.1. They are usually of high complexity, low reliability approaches, as well as high cost and high energy consumption. As a result, another range of protocols and technologies have arisen with the promise to complement the existing cellular and short range wireless technologies in addressing diverse requirements of IoT applications. Low-Power Wide Area Networks (LPWANs) offer unique sets of features including wide-area coverage for low power and low data rate communications; it also proposes to reduce operational cost of traditional cellular networks.

This type of networks is characterized by exploiting the sub-GHz unlicensed, ISM frequency band, and by sporadically transmitting small packets at low data rates. LPWANs are considered exceptional candidates for IoT applications, since they provide different trade-offs compared to the traditional technologies. Figure 2.2 highlights these differences. Most of LPWANs applications will be new ones, since they will be connecting devices for which no suitable connectivity solution existed [8]. Figure 2.3 shows the variety of applications in many business sectors on which LPWANs can be used to connect devices.

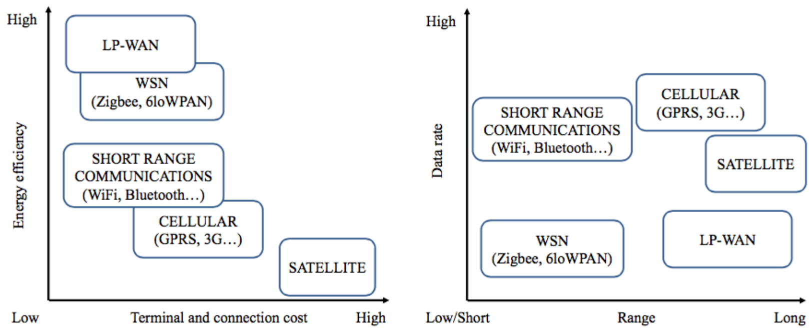


Figure 2.2: LPWAN vs. legacy wireless technologies, based on [9]

Figure 2.3: LPWANs applications in several sectors, based on [10]

2.2.2 Common features

Plenty of applications are envisioned for IoT, such as smart cities, home automation, wearable electronics, environmental monitoring, logistics, smart metering and alerting. To successfully support these IoT applications, some key requirements are demanded for the LPWANs [8] [10] [11] [12]:

Low Power Consumption Most of IoT applications require long battery life devices, once that in most IoT deployments, due to the accessibility restrictions, manually batteries replacing leads to an enormous logistical expenses. Typically, LPWANs are supposed to sporadically transmit small size packets, also, they have simplified protocol stacks and improved sensitivities, which lead to a low power consumption. Margelis et. al [13], debate the suitability of low throughput networks in diverse applications.

Extended Coverage LPWAN technologies are designed for large coverage ranges and deeper indoor coverage by enhancing link budget for 15-20 dB. This permits the end-devices to be able to connect to the base station at a distance that could be few meters or tens of kilometers depending on the deployment environment (urban, rural, etc.). It also turns possible to communicate with devices that could be located in basements, behind concrete walls or elevators.

Low device and deployment cost In order to permit a reliable and profitable business case for IoT, the devices cost should be extremely low, since the commercial success of these networks is tied to connecting a large number of end-devices, while keeping the cost of hardware below 5$ [3].

Scalability Due to the exponential increase of the IoT devices, the support for massive number of connected devices and incoming traffic volume is one key requirement for the LPWANs technologies.

2.2.3 Solutions

Contrary to the general tendency that leads the new generation of wireless technologies towards higher frequency bands, LPWANs technologies mainly use the band from 863 MHz to 870 MHz in Europe (so-called SDR860) [14] [15]. Depending on the band adopted by the different technologies, they can be commonly divided in two categories:

- **Ultra Narrow Band (UNB)**: Using narrowband channels with a bandwidth of the order of 25 kHz;
- **Wideband**: It uses a larger bandwidth (125 kHz or 250 kHz) and employs some form of spread spectrum multiple access techniques to hold multiple users in one channel.

In the following it is presented an overview of some of the most prominent LPWAN platforms so far, deferring to Sub-Section 2.2.4 the description of LoRa technology.

Sigfox The first LPWAN technology and the one in the most advanced deployment state in Europe is Sigfox [16], which was founded in 2009 in France. Sigfox claims to have covered most of the area of Spain, France and Russia, among others. Sigfox employs a proprietary UNB (100 Hz) modulation (Differential Binary Phase Shift Keying (DBPSK)) in the sub-GHz ISM band carrier. By using UNB, Sigfox promotes bandwidth efficiency and experiences very low noise levels. This results in a high receiver sensitivity and a very low power consumption. However, this benefit comes at a price, with Sigfox only achieving a maximum throughput of 100 bps. Further, Sigfox initially supported only uplink communications, having evolved later to support downlink communications also, although with a significant link asymmetry. Packets in Sigfox are limited to a payload of 12 bytes in uplink and 8 bytes in downlink, with a maximum of 140 messages that an end-device can send and only 4 messages allowed for the gateway to send. Sigfox claims that each base station can handle up to a million connected devices, with a coverage area of 30-50 km in rural areas and 3-10 km in urban area.

Weightless Weightless Special Interest Group (SIG) [17] proposed three open LPWAN standards, each one with its own particularities in terms of features, range and power consumption. The three standards work in the sub-GHz bands and can operate in license-free as well as in licensed spectrum.

Weightless-W The original Weightless-W standard is a system with star topology operating in TV white spaces spectrum and supports several modulation schemes, spreading factors and packet sizes. Depending on the link budget, it claims to achieve two-way communication rate between 1 kbps and 10 Mbps with very low overhead. The edge-nodes communication to the base station can be performed along five kilometers depending on the environmental conditions. Since the shared access to the TV white spaces is not available in most regions, two alternative standards using the ISM band were defined.

Weightless-N Supports a star network architecture and uses a class of low-cost technology such as the one used by Sigfox. Thereby, UNB (DBPSK) modulation is used in the sub-GHz spectrum, with an excellent range that can achieve several kilometers even in harsh urban environments. Only one-way communication with a throughput of 100 bps in provided.

Weightless-P This version puts together the most appropriate attributes of the previous standards and claims to be essentially focused on the industrial sector. It provides two-way communication and modulates the signal using Gaussian Minimum Shift Keying (GMSK) and Quadrature Phase Shift Keying (QPSK), two known schemes already used in different commercial products, which enables the use of non-proprietary chipsets. Weightless-P uses narrowband channels of 12.5 kHz in both ISM and licensed spectrum, with an adaptive data rate in the range between 200 bps and 100 kbps. It provides full support for valuable characteristics such as, acknowledgments, auto-retransmissions and channel coding, among others.

Ingenu Formerly known as On-Ramp Wireless, Ingenu [18], developed and owns the rights of the patented physical access scheme named as Random Phase Multiple Access (RPMA) [19]. Ingenu has been pioneering the standardization of the physical layer specifications under IEEE 802.15.4k standard [20]. Conversely to the other LPWAN implementations, Ingenu uses the 2.4 GHz band, although, due to its robust physical layer design, it can still operate over long-range distances and under challenging Radio Frequency (RF) environments. RPMA is reported to achieve a receiver sensitivity of about -142 dBm and 168 dB link budget [21].

DASH7 The DASH7 [22] is a full Open Systems Interconnection (OSI) stack specification by the DASH7 Alliance that operates in the sub-GHz ISM band and uses the Bursty

Light Asynchronous Stealth Transitive (BLAST) technology. The main advantage provided by DASH7 protocol is that it enables both star and tree topologies to facilitate the management of large networks. DASH7 can achieve a data rate up to 167 kbps. However, only some pilot projects have been developed so far [23].

As an alternative, the Third Generation Partnership Project (3GPP) is addressing the M2M and IoT market by evolving its existing cellular standards to reduce complexity and cost, improve the range and signal penetration and extend battery lifetime. Using its diverse licensed solutions such as LTE for Machines (LTE-M), Narrow Band IoT (NB-IoT) and Extended Coverage GSM (EC-GSM) offer distinct trade-offs between cost, coverage, power consumption and data rate to address several needs of the M2M and IoT applications. However, a common objective of these standards is to be able to maximize the re-use of the existing cellular infrastructures and owned radio spectrum.

A more detailed comparative study between the referred technologies can be found in [9], [10], [11], [24] and [25].

2.2.4 Long Range (LoRa)

LoRa technology was first proposed by Semtech and is actually developed by the LoRa Alliance [26]. This system is assumed to be usable in battery-powered devices that require a long life time, thus low energy consumption is a major requirement. LoRa can be associated to two different layers: a Physical (PHY) layer using Chirp Spread Spectrum (CSS) radio modulation technique [27]; and a MAC layer protocol defined as LoRaWAN, although the global communication system also requires a specific access network architecture.

2.2.4.1 LoRa Physical Layer

LoRa physical layer modulates the signals in the sub-GHz ISM band using a proprietary spread spectrum technique [28]. A bi-directional communication is achieved by a special CSS technique, which spreads the narrow band signal over a wider channel bandwidth. The resulting signal presents noise like characteristics, which makes it more difficult to detect or jam. The processing gain provides resilience to noise and interference.

The transmitter makes the chirp signals vary their frequency over time without changing their phase between adjacent symbols in order to encode information (Figure 2.4). Because the linearity of the chirp pulses, any frequency deviation between the receiver and the transmitter are equivalent to timing offsets, easily eliminated in the decoder. Distant receivers can decode a highly attenuated signal several decibels (dBs) below noise floor.

LoRa supports diverse Spreading Factors (SFs) that provide different trade-offs between range and data rate. Higher SFs provides long range in exchange of a lower data rate and vice versa. Also, LoRa implements Forward Error Correction (FEC) along with the spread spectrum technique to further increase the receiver sensitivity. The data rate ranges from 300 bps to 38.4 kbps depending on the SF and channel Bandwidth (BW).

A LoRa radio has four configuration parameters: carrier frequency; spreading factor; bandwidth; and coding rate. The combination of these parameters provide different energy

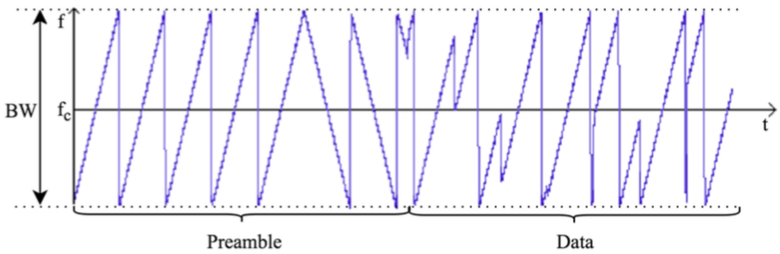


Figure 2.4: Frequency variation over time of an emitted signal by LoRa, based on [29]

consumption, transmission range and resilience to noise. In the following the SX1272 module [30] is used as reference.

Carrier Frequency (CF) The CF is the center frequency used for the transmission band. For the SX1272 it is in the range of the 860 MHz to 1020 MHz.

Spreading Factor (SF) The SF parameter expresses the ratio between the symbol rate and chip rate. A higher SF increases the Signal-to-noise Ratio (SNR), as well as the sensitivity and range, but also increases the Time on Air (ToA) of the packet. The number of chips per symbol is obtained as 2^{SF}. As an example, with an SF of 10 1024 chips/symbol are used. Each increment in the SF halves the transmission rate and as a consequence doubles the ToA and also increase the energy consumption. The SF value can be chosen from a range of 7 to 12.

Bandwidth (BW) BW represents the range of frequencies in the transmission band. Higher BW values give a higher throughput (thus a lower ToA), but a lower sensitivity (due to integration of additional noise) and vice versa. Data is sent at a chip rate equal to the BW. Thus, a BW 125 kHz corresponds to a chip rate of 125 kcps. The possible values for the BW are 125 kHz, 250 kHz and 500 kHz.

Coding Rate (CR) To perform the FEC a CR has to be defined, it offers protection against burst of interference. A higher CR value gives extra protection, although it also increases the ToA. Radios with different CR values but that maintain the same SF/BW/CF are able to communicate. The CR of the payload is stored in the packet header, which is always encoded at 4/8. The CR is equal to $\frac{4}{4+n}$ with n between 1 and 4.

Equation 2.1 allows to compute the useful bit rate (R_b) of a LoRa transmission, taking into account the parameters previously described [31].

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \tag{2.1}$$

Even thought LoRa modulation can be used to transmit arbitrary data, a physical frame format is defined by Semtech and used in their manufactured chipsets. The BW and SF are constant for a frame. Figure 2.5 presents the structure of a LoRa frame.

A LoRa frame starts with a preamble. The preamble is used to sync the receiver with the transmitter at the begining of a transmission.

After the preamble, there is an optional header that carries the information about the LoRa configuration and the size of the payload. The optionality of the header is used so it is possible to disable it when the configurations and payload size are known in advance. As it was stated before, the header is always encoded with a 4/8 CR.

The frame payload is sent after the header, and an optional Cyclic Redundancy Check (CRC) is sent at the end of the frame.

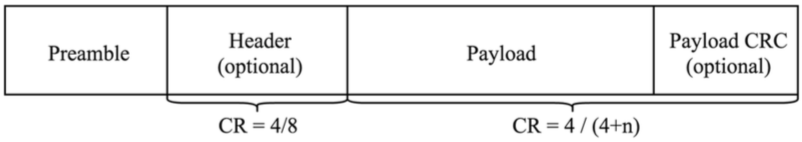


Figure 2.5: Structure of a LoRa frame

The LoRa radios present some interesting characteristics. Transmissions on the same CF, but with different SF, are orthogonal, which enables the division of the channel into virtual sub-channels. When transmissions occur at the same time with the same configuration parameters, the strongest signal has a higher probability of being received. This is possible due to concurrent transmissions being non-destructive even when their contents are different.

2.2.5 LoRaWAN Protocol

LoRaWAN [32] is a MAC protocol that was built to use the LoRa PHY layer. Mainly designed for sensor networks, wherein sensors exchange data occasionally and at a low data rate. In Figure 2.6 is shown the LoRaWAN network architecture.

The LoRaWAN system requires three main components:

- **End-devices** are low-power sensors/actuators connected via LoRa radio interface to one or more gateways;
- **Gateways** act as concentrators that forward packets coming from end-devices to a network server through an IP backhaul interface, which provides a higher throughput, such as 3G or Ethernet. It is possible that multiple gateways coexist in the same LoRa deployment, which could lead that different gateways receive the same packet;
- **NetServer** is the entity that controls the whole network, including radio resource management, admission control, security, etc.

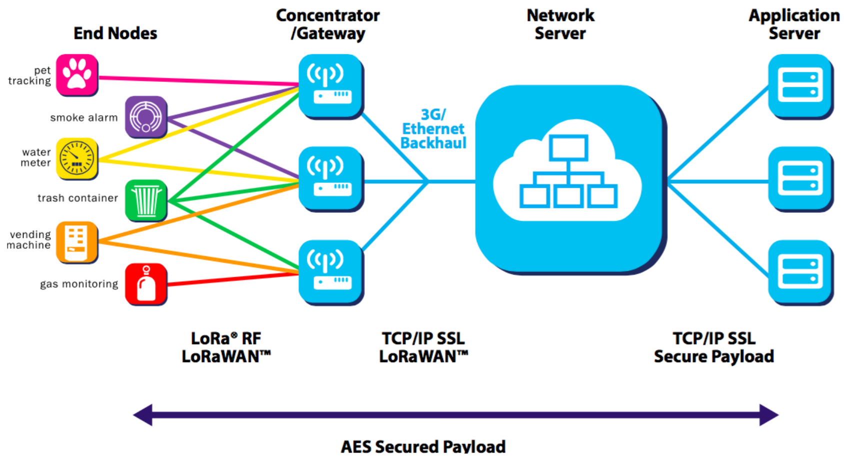


Figure 2.6: LoRaWAN network structure, based on [33]

As it is presented in Figure 2.6, the LoRaWAN architecture relies on a star-of-stars topology, where end-devices transmit their packets via single-hop communication to one or more gateways, that, in turn, are connected to a common NetServer. The gateways serve simply as a link layer relay and are hence totally transparent to the end-devices, which are logically directly connected to the NetServer.

A distinguishing property of LoRaWAN is its classification of end-devices, each associated to a distinct operating mode [32]. It defines three different classes, all of which support bi-directional communication, but with different downlink latency and power requirements.

Class A (for All) devices achieve the longest battery lifetime, but with the highest latency, since it listens for a downlink communication only shortly after its uplink communication. Networks of this class are mainly used for monitoring applications, where data produced by the end-devices have to be collected by a control station.

Class B (for Beacon) devices, in addition, can schedule downlink receptions from the base station at defined time intervals. Thus, only at these certain times, applications can send control messages to the end-devices. A synchronized beacon from the gateway is required in order that the NetServer knows when the end-devices are in the listening windows. Class B is used for applications where the end-devices need to receive commands from a remote controller, or need to provide data at user's request.

Class C (for Continuously listening) is defined for end-devices that do not have energy consumption restrictions, which permits that the reception window is always open.

At the MAC layer, LoRaWAN applies a simple ALOHA scheme that in combination with LoRa physical layer enables multiple devices to communicate at the same time but using different channels and/or orthogonal SFs.

2.3 Related Work

2.3.1 LoRa Coverage Studies

One of the most addressed issues of debate about LPWANs technologies is its effective coverage range.

Aref et. al [34] was one of the first to publish results about range measurements with LoRa technology. There were performed range tests in the region of Offenburg, Germany, where a module was put at a height of 20 meters and various points were tested. The authors configured the module with a 250 kHz BW, a SF of 10 and a CR of 4/6, achieving a maximum range of about 8 km for different packet sizes.

Petäjäjärvi et. al [35] proposes a channel attenuation model for LoRa. With range tests in the region of Oulo, Finland, packets were received in a ground area between 10 and 15 km of distance, although, with a packet loss ratio of 74%. LoRa was tested also in a boat, achieving communication in a range between 15 and 30 km with a packet loss ratio of 38%. The tests were done with the configuration that permits a higher range, SF 12 and BW 125 kHz.

Centenaro et. al [15] made a coverage test in the region of Padova, Italy, in order to define how many gateways would be necessary to cover the 100 square kilometers of the municipality. With a conservative coverage range of 1.2 km, the authors defined that a total of 30 gateways would be enough to cover all area, which they claim to be half the number of sites deployed by one of the major cellular operators in Italy to provide mobile cellular access over the same area.

Trasviña-Moreno et. al [36] tested the difference between transmitting in the 433 MHz or 868 MHz band. The tests were performed in Zaragoza, with Line of Sight (LoS) conditions, from a fixed distance of 7 km and an elevation difference of 300 meters. Different SF values were tested, as well as different BWs. The authors showed from the analysis of the Received Signal Strength Indicator (RSSI), Packet Error Rate (PER) and Packet Error Loss Rate (PELR), that transmissions in the 433 MHz band present a better RSSI and low BWs present high PER and PELR.

Augustin et. al [29] conducted coverage tests in a suburb in Paris with mainly low-rise residential dwellings. There were tested 5 points with increasing distance and different SF configurations. A maximum range of 3400 meters was achieved with the highest SF value and presented a packet delivery of nearly 40% at these conditions.

Following the work developed in [35], Petäjäjärvi et. al made experimental tests to study how LoRa suits for Non Line of Sight (NLoS) indoor operation in general and the human wellbeing in particular [37]. The tests were conducted in the campus of the University of Oulu, Finland, with a device that was attached to a researcher's arm and sending data from defined points to the base station located outside the building at a height of 24 meters above sea level. The authors show that LoRa can handle indoor environments quite well with a total success ratio of 96.7%. A mobility test was also developed showing 95% of success ratio.

Even though recently an increasing number of studies in the coverage of the LoRa

LPWAN technology have been published, none of them present a comparative study between a rural and urban scenario. Also, different urban scenarios with distinct characteristics (eg. plane and uneven terrain, number of obstacles in the signal path) that can affect the LoRa performance are not yet totally explored.

2.3.2 Data Gathering

An example of an ongoing real smart city implementation is SmartSantander (Spain) [38]. It proposes a unique city-scale experimental research facility in support of typical applications and services for a smart city. This facility includes a large number of IoT devices deployed in several urban scenarios. With a close concept to the SmartSantander is the Oulo Smart City [39] in Sweden, with diverse outdoor sensors, and CitySense [40] that comprises sensors mounted on buildings and streetlights across the city of Cambridge (USA).

UrbanSense [41] is a platform deployed on the city of Porto that aims to collect key environmental data, such as air quality parameters, solar radiation, noise, rain and wind speed and direction. The UrbanSense platform present some particular characteristics: it is designed for affordability and extensibility; it uses different possibilities to send data to the cloud, including both real-time and delay-tolerant communications; and, it uses IoT integration to display the data streams to smart city tools and applications.

Some platforms that resemble to the UrbanSense, although only monitoring air quality, are Clairity (MIT campus) [42], OpenSense (Zurich, Switzerland) [43], and EcoSensor (Valencia, Spain) [44].

Regarding large-scale platforms capable of providing diverse integrated data services for smart cities, we highlight City of Things [45] (Antwerp, Belgium), with a city-wide open testbed infrastructure that allows performing experiments at the network, data and user levels, while supporting a cross-technology features, that includes, LoRa, DASH7, WiFi, IEEE 802.15.4, among others.

LoRa FABIAN [46] is a Network Protocol Stack and experimental network setup, deployed in Rennes, France, for IoT needs. Although being mostly designed for LoRa and the associated constraints it can be reused on top of any Layer 1 technology. It aims at democratizing access to low-power long-range technologies, by abstracting the network complexity using common Internet protocols such as CoAP, DNS, HTTP(s), among others. LoRa FABIAN uses different components that are necessary to test and provide coverage for IoT applications, this include both communicating and gathering data from IoT devices, and connecting them to the Internet (both to send data, and receive remote commands).

Habitat and environmental monitoring are another possible application of WSN and eventually LPWANs. Mainwaring et. al [47] applied a WSN to real-world habitat monitoring, that consists of seabird nesting environment and behavior monitoring, deploying 32 nodes on a small island off the coast of Maine that streams useful live data onto the web. Barrenetxea et. al [48] studied the requirements of a reliable environmental monitoring network, including practical experiments in harsh environments such as the Swiss mountains. Nadini et. al [49] developed a network for monitoring and classifying animal behavior using ZigBee that was applied in Bramstrup on Fyn Island in Denmark. Toldov et. al [50] evaluates the performance of LoRa when applied to the existing PREDNET wildlife animal tracking project.

One objective of the work of this dissertation is to be able to extend the available communication technologies in the UrbanSense platform.

2.3.3 Media Access Control (MAC)

Media Access Control (MAC) is an essential technique that enables the successful operation of the network in shared-medium conditions. There are several MAC protocols designed for wireless voice and data communication networks. Typical examples include Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA) and contention-based protocols like IEEE 802.11. WSNs and IoT applications require the fulfillment of some specifications in terms of the MAC protocol, such as multi-hop communication, resilience and sometimes low-latency, among others [51].

A vast number of protocols exist to implement these requirements [52] [53], however, these options are not designed taking into account the specific features of LoRa, such as possible high time-on-air of the packets and the ability to receive one message out of a pool of concurrent transmissions [54].

Most of WSNs MAC implementations are focused more in reducing the power consumption than in other aspects, such as latency or delivery ratio. Since the power issues are not addressed in this dissertation, there is not a protocol that is appropriate for what it is aim in the coordination of LoRa transmissions.

The MAC protocols for WSNs can be categorized in two types: Schedule based and Contention based. On the one hand, the schedule protocols present some advantages such as avoiding collisions, overhearing and idle listening, achieving these features with schedule transmissions and listen periods; however, this leads to strict time synchronization requirements. On the other hand, the contention based approaches relax time synchronization requirements, since they are based on Carrier Sense Multiple Access (CSMA) technique, which leads to higher costs for message collisions, overhearing and idle listening. Although there are many implementations of MAC protocols for WSNs, there is no protocol considered as a standard [52]. One of the reasons for this lack of standardization, is the fact that the chosen MAC protocol will, in general, be application-dependent, which means that it is difficult to define *one* standard MAC protocol for WSNs. Another reason that can be pointed is due to the lack of standardization at lower layers, such as the physical layer.

As it was presented before, LoRaWAN implements a simple MAC protocol based on ALOHA, although LoRaWAN only supports one-hop from the node to the gateways, being these gateways devices that are powerful and able to run LoRaWAN. This is not available to the development of this dissertation's work, since only simple LoRa radios are available to implement both the end-devices and the sink/gateway.

Bor et. al propose LoRaBlink [54], a protocol that aims to support reliable and energy efficient multi-hop communication, as well as low-latency bi-directional communication. However, the authors make some assumptions: the network has low density, low traffic volume, and contains a limited number of nodes, which can not be guaranteed in the scenarios expected in the work developed in this dissertation. Another property of LoRaBlink is the synchronization between the nodes to defined slotted channel access, an implementation that we do not want to rely on.

2.3.4 Clustering

Organizing sensor nodes into clusters has been generally pursed by the research community in order to achieve the network scalability objective [55]. Every cluster would have a leader, usually defined as the Cluster-Head (CH). A vast number of clustering algorithms have been proposed for WSNs [55]. These proposed clustering techniques vary depending on the nodes deployment, the desired network architecture, the properties of the potential CH nodes and the network operation model.

The clustering process can stabilize the network topology at the level of sensors and thus put aside topology maintenance, reducing the overhead. In this case, sensor would only communicate with their CHs and would not sense changes in the level of inter-CH tier. Furthermore, a CH can aggregate the data collected from its member nodes and thus decrease the number of relayed packets.

Figure 2.7 shows the architecture of a generic WSN implementation.

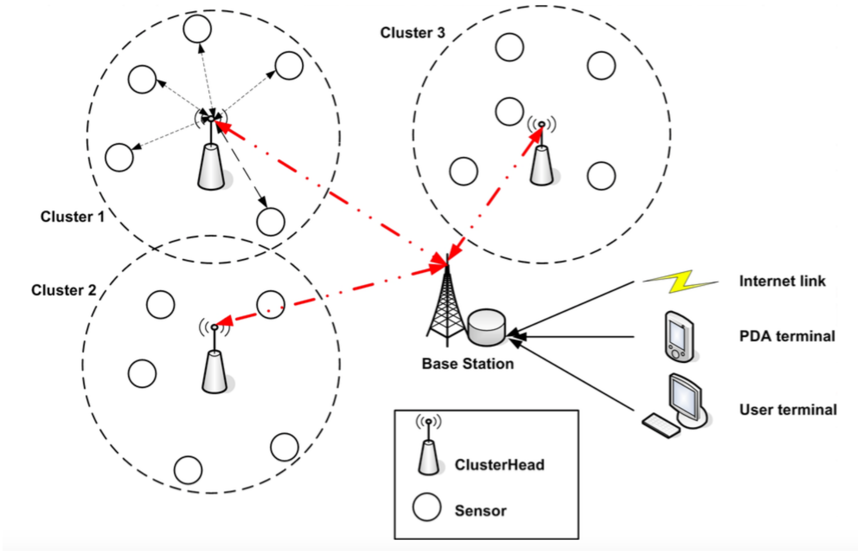


Figure 2.7: General Sensor Network Architecture, based on [56]

The WSN and the clustering process have a well defined set of elements that compose the architecture:

Sensor Node The sensor node is the base of a WSN: it can take on multiple roles, such as data collection, data storage, routing and data processing.

Clusters They are the organizational scheme used in WSNs. Due to the dense nature of the WSNs, these networks require an organization process in order to simplify processes, such as the communication.

Cluster-Head The CH is the cluster leader. It is responsible to aggregate the data that comes from the sensor nodes, and in case of a schedule based communication process, to provide this schedule.

Base Station The base station is at the top of the organizational hierarchy; it provides the link between the sensors and the end-users.

End-User The end-user is the entity that is going to use the data acquired by the network. This data is mostly accessed over an Internet connection. Also, if the network is query-based, it is the end-user that generates this query.

Low Energy Adaptive Clustering Hierarchy (LEACH) is one of the most popular clustering algorithms for WSNs [57], so that many variants have been proposed so far [58]. The clusters are formed taking into account the received signal strength, and use the CH nodes as bridges to the base station. The data processing, such as aggregation and fusion are done at the cluster level. In LEACH the clusters are formed using a distributed algorithm where the nodes are self-organized, without a centralized control. Initially a node decides to be a CH with a probability p and broadcasts its intention. Each node not contending for CH chooses a CH based on the energy consumption that is required to reach each CH. In order to balance the load in the cluster, the CH role is periodically rotated among the nodes.

The decision to change the CH probabilistically can lead to a selected node that has very low energy remaining, and when this node dies, the whole cell becomes dysfunctional.

Even though the LEACH protocol has not been applied in its all in the proposed solution, the hierarchical structure and the received signal strength usage in the CH selection were adapted to the characteristics of the network in use. More precisely, the use of the received signal strength is used in the CH selection process, but it takes into account the link quality between the nodes and the Sink. The CH rotation was not used since that is assumed that the nodes are static and do not have energy issues.

2.4 Chapter Considerations

This chapter provided an overview on the emerging LPWANs technologies, as well as its applications and future trends and the related work that is included in one or more parts of the proposed solution.

First it was presented how the LPWAN technologies emerged in the IoT market and what kind of advantages they bring compared to the existing legacy technologies. Then, some of the more relevant available solutions are summarized.

The LoRa technology was analyzed in more detail due to its usage as an integral part of the solution. The technology physical layer was depicted, as well as the configuration parameters that influence the communication properties. The features of the Lora-Alliance proposed protocol for LoRa were already described.

In terms of related work, the LoRa coverage studies already done were presented and compared with the study done in this dissertation. Data gathering and Smart Cities real implementations and testbeds are also presented. MAC and clustering algorithms are an important requirement of the solution, so a study on the existing proposals was made and debated on how these implementations can (or cannot) be adapted to a LoRa based network.

In the next chapter it is presented the proposed solution.

Chapter 3

Proposed Architecture

3.1 Introduction

This chapter presents the designed architecture, as well as the mechanisms and protocols proposed to achieve the final purpose of making data acquired by the sensors able to reach a server through different technologies and networks.

Section 3.2 presents the designed architecture overview with a brief explanation of each relevant working mode.

Section 3.3 gives an explanation on how the data is gathered and structured to be then forwarded through its path to the server.

Section 3.4 explains how the multi-technology communication is achieved, the details about each technology functioning mode and the management of all communication process and data gathering.

Section 3.5 provides information on how the media access is handled in order to control simultaneous transmissions attempts of the devices.

Section 3.6 refers to the clustering process that is developed as well as its details, such as, the cluster head selection, the node association/management and the packet forwarding used.

Section 3.7 presents the chapter considerations.

3.2 Architecture Overview

The proposed architecture showed in Figure 3.1 aims to provide several communication technologies beyond WiFi, to transport data acquired by a set of sensors to a server. In this case, the communication is performed also using the LoRa technology.

The sensor nodes are organized in clusters, which is explained in Section 3.6. With the purpose to achieve a more organized and scalable network, the aggregated information from the clusters is then forwarded to a sink through the communication channel.

Each node has the capability of sending its data through WiFi (to a fixed hotspot or through a mobile hotspot attached to a vehicle) or LoRa, since the two technologies are

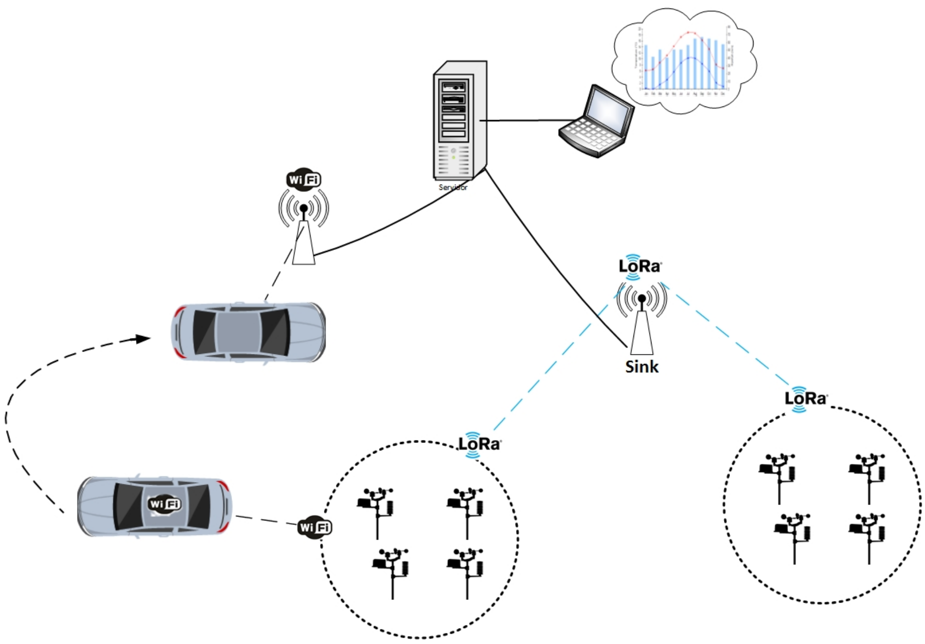


Figure 3.1: Architecture Overview

available. To manage to which technology the data should be forwarded, a Connection Manager is proposed and created.

The LoRa modules in the sensor nodes as well as in the sink, that has LoRa reception capabilities, are half-duplex radios, that is, they can only receive or transmit at a time. In order to ensure that the communication between nodes through LoRa is reliable, that is, a minimum number of collisions and packet retransmissions occurs, a MAC protocol has to be developed taking into account the possible constraints that a LoRa network is exposed to.

The sensors along with the different communication technologies form a set called Data Collecting Unit (DCU) whose architecture is present in Figure 3.2. The DCU architecture can be divided in two sub-modules: one that is formed by an existing data gathering device, which has a set of sensors; the other one is composed of a Raspberry Pi 2, a WiFi dongle and a SX1272 LoRa module along with a Multiprotocol Radio Shield commercialized by Libelium. The two Raspberry Pis communicate through a physical Ethernet connection. The reason why a second Raspberry Pi is needed relates to the impossibility of connecting the LoRa module to the existing sensor set, since it is already occupied with the sensors control board.

The Sink architecture is also present in Figure 3.3; it is composed of a Raspberry Pi 2 and a SX1272 LoRa module along with a Multiprotocol Radio Shield, since this Sink is the end point of the LoRa communication.

The proposed software architecture that is designed to implement all the proposed capabilities of the DCU is presented in Figure 3.4. It includes: the data gathering software, that is responsible to acquire the sensors data and store it locally; the connection manager,

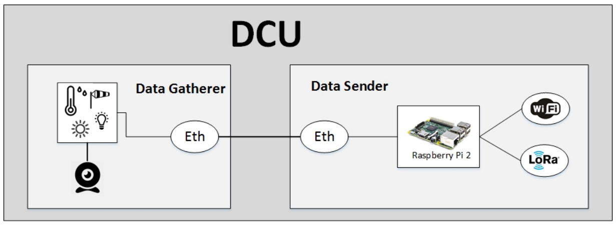


Figure 3.2: DCU Architecture

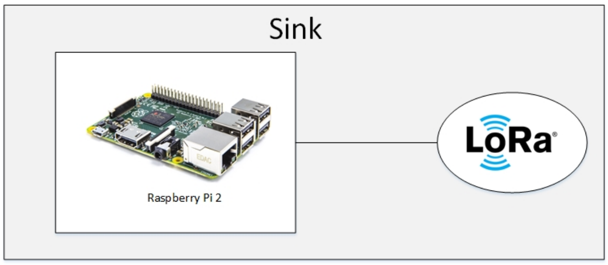


Figure 3.3: Sink Architecture

---

that decides which is the most suitable technology to forward the data; and the technology managers, that are responsible to handle the communication of each technology interface.

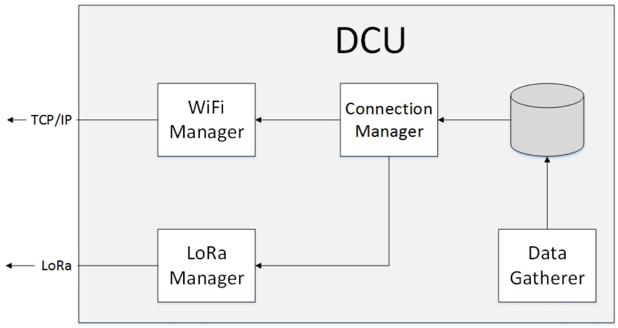


Figure 3.4: DCU proposed software architecture

3.3 Data Gathering

Data gathering is performed by a static set of sensors that includes the measurement of:

- Temperature;
- Humidity;
- Luminosity;
- Wind Speed;
- Wind Direction;
- Precipitation;

- Carbon Monoxide (CO);
- Nitrogen Dioxide (NO_2);
- Ozone (O_3);
- Particles.

The data that is acquired by the sensors in use is grouped into two types of data: environmental sensors, that include temperature, humidity and luminosity; and weather sensors, that include precipitation, wind speed and direction.

Sensors data samples are then packed within a defined structure that includes the data type group, timestamp of when the data was acquired, acquisition duration time (that is useful for wind speed and precipitation) and the data of each sensor. Figure 3.5 shows the sensor samples structure.

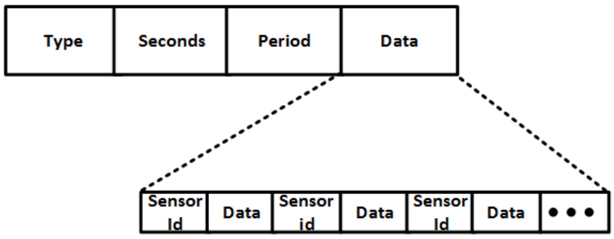


Figure 3.5: Sensor Data Structure

Besides the environmental and weather sensors, some nodes can also have a camera attached, which enables the nodes to acquire data of the surrounding environment in the image form. It is possible that the data categories mentioned can have different sampling periods.

The data gatherer hardware architecture is shown in Figure 3.6. It is composed of a sensor board that comprises the diverse sensors available, a control board that implements an interface of communication between the data gathering software and the sensors, and a Raspberry Pi 2 on which the control board is assembled.

The software architecture is shown in Figure 3.7. It comprises the data collector software running in the Raspberry Pi of the data gatherer device and the data management process in the Raspberry Pi responsible for the communication process.

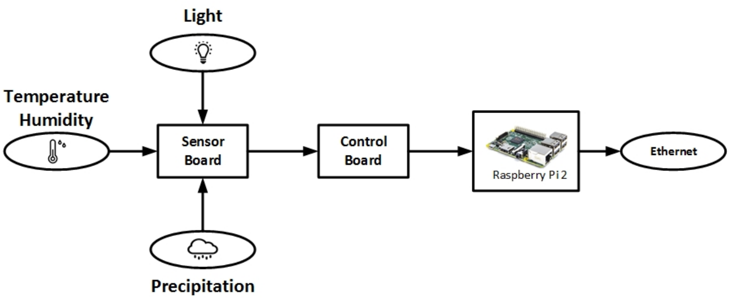


Figure 3.6: Data Gatherer Hardware Architecture

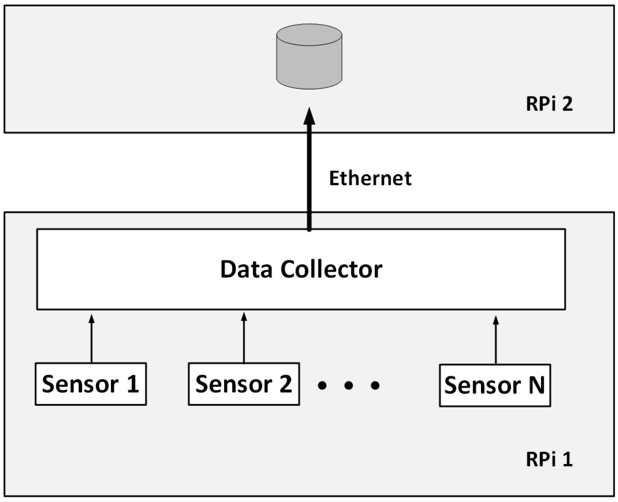


Figure 3.7: Data Gatherer Software Architecture

3.4 Multi-Technology Communication Approach

The multi-technology approach aims to provide a more versatile and adaptive data gathering protocol. For this purpose two communication technologies with distinct properties are chosen. It is intended to have a technology that allows a sensor network deployment scenario in an environment where this technology can already be found and easily be integrated with the protocol. In that way, the WiFi technology was the one that fulfilled this requirements. Moreover, it is also desired that an alternative exists for when the WiFi technology is not available. The alternative technology that was found with potential capabilities to suppress the WiFi limitations was LoRa, since it provides a long range communication and is able to reach a wider space.

3.4.1 Technologies

WiFi is one of the chosen technologies to be used along with the sensor nodes. This is because of its common use and, in case of a deployment of a sensor network in a city, it is quite easy to find a public Access Point (AP) where the DCU can connect to dispatch its data. Furthermore, if the deployment is made in a city that has a vehicular network as the one in Porto city, there is another communication possibility, that allows the DCUs that are not in a range of a fixed access point, to send their data through the vehicular network, since the vehicles act as moving APs. As vehicles can act as data mules, drones can also be used for this purpose, which gives several possibilities for data collection among the nodes.

The chosen alternative technology is LoRa since it is in constant growing in the IoT networks and M2M market, also because of the various capabilities and advantages that it carries to the sensor networks. With the long-range communication capacity as its main strength, low-power feature and reduced cost, LoRa provides a new set of options for the WSNs and IoT networks implementation.

With various operation modes that give different trade-offs between range and throughput, LoRa ensures an adaptation to several types of environments, both rural and urban. Since the data aimed to be forward by LoRa are sensor measurements, it is expected that no more than a few bytes are required to dispatch this data. Thus, LoRa fits in all requirements and provides a solid alternative technology.

3.4.2 Connection Manager

In order to achieve the multi-technology communication, an entity, designated *Connection Manager*, is a module responsible for managing how the communication shall be done and which technologies shall be used for the different services. The *Connection Manager* has several roles that it has to perform, that are:

Network Scanning The *Connection Manager* is responsible for scanning if any available WiFi networks are in range. The scanning period can be defined to a value considered appropriated.

Network Managment Once a proper network is found in range, the *Connection Manager* connects to that network, constantly monitoring its state, and eventually disconnecting from the network if it does not show enough quality.

Tecnhology and Traffic Managment Depending on the technologies available, the *Connection Manager* has to take an action according to the nodes characteristics and generated traffic. From this, different situations and actions can occur:

- **Single Technology** Since the LoRa technology interface is mostly available (due to its long range), the most common situation that could occur is that it is the only available technology to perform the communication. In this situation the *Connection Manager* forwards only sensor data through the LoRa interface. This happens because it is the traffic that does not require a high throughput communication channel, due to its small packets size. If any of the data stored requires a more demanding communication channel, it will have to wait until one is available.

- **Technology Preference** If more than one technology is available and only sensor data traffic is being generated, the *Connection Manager* acts in a way that gives preference to the technology that presents better communication conditions. So, if a WiFi network shows up and a node has traffic to send, the *Connection Manager* selects this WiFi network to forward the traffic. If the node is sending data through the LoRa interface and a WiFi network appears in range, the *Connection Manager* only changes the forwarding interface after all the association and authentication processes are done. This leads to a make before break process, in this way no time is wasted and the interface changing is unnoticeable in terms of breaking the connectivity.

Traffic Differentiation Different types of traffic can be generated by a node. If it has a camera attached, from time to time it will generate image traffic data that has a larger size than the sensor data. When the two different technologies are available the *Connection Manager* is capable of differentiating the traffic by the interfaces, assigning the traffic that requires a higher throughput, to the WiFi interface. The LoRa interface is responsible to forward the sensor data traffic. This way it is possible that the two technologies coexist at the same time without interfering with each other, being used simultaneously for different types of data.

The three communication situations that can occur in a node are exemplified in Figure 3.8:

- Only LoRa technology available (Single technology communication);
- Technology preference;
- Technologies simultaneous communication.

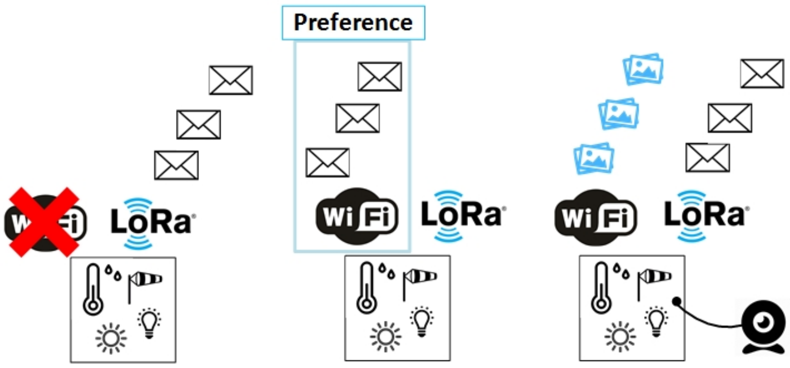


Figure 3.8: Connection Manager Communication Possibilities

In order to achieve the features proposed for the *Connection Manager*, the module is designed taking into account three processing lines, that have different responsibilities, as shown in Figure 3.9:

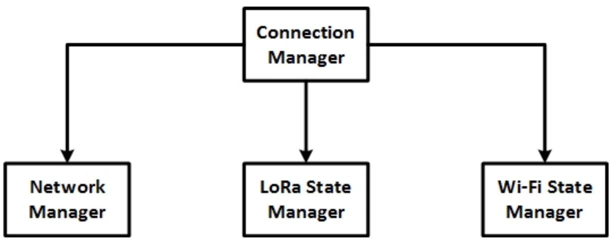


Figure 3.9: Connection Manager Contexts

Network Manager It is responsible for the network scanning and management processes;

LoRa State Manager Handles the communications through the LoRa radio interface, taking into account which type of data is in the waiting state to be sent;

WiFi State Manager When a network is available, this module handles the data forwarding through the WiFi interface.

3.5 Media Access Control (MAC) in LoRa

Since LoRa communication is achieved through a single radio transceiver, that is only able to transmit or receive at a time, and the network is supposed to be composed of many nodes in the same range of each other, MAC is an essential part of the communication stack for the overall architecture. For the MAC protocol design there are some essential features aimed to be provided [59]:

Framing Define the proper frame format and data encapsulation and decapsulation for communication between nodes.

Media Access Manage which nodes shall communicate when they need to. Its function is to minimize the number of collisions and corrupted packets through this management.

Reliability Provide reliable data delivery between nodes. For this purpose Acknoledgment (ACK) messages and retransmissions may be used.

In this MAC protocol design, power consumption awareness and management is not taken into account.

In order to provide a protocol that is able to fulfill all the requirements, some network characteristics have to be taken into account:

Node Disposition The nodes can be deployed at a predetermined location or can have a random disposition. The distance between nodes can vary as well and can go from some dozens of meters to some kilometers.

Packet Time-On-Air The different LoRa configurations define different communication times that are needed to send data. The communication time can be some millisecond or a few seconds according to the desired configuration.

Asynchronous Communications Although the data gathering is periodic in the nodes, there is no guarantee that the communications between nodes would be as well. With this in mind, it was assumed that the communications can occur at any time.

With the requirements defined and the network characteristics verified, some considerations about the MAC protocol design can be stated. An example is the fact that a scheduled/synchronized protocol approach would be difficult to implement due to the possible long distance between the nodes. This can lead to a relevant delay, caused by the configuration that is needed to achieve this long distance. Once the schedules need to be exchanged in the scheduled/synchronous algorithms, this makes it an unsuitable protocol to be used. This leads to a requirement that is the fact that the protocol shall be able to handle communications at any time.

With these constraints allied to the requirements of minimizing collisions and packet retransmissions, an approach based on an extended version of CSMA [60] called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), (this uses Request to Send (RTS)/Clear to Send (CTS) message exchange to control the media access of the devices) is the starting point of the protocol's design. Along with the RTS/CTS basis, other features provided by improvements of CSMA, such as Multiple Access with Collision Avoidance (MACA) [61], Multiple Access with Collision Avoidance for Wireless (MACAW) [62] and from IEEE 802.11 MAC Layer [63] are taken into account to build a simple, yet solid and reliable, protocol that can work along with LoRa.

The characteristics inherent to the developed protocol are:

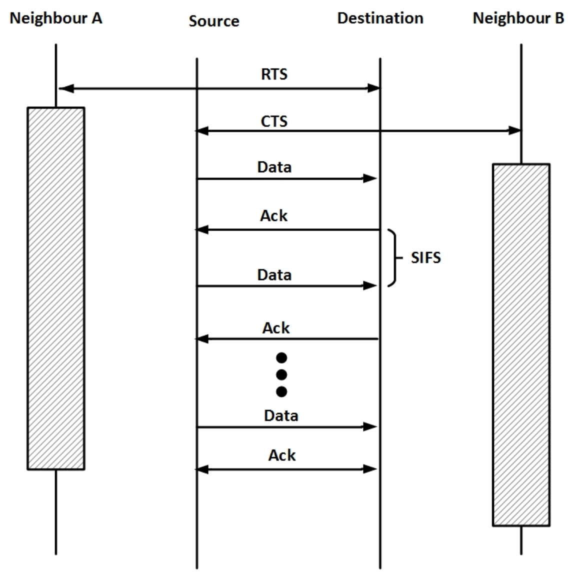
- Carrier sensing is not applied. Only messages timeout and responses;
- RTS/CTS message exchange before data transmissions;
- Data is sent in bursts of packets, meaning that using one RTS/CTS message exchange, more than one data packet can be sent without a new channel reservation;
- ACK messages are sent for every data packet received;
- Packet retransmissions are used;
- Short Interframe Space (SIFS) are applied;
- Expected transmission time is calculated and sent in the RTS and CTS packets;
- The sending node has the calculated expected transmission time to try to deliver the packets. After this time the node is obliged to leave the media free;
- The nodes use always-listening state;
- Nodes that overhear RTS or CTS packets enter in a backoff state for the amount of time contained in the message;
- Wait To Send (WTS) packet was created.

When a node receives an RTS and is able to receive data from the requesting node, it replies with a CTS packet. This exchange is made in broadcast, so that other nodes can overhear it and do not attempt to access the channel for certain time. Once the requesting node receives the CTS, it waits a SIFS before starts transmitting its data. Then it waits for the respective ACK. Between each data packet delivery attempt it is also applied a

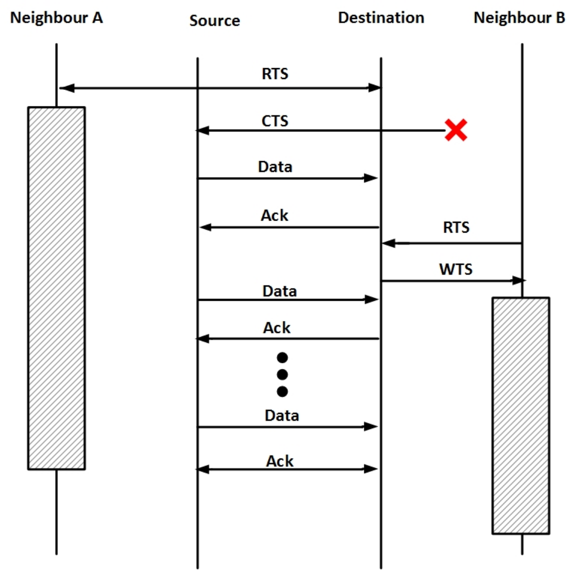
SIFS. This data/ACK sequences are made in unicast. After the sending node delivers all the packets or the expected communication time ends, the media becomes free to be reserved again.

If a node sends an RTS and does not get any CTS response, it will try to request the media two more times before entering the backoff state for a random amount of time. The same occurs for the number of retransmissions for data packets before another packet is tried to be sent. If a node receives an RTS while receiving data from other node, it informs the node that tried to reserve the communication channel that it is currently busy with a WTS packet that contains the remaining time for the communication to end. The decision to create the WTS packet is due to the LoRa non-destructive communication property, which enables that a packet with a stronger signal is not interfered by a packet with a weaker signal. This can lead to an RTS reception in the middle of a Data transmission, to overcome this issue the WTS packet is created.

In Figure 3.10 is shown the different behaviors that the protocol adopt according to aforementioned premises.



(a) Normal communication channel request



(b) Second node trying to request the communication channel

Figure 3.10: Protocol's behavior

3.6 Clustering

Instead of a flat single-hop organization (Figure 3.11), a cluster based approach is used, in order to avoid a large number of simultaneous transmissions from the nodes to the sink. Since the LoRa communications can take a while, it would be expected that a significant number of collisions may occur when a large amount of nodes try to send their data to the sink.

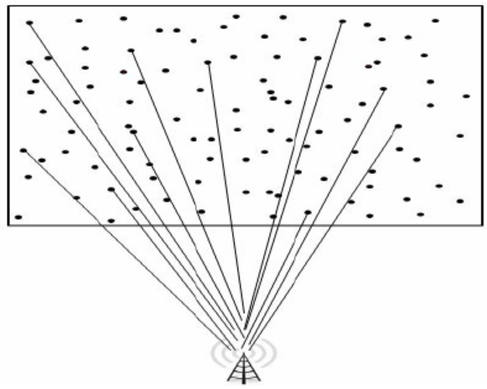


Figure 3.11: Single-Hop Routing

To ensure that the data delivery is reliable, the network is scalable and organized, a LEACH based cluster organization is adopted, where the clusters are formed by normal sensor nodes and a CH. This CH is responsible for the aggregation of the data received from all the nodes that belong to the cluster as well as its own gathered data. This aggregated data is then forwarded by the CH to the Sink. Figure 3.12 shows the proposed organization model.

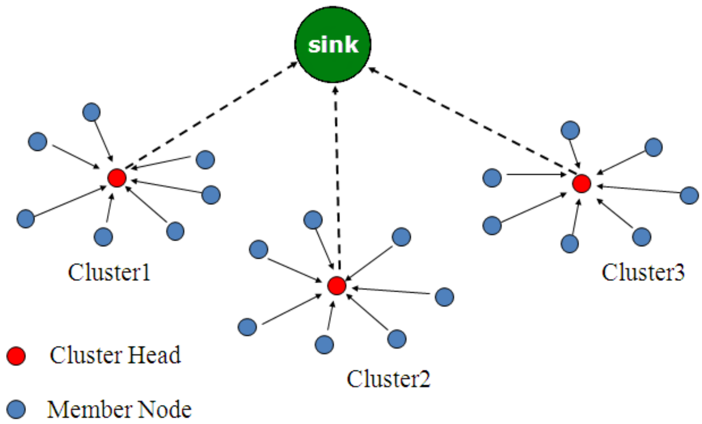


Figure 3.12: Clustering organization

3.6.1 Clusterhead Selection

The CH selection is the first step in the clustering process proposed. It is triggered by the Sink when it sends a specific packet, that signals to all listening nodes that the cluster formation will start.

The clusterhead selection takes into account the link quality between the nodes and the Sink. In order to get this quality, a fixed number of packets, called beacons, are sent by the sink to the nodes. When a node receives a beacon, it gets the corresponding SNR and RSSI[1] in order to estimate its weight to the sink when all the beacons have been sent. The link weight estimation is presented in Equation 3.1.

$$f_{weight} = \frac{RSSI_{mean}}{RSSI_{min}} \times \frac{1}{3} + \frac{SNR_{mean}}{SNR_{min}} \times \frac{1}{3} + (1 - \frac{ReceivedBeacons}{TotalBeacons}) \times \frac{1}{3} \tag{3.1}$$

The minimum value of the function is 0 and the maximum value of the function is 1, meaning that the receiver did not hear any beacon. This implies that a lower value of the `link weight` means that the link between the sink and the node has a high quality. After all nodes get their `link weight` to the Sink, they wait a time proportional to their `link weight`, that is, the lower the link weight the lower the time that the node will wait. The waiting time approach is used to minimize the convergence time. When the node is in the waiting process, two situations can occur:

- Waiting time expires and the node broadcasts its link weight to other nodes in its range, signaling that it is a CH contender. One of the two situations occur:
  - This CH contender has the lowest link weight in its range and only receives join requests from other nodes;
  - A node in this CH contender range has lower link weight than the CH contender, so it drops the CH state and joins this new CH contender.
- Node hears a CH contender. One of the two situations occur:
  - If the CH contender link weight is lower that its own, the node will join the CH contender;
  - If the CH contender link weight is higher that its own, then this node sends a CH contender message and assumes the CH state.

3.6.2 Packet Forwarding

The packet forwarding is made from the bottom to the top of the hierarchy organization of the nodes and is unidirectional. This means that, as the sink is the top node of the hierarchy, it will be the end point of the LoRa forwarding process, the cluster member node is in the base of the routing process and the CH is the middle element.

When a node reaches the threshold to deliver the packets to the next point in the hierarchy, one of the following situations can occur:

- Node is a normal sensor node and sends its packets to its CH;
- Node is a CH and sends its packets to the sink.

[1]RSSI is a negative value

3.7 Chapter Considerations

In this chapter the main accomplishments of this dissertation were mentioned and the main concepts were explained. The proposed architecture to achieve the final objective was presented. This architecture includes the data gathering process, the multi-technology possibility through WiFi and LoRa, as well as the software modules developed to handle the data dispatch from the DCU to the server.

First, an entity that shall be capable to decide which technology presents the best conditions to be used for performing the communication is designed and denominated Connection Manager. To provide a reliable communication between the LoRa nodes presented in the network, a MAC protocol is designed taking into account the needs of the technology. At last, in order to make the network scalable and of easy adaptation to different scenarios, a cluster based organization of the nodes is proposed.

The next chapter presents in more detail how the proposed protocols/modules are implemented and how they interact.

Chapter 4

Implementation

4.1 Introduction

This chapter describes with more detail the implementation of the protocols that were presented in Chapter 3.

Section 4.3 presents the implemented packet structure and the fields that composes this structure.

Section 4.4 presents the implementation of the cluster formation process. It is explained how the nodes behave according to their role and neighborhood. The specific cluster formation packets are also presented.

Section 4.5 explains the implemented MAC protocol, as well as its specific messages.

Section 4.6 explains how the data is managed depending on its type, and how it is kept until it is delivered to the next hierarchy member.

Section 4.7 presents the Connection Manager module and explains how it works, the implementation of its elements and how they interact.

Section 4.8 presents the chapter considerations.

4.2 Base

To implement the proposed solution, the LoRa modules used are SX1272 manufactured by Libelium. Libelium's LoRa module works in both 868 and 900 MHz ISM bands. Those frequency bands are lower than the popular 2.4 GHz band, so path loss attenuation is better in LoRa. In addition, 868 and 900 MHz are bands with much fewer interference than the highly populated 2.4 GHz band. Besides, these low frequencies provide great penetration in possible materials (brick walls, trees, concrete), so these bands get less loss in the presence of obstacles than higher bands.

Figure 4.1 and Table 4.1 present the module used and its characteristics.

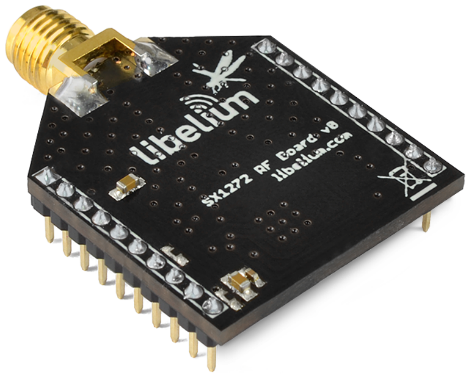


Figure 4.1: SX1272 Module

LoRa	
Module	SX1272
Dual Frequency Band	863-870 MHz (Europe) 902-928 MHz (US)
Transmission Power	25 mW
Sensitivity	-134 dBm
Channels	8 (868MHz) 13 (900MHz)
Range	LOS = 21km NLOS = +2km

Table 4.1: SX1272 Module Characteristics

The SX1272 LoRa module is connected along with the Multiprotocol Radio Shield to the Raspberry Pi 2 (Figure 4.2).

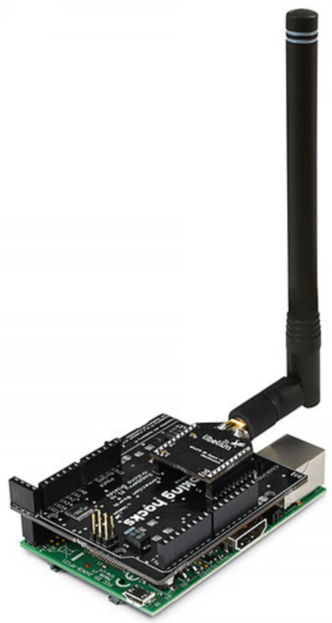


Figure 4.2: Raspberry Pi 2 with Multiprotocol Radio Shield and SX1272 Module

The SX1272 LoRa module counts with a C++ library that provides the management of the SX1272 LoRa module in a simple way. This Application Programming Interface (API) offers a simple-to-use open source system. Some of the main functions are listed in Table 4.2.

Table 4.2: General SX1272 functions

Function	Comments
ON()	Opens the SPI and switches the SX1272 LoRa module ON.
OFF()	Closes the SPI and switches the SX1272 LoRa module OFF.
setLORA()	Sets the module in LoRa transmission mode.
setMode()	Sets the BW, CR and SF of the LoRa modulation.
setHeaderON()	Sets the module in explicit header mode (header is sent).
setHeaderOFF()	Sets the module in implicit header mode (header is not sent).
setCRC_ON()	Sets the module with CRC on.
setCRC_OFF()	Sets the module with CRC off.
setChannel()	Sets the indicated frequency channel in the module.
setPower()	Sets the signal power indicated in the module.
setNodeAddress()	Sets the node address in the module.
getSNR()	Gets the SNR value in LoRa mode.
getRSSI()	Gets the current value of RSSI from the channel.
getRSSIpacket()	Gets the RSSI of the last packet received in LoRa mode.
sendPacketTimeout()	Sends a packet to the specified destination before a timeout expires.
sendPacketTimeoutACK()	Sends a packet to a destination before a timeout and wait for an ACK response.
sendPacketTimeoutACKRetries()	Sends a packet to a destination before a timeout and wait for an ACK response and retry to send the packet if ACK is lost.
receivePacketTimeout()	Receives information before a timeout expires.
receivePacketTimeoutACK()	Receives information before a timeout expires and responds with ACK.
receiveAll()	Receives all the information on air with maximum timeout.

There are ten predefined modes in the API, including the largest distance mode, the fastest mode, and eight other intermediate modes that Libelium has found interesting. All of them can be modified or deleted, and also it is possible to attach new modes in the appropriate function. The predefined modes and its properties are shown in the next table. Table 4.3 presents the different modes and its configuration parameters.

Table 4.3: LoRa Modes

Mode	BW (Hz)	CR	SF	Sensitivity (dB)	Comments
1	125	4/5	12	-134	max range, slow data rate
2	250	4/5	12	-131	
3	125	4/5	10	-129	
4	500	4/5	12	-128	
5	250	4/5	10	-126	
6	500	4/5	11	-125.5	
7	250	4/5	9	-123	
8	500	4/5	9	-120	
9	500	4/5	8	-117	
10	500	4/5	7	-114	min range, fast data rate, minimum battery impact

The proposed solution implementation is developed using Python 3.5 programming language, and because of that, the provided SX1272 API is not directly usable. In order to use the provided functions, it is created an extended Python module with C++. Such *extension modules* can do two things that can not be done directly in Python: they can implement new built-in object types, and they can call C/C++ library functions and system calls [64]. The functions provided by this API are the base point for all the modules implemented.

The API has a defined packet structure that is shown in Figure 4.3. This structure has many fields to be filled by the user or the application:

- **dst** - Destination node address: this parameter is indicated as an input in the function used by the user.
- **src** - Source node address: this parameter is filled by the application with the module's address (previously set by the user).
- **packnum** - Packet number: this parameter indicates the packet number and is filled by the application. It is a byte field, so it starts in 0 and reaches 255 before restarting. If the packet is trying to be retransmitted, the packet number is not incremented.
- **length** - Packet length: this parameter indicates the total packet length and is filled by the application.
- **data[MAX_PAYLOAD]** - Data to send in the packet: It is used to store the data to send to other nodes. All the data to send must be stored in this field. Its maximum size is defined by MAX_PAYLOAD. The maximum payload size is 250 bytes.
- **retry** - Retry counter: this parameter is filled by the application. It is usually equal to 0. Only when the retries feature is used, this value is incremented from 0 to the maximum number of retries stored in the global variable _maxRetries which value is 3 by default. If the packet is sent successfully, or if the maximum number of retries is reached without success, the retry counter is set to 0.

dst	src	packnum	length	data	retry
1 byte	1 byte	1 byte	1 byte	variable bytes	1 byte

Figure 4.3: Lora Base Packet

4.3 Packet Structure

In order to implement the proposed solution, a new packet structure is proposed. This new packet is implemented on top of the *data* field of the LoRa SX1272 API.

The proposed packet structure is presented in Figure 4.4. This packet has a maximum length of 250 bytes from which ten bytes are for the header and the remaining 240 bytes are used for the payload of the proposed packet.

Destination Address			Source Address			Packet Type		Payload
4 Bytes			4 Bytes			2 Bytes		240 Bytes (Max)
LoRa	**Cluster**	**Node**	**LoRa**	**Cluster**	**Node**	**Class Type**	**Sub-Class Type**	**Payload**
1 Byte	1 Byte	2 Bytes	1 Byte	1 Byte	2 Bytes	1 Byte	1 Byte	Variable

Figure 4.4: Solution Base Packet

The header is composed by the destination node address, the source node address and the packet type. Each one of the packet elements is detailed next:

Source/Destination Address The address fields have a size of four bytes and are formed by the association of the correspondent LoRa address of the node, the cluster number and the node number;

- **LoRa Address** The `LoRa Address` is composed of a 1-byte integer and represents the SX1272 LoRa module address. This address is used by the base SX1272 module to filter the incoming packets according to their destination address. The LoRa address can take a value from 0 to 255, wherein the 0 value is used as LoRa broadcast address;
- **Cluster Number** The `Cluster Number` field contains a 1-byte integer. This lead to a range of 1 to 127 for the `Cluster Number`, which mean that a maximum of 127 clusters can coexist in one sink's range;
- **Node Number** This field is a 2-byte integer and it is the node's unique identifier. Once the field has a two bytes length, it allows that a node can be identified at the data server side by its `Node Number` in a multiple sink scenario, even if it has the same LoRa address of other nodes;

Packet Type The `Packet Type` field is composed by two bytes. The most significant one represents the packet class of the message, which is associated with the different developed protocols, such as: the cluster formation; MAC; gathered data. The least significant byte represents each class specific messages denominated here as the subtype of the message;

Payload The message `Payload` is where the information of the different type of packets is carried. The `Payload` field has a variable length that can go from an empty payload to a 240 bytes payload.

The packets have to be sent to the network from the sensor node to the remote server. So, right from the start, the packets are packed with network byte order, referred as to the big-endian architecture.

4.4 Clustering Process

The cluster formation process is the first procedure that has to be executed in order to organize the nodes in groups depending on their location and distance between nodes.

These groups are called clusters.

4.4.1 Cluster Formation Messages

There are specific packets that have been created for the cluster formation process. These packets are included in the `Cluster Formation` packet class. This class is defined in the `Packet Type` field by the most significant byte with the value `0x01`. Belonging to this packet class there are seven packets that are differentiated by the value of the least significant byte of the `Packet Type` field. These packets are described next.

Cluster Formation Start This is the first packet that is received by the nodes in the cluster formation process. It is sent by the sink to ensure that all nodes in range are synchronized at the time of the first beacon. This packet has no payload and it is defined with the value `0x01`;

Beacon The `Beacon` packet is sent by the sink to all nodes in range so they can calculate a link weight between them and the sink based on the quality of the beacons received. This packet has no payload and it is represented with the value `0x02`;

Clusterhead Contender The `Clusterhead Contender` is the packet that is necessary to be sent by a node that wants to contend to be the CH, to all the nodes in its range. The payload of this packet contains a 4-byte floating point value, that represents the link weight of the contender to the sink. The `Clusterhead Contender` message is represented with the value `0x03`;

Join Clusterhead When a node wants to join a CH to form a cluster, it has to send a `Join Clusterhead` message to the CH contender. Posteriorly, the CH contender can add this node to its neighbor table and assume that it is a cluster member. The `Join Clusterhead` message does not possess any information in its payload and is represented with the value `0x04`;

Join Clusterhead Acknowledge A `Join Clusterhead Acknowledge` message is sent back by the CH contender, to the node that requested the association. This confirms that the node is effectively associated to that CH contender. As the `Join Clusterhead` message, the `Join Clusterhead Acknowledge` does not carry any information in its payload and has the value `0x05` associated to its type;

Cluster Formed Once established the CH contenders, they send a `Cluster Formed` message to the Sink to inform it that they are a CH contender. This packet's payload is composed of a 1-byte integer that carries the number of nodes associated to this CH contender. To identify this packet the value `0x06` is used;

Cluster Number For each `Cluster Formed` message received by the sink, a cluster number is generated and given to each CH. After the CH contender receives the given cluster number, it broadcasts it to the nodes associated to the CH contender. The

Cluster Number packet payload is a 1-byte integer that represents the cluster number. This message is represented with the 0x07 value.

4.4.2 Cluster Formation Process

Before the nodes can begin the cluster formation process, the Sink node needs to send a Cluster Formation Start packet to all the nodes in range, so they will enter in the receiving state for the incoming Beacon packets. After that, the Sink sends a fixed number of Beacons (we use ten) with a fixed time spacing of five seconds since the end of a delivery to the beginning of the other. Once all Beacons have been dispatched, the Sink waits for incoming Cluster Formed messages. For every Cluster Formed message received, the Sink returns a Cluster Number message.

If the total number of nodes received in the Cluster Formed messages equals the total deployed number of nodes, or, if the Sink has not received any Cluster Formed message in a specified time interval since the last one received, it declares the cluster formation completed.

After the node starts to receive the Cluster Formation Start message, it will listen for Beacons in ten listening time slots. From every received Beacon, the node acquires its RSSI and SNR values. If in a time slot a Beacon is not received, it is considered that the minimum RSSI and SNR values are assigned. After consumed all of the time slots, the nodes calculate their link weight to the Sink using the Equation 3.1.

Once the link weight is obtained, it is used to define the amount of time that the nodes will listen for CH contender messages before they assume themselves as a CH contender. For that, a value of 15 seconds is used as the maximum waiting time. This waiting time is obtained by multiplying the link weight by the maximum waiting time, which gives a proportional reasoning between the waiting time and the link weight. With this approach, it is expected that a minimal number of CH contenders arise, being these defined as fast as possible.

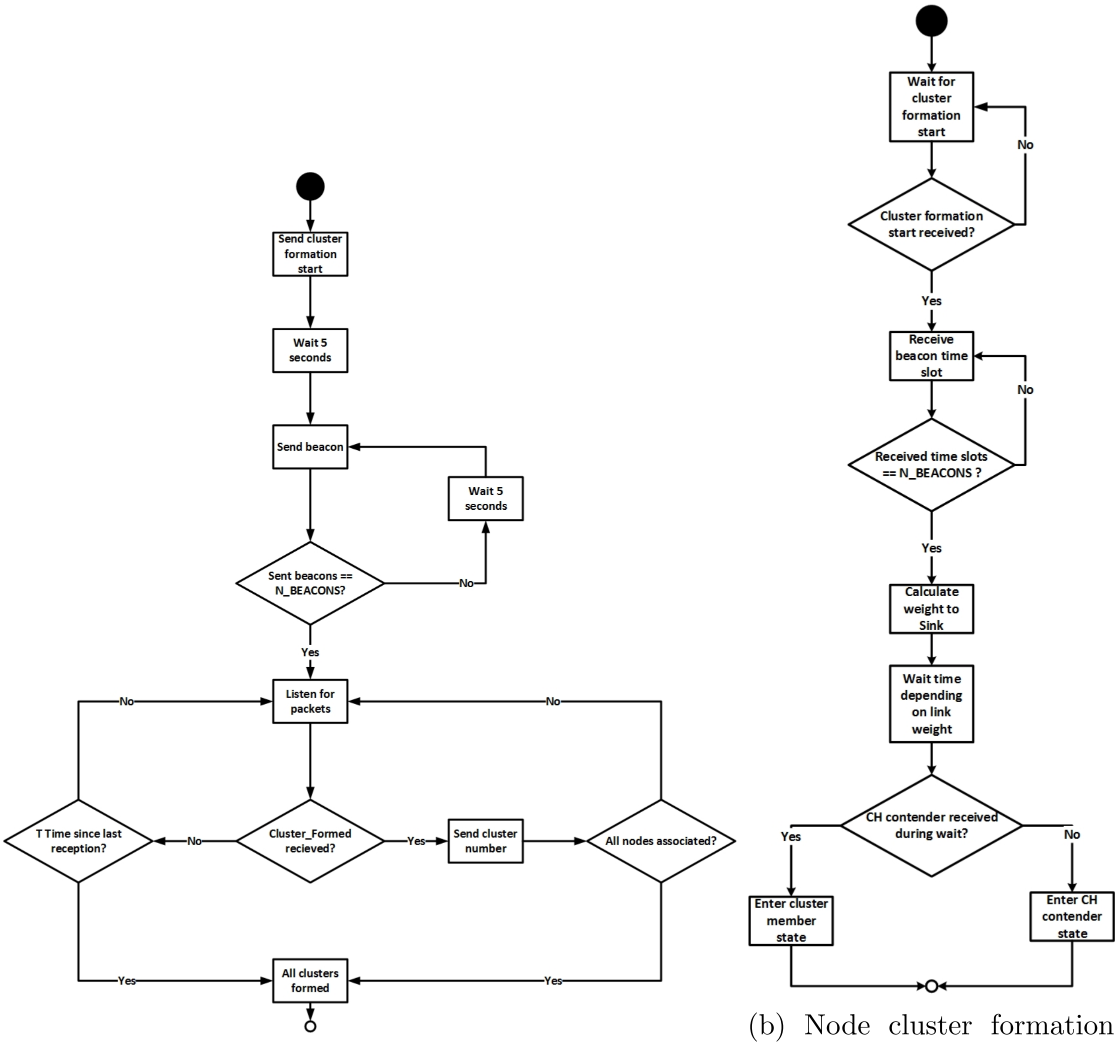
After the link weight has been calculated and the waiting time has been defined, the node can adopt two different states triggered by two distinct situations: CH contender state when the waiting time expires; cluster member state when a CH contender message is received during the waiting time.

In Figure 4.5a it is shown the Sink's behavior during the cluster formation process, and in Figure 4.5b it is shown the node's actions during the process.

As it was stated, during the cluster formation process, the nodes can be at one of two states: CH Contender or Cluster Member.

Figure 4.6a shows how the node behaves in the cluster member state, and in Figure 4.6b it is presented the CH contender state.

CH Contender When the waiting time expires, the node enters in the CH contender state. The first action of the node is then to broadcast a Clusterhead Contender message so that the nodes in range know that a node is trying to be a CH. After the Clusterhead Contender message has been sent, the CH contender awaits for



(a) Sink cluster formation flowchart

(b) Node cluster formation flowchart

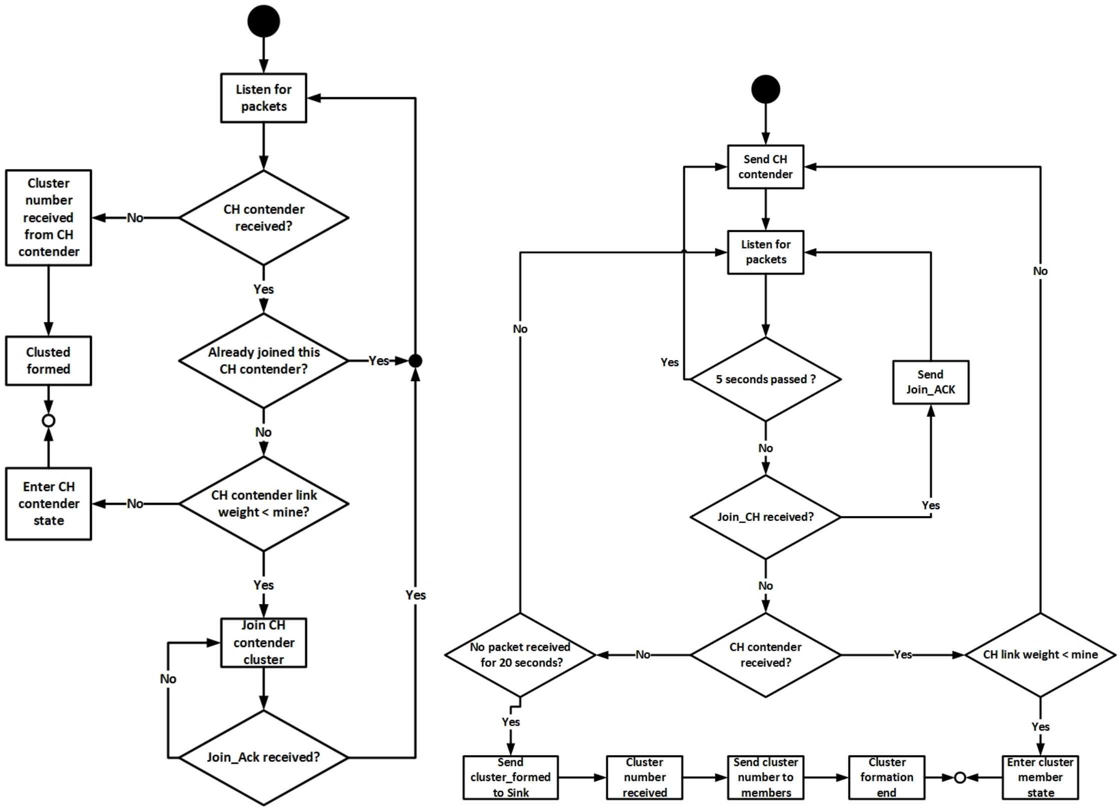
Figure 4.5: Cluster formation process

---

incoming packets at the same time as it repeats the `Clusterhead Contender` message in a period of five seconds. When a `Join Clusterhead` message is received, it is replied with a `Join Clusterhead Acknowledge` and the node becomes associated to the respective CH contender.

If 20 seconds have passed since the last `Join Clusterhead` message was received, it is considered that the cluster is formed. Then, the CH contender sends a `Cluster Formed` message to the Sink and waits for a `Cluster Number` message to be sent back. After the `Cluster Number` is received, it is then broadcasted to all the cluster members and the cluster formation process is considered complete.

In a case where a `Clusterhead Contender` message is received, one of two situations can occur: the contender's link weight is higher than the actual CH, and a



(a) Cluster member state flowchart

(b) Clusterhead contender state flowchart

Figure 4.6: Node cluster formation process

`Clusterhead Contender` message is broadcasted by the actual CH followed by the actions referred before; the new CH contender presents a lower link weight to the sink than the actual CH, dropping the CH contender state and assuming the cluster member state.

Cluster member The node enters the cluster member state either from receiving a `Clusterhead contender` during the waiting time or either coming from the CH contender state. When a node receives a `Clusterhead contender`, it compares the link weight of the incoming CH contender with its own, from there three outputs can occur: the node is already associated to the CH contender and no action is taken; the CH contender has lower link weight than the node's, and after a random waiting time, the node starts the process of joining that CH contender; the CH contender presents a link weight higher than the node's and then the node changes to the CH contender state.

When a `Cluster Number` message is received by a node, it assumes that the cluster where it is associated is effectively formed.

4.4.3 Neighbor List

Each node maintains a neighbor list where it keeps relevant information about the nodes that have any kind of relation with this node, for example, a CH maintains in its list information about all the nodes associated to the CH, plus the Sink information. In Figure 4.7 it is presented how the neighbor list is organized.

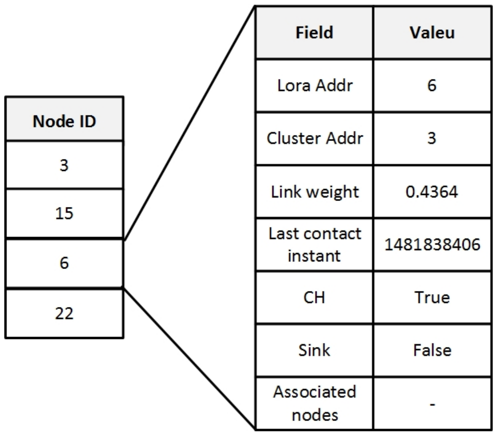


Figure 4.7: Neighbor list

This neighbors list is organized by the node number. Associated to this node number are the specific relevant information, which are:

LoRa Address Stores the LoRa address of the node;

Cluster Address Stores the cluster number where the node is associated;

Link Weight Stores the Link Weight to the sink of the node;

Last Contact Time This field keeps the timestamp of the last time when a contact occurred between the nodes;

CH This field indicates if the node is a CH or not;

Sink This field is used to indicate if the node is the Sink;

Total Nodes in Cluster This field is used in the Sink neighbor list to indicate how many nodes are in the cluster led by the corresponding CH. This implies that the `CH` field is set to True. In the other nodes this field has no value.

4.4.4 LoRa Communication Modes

The LoRa technology provides diverse combinations of the SF value and the BW of the signal, which leads to different characteristics of the resulting mode of communication. Two distinct modes of communication are adopted. One is used to perform intra-cluster communication and for that matter, a configuration that provides a shorter range and a higher throughput is selected as the most suitable one. Since it is required that a cluster is formed by nodes that are in a close range of each other, this clusters can have a significant number of nodes.

To perform the communication between the CHs and the Sink, a mode that is capable of achieving higher range communication (in exchange of a decrease in the throughput) is used. In this way it is achieved a significant coverage area that a Sink can handle. Although this approach may cause an increase in the end-to-end delay of the data packets, it is not an aspect that causes a major concern taking into account the type of data that is being treated. If the purpose of the network is to deliver the data with the minimum possible delay, a higher throughput mode can be used. However a larger number of Sink nodes should be deployed in order to cover the same area.

4.5 Media Access Control (MAC) in LoRa

In radio communications with a substantial number of nodes and simultaneous communications, a MAC protocol has to exist in order to guarantee that the network is reliable and that information exchange can occur. As said in Section 3.5, the developed MAC protocol for LoRa communications is based in the RTS/CTS message exchange.

4.5.1 MAC Messages

There are specific packets that have been created to handle the MAC process, which are included in the `LoRa MAC` packet class. This class is defined in the `Packet Type` field by the most significant byte with the value `0x02`. Belonging to this packet class there are three packets that are differentiated by the value of the least significant byte of the `Packet Type` field. These packets are described next.

Request to Send (RTS) When a node has reached the threshold of packets to send to the next node in the hierarchy, it starts by trying to access the media. For that, the node that wants to access the communication channel sends an `RTS` packet in broadcast. The `Destination Address` field of the packet's header is set with the address of the node with whom it wants to communicate. The `RTS` message payload is composed by three bytes, where one byte refers to how many packets are going to be sent, and the other two represent how much time, in seconds, it is expected for the communication to take. The `RTS` packet is defined with the value `0x01`.

Clear to Send (CTS) The `CTS` message is used by the node that receives an `RTS` to inform the requesting node that the media is free and it can send its data. The `CTS` is equally sent in broadcast with the address of the receiving node in the `Destination Address` field of the packet's header. The payload of the `CTS` packet is the same of the received `RTS`. The `CTS` message is identified with the value `0x02`.

Wait To Send (WTS) The created `WTS` message is used by the nodes that receive an `RTS` message during a reception of data packets from another node. When a node receives an `RTS` and is busy receiving data packets, it generates a `WTS` message and sends it in broadcast with the address of the requesting node in the `Destination Address`. The information carried in the `WTS` packet's payload is the same as in the `RTS` and `CTS`, with the difference that in the `WTS` it is indicated the number of remaining packets to receive and the remaining time that the other node has to complete the data transmission.

4.5.2 MAC Process

When a node has data to send, it starts by evaluating if the communication channel is available. For that purpose, the node begins by sending an RTS message in broadcast. As it was described in Sub-section 4.5.1, the RTS message carries in its payload information about the communication that the node wants to perform, more precisely, the number of packets to be sent and the expected time that the communication may take.

In order to get a time estimation, there are three variables that are needed to be taken into account:

- Number of packets to send;
- Mean packet size;
- Time on Air (ToA) of the packet.

The LoRa packet ToA is obtained using the data presented in Figure 4.8 and Figure 4.9. Figure 4.8 shows the values of the ToA for the different LoRa communication modes and for different packet sizes. In Figure 4.9 it is shown how the ToA varies depending on the packet size. With this information, it is possible to obtain a linear regression for each of the modes, from where it is possible to obtain the ToA depending on the packet size.

				time on air in second for payload size of						
LoRa mode	BW	CR	SF	5 bytes	55 bytes	105 bytes	155 Bytes	205 Bytes	255 Bytes	max thr. for 255B in bps
1	125	4/5	12	0.95846	2.59686	4.23526	5.87366	7.51206	9.15046	223
2	250	4/5	12	0.47923	1.21651	1.87187	2.52723	3.26451	3.91987	520
3	125	4/5	10	0.28058	0.69018	1.09978	1.50938	1.91898	2.32858	876
4	500	4/5	12	0.23962	0.60826	0.93594	1.26362	1.63226	1.95994	1041
5	250	4/5	10	0.14029	0.34509	0.54989	0.75469	0.95949	1.16429	1752
6	500	4/5	11	0.11981	0.30413	0.50893	0.69325	0.87757	1.06189	1921
7	250	4/5	9	0.07014	0.18278	0.29542	0.40806	0.5207	0.63334	3221
8	500	4/5	9	0.03507	0.09139	0.14771	0.20403	0.26035	0.31667	6442
9	500	4/5	8	0.01754	0.05082	0.08154	0.11482	0.14554	0.17882	11408
10	500	4/5	7	0.00877	0.02797	0.04589	0.06381	0.08301	0.10093	20212

Figure 4.8: Lora Time on Air for different packet sizes, from [65]

The mean packet size can be obtained with the queue status information, such as, the total number of packets in the queue and the accumulated size of these packets in bytes.

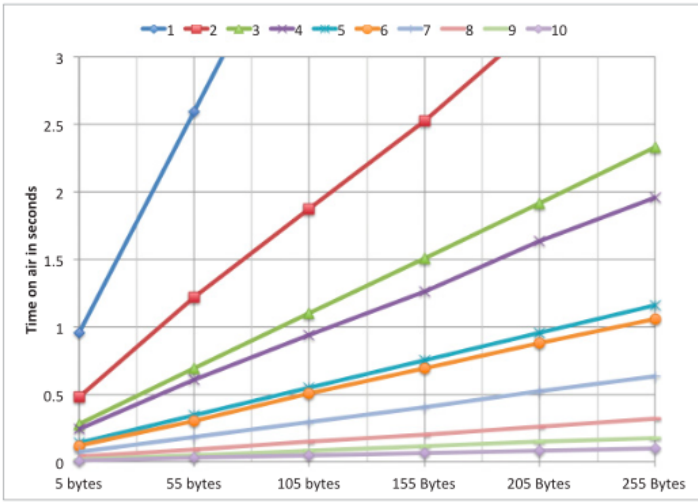


Figure 4.9: Lora Time-On-Air with payload size variation, from [65]

With these two variables known it is possible to estimate a mean packet size. The number of packets to send is defined by the node's transmission threshold.

Thus, it is proposed the Equation 4.1 to determine the expected communication time between the nodes.

$$conn_{time} = \lceil (n_{packets} \times ToA_{meanPacket} + n_{ACKs} \times ToA_{ACK} + n_{packets} \times SIFS) \times 1.5 \rceil \tag{4.1}$$

Where each variable is explained next:

$\boldsymbol{conn_{time}}$ Is the obtained value for the communication time, in seconds and rounded up to the nearest integer value;

$\boldsymbol{n_{packets}}$ Indicates the number of data packets that are going to be transmitted;

$\boldsymbol{ToA_{meanPacket}}$ ToA relative to the mean packet size of the data queue items.

$\boldsymbol{n_{ACKs}}$ Indicates the number of ACK packets that are supposed to be received.

$\boldsymbol{ToA_{ACK}}$ Represents the ToA that the ACK packet takes.

$\boldsymbol{SIFS}$ SIFS is the time used between each data packet sending, and for that reason, it needs to be taken into account. The SIFS value is defined as 0.5 seconds.

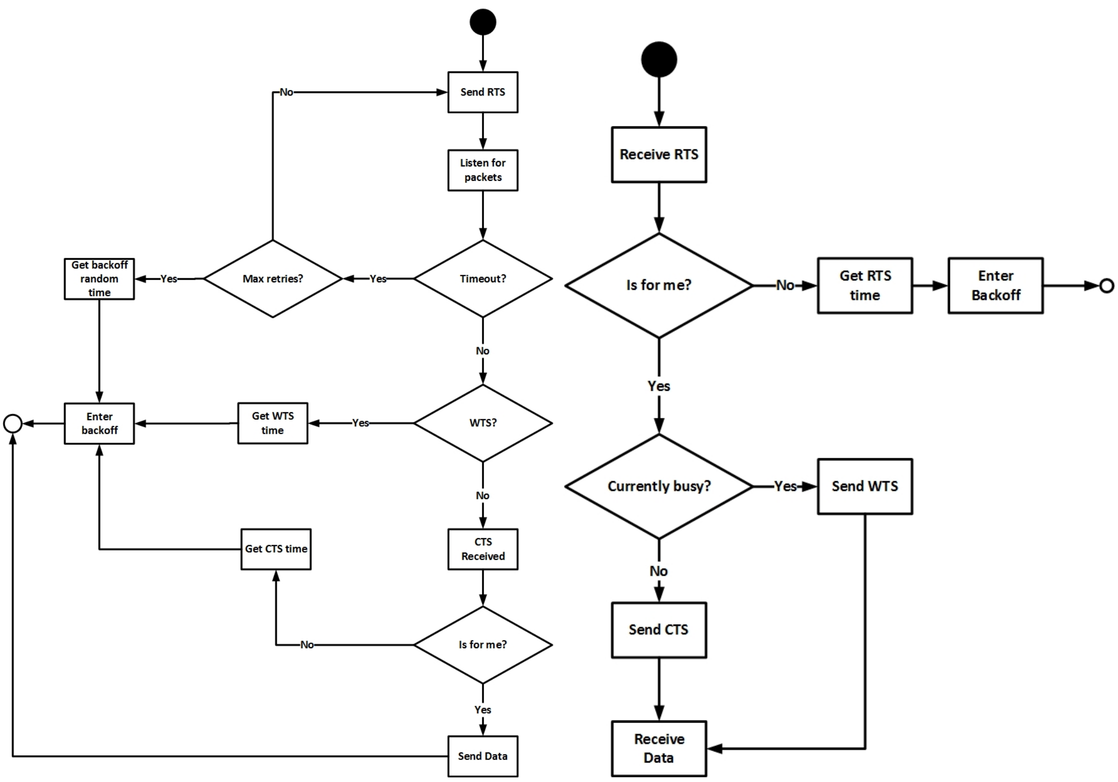
At the end of the sum of all time portions, this value is multiplied by 1.5 to give a 50% margin in case some packet may need to be retransmitted. In fact, the time present in the RTS/CTS packets works like a time limit for the nodes to perform the communication.

After the connection time has been determined and the packet is built, the message exchange between the nodes begins. First, the requesting node starts by sending an RTS packet with the destination node address, then it waits for a response from the receiving node, that can be a CTS or a WTS. If the reception for a response gives a timeout, the node resends the RTS packet to a limit of three retries, after which it enters in the backoff state for a random time.

If a WTS message is received by the requesting node, then this node enters in the backoff state for the time specified in the message's payload and tries to access the media later, after the backoff time expires. If otherwise, a CTS is received, the node verifies if the destination address is its own address and if it is effectively a response for the RTS sent, then, it begins the data transmission.

From the receiver side, when an RTS is heard, first it is checked the destination address. Then, if the RTS is not destined for that receiver, it gets the time that the communication may take and goes to the backoff state. If the RTS is destined for that receiver, it can answer with two messages a CTS, if it is not busy; or a WTS, if it is already handling a communication.

The media access flow from the requester and receiver sides can be seen in Figure 4.10.



(a) Channel access sender side (b) Channel access receiver side

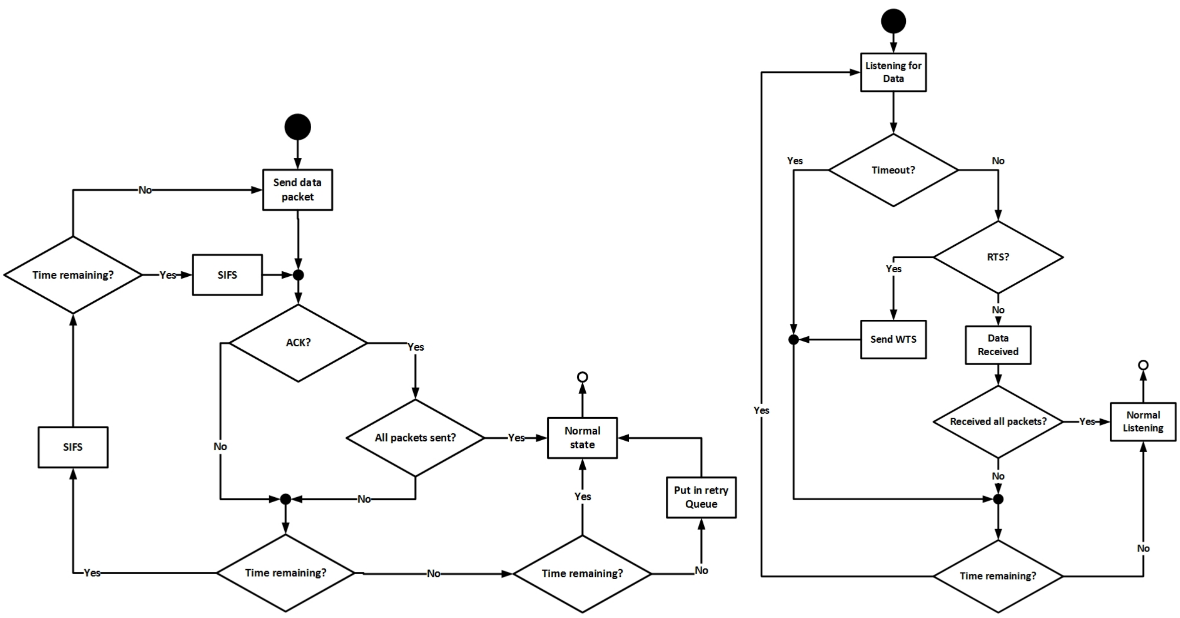
Figure 4.10: Channel access process

When a node is granted with the access to the media, it can begin the data packets transmission. The data packets are sent in unicast and their correct reception is confirmed with an acknowledgment. After a node sends a data packet, it waits to receive the correspondent ACK message. If the ACK is not received, the node verifies what is the time remaining in the communication time. If there is still time remaining, the node waits SIFS and retransmits the packet. If the node run out of time and can not be able to deliver the packet, this packet is marked as a *Retry Data Packet* and in the next contact it is on the top of the queue to be sent.

If the data packet is successfully delivered, the node verifies if all the packets that were defined to be sent had already been sent. If so, the listening state is adopted. In case there are still packets to be sent and there is time remaining to continue with the communication, the next packet is tried to be sent after a SIFS. If the expected time expires before the node can deliver all the packets, the remaining packets stay in the queue and the node goes to the listening state.

When a node informs a requester that it is free to receive its data packets, the receiver enters in the reception mode for data packets. If after the reception slot a timeout is given, the receiver verifies if there is time remaining and if so, begins a new reception slot. In case that a packet is received, the receiving node checks its type. In case it is an RTS from another node, a WTS message is sent to the requesting node. When the received packet is a data packet, the receiver proceeds to a new reception slot if there are still packets to receive and time remaining. Otherwise, the node goes back to the listening state for new communications.

The data transmission process flow is presented in Figure 4.11.



(a) Data sender side (b) Data receiver side

Figure 4.11: Data exchange process

4.6 Data Management

There are two distinct types of data that can be gathered by the nodes, sensors data and camera images. Each of these data needs to be treated and maintained in the node while it is not transmitted. For that purpose, specific data messages for the sensors data and strategies to store the data locally were implemented.

4.6.1 Data Messages

As it was described in Section 3.3, the sensors data acquired are divided into two types, environmental data and weather data. In order to identify data messages, a `Data` packet class is created and identified in the `Packet Type` field by the most significant byte with the value `0x03`.

Belonging to this class are two packets that have the same payload organization. The payload starts with a 2-byte integer that carries the data Sequence Number (SN), and the following data refers to the sensors acquired data.

Figure 3.5 showed the organization of the information acquired by the sensors. This information is composed by: a 1-byte integer that represents the data type; an 8-byte float value that contains the timestamp of when the data has been acquired; a 2-byte integer that indicates the data acquisition period in seconds; and a sequence of a 1-byte integer followed by a 2-byte integer that refers to the sensor ID and its measurement, respectively. Thus, the two messages included in the `Data` class that are differentiated by the value of the least significant byte of the `Packet Type` field, are:

Environmental Data The `Environmental Data` message carries the information about the temperature, humidity and luminosity and it is identified with the value `0x01`.

Weather Data The data that refers to the weather category such as, precipitation, wind direction and wind speed, are sent in the `Weather Data` message, that has the value `0x02` in its packet type.

There is another type of sensors, the image gathering cameras, that are directly classified and managed by the Connection Manager, and therefor, they are not part of these data messages.

4.6.2 Data Maintenance

The data acquired during the gathering process needs to be maintained locally while it is not transmitted. For this purpose, a priority based organization is used, where three priorities are defined as:

Data Packets This is the standard priority used for any incoming or generated data packet.

Retry Data Packets This priority is used for the packets that have been sent but the acknowledgment has not been not received. When this happens, the packet is put back in the queue with a higher priority in order to be one of the first packets to be sent in the next data transmission.

Urgent Packets This priority is the highest possible in the queue and is used for any urgent packets that forcibly need to be delivered as soon as possible.

With the priorities defined, three queues are created, one for each priority. These three queues are then aggregated, and a layer of abstraction is created in order to manage the data as if it was just one priority queue. It is provided an interface that permits the insertion/removal of packets and the state of the queue, more precisely, its size in number of elements and the accumulated size of the items in bytes. In Figure 4.12 it is exemplified how the queue is implemented.

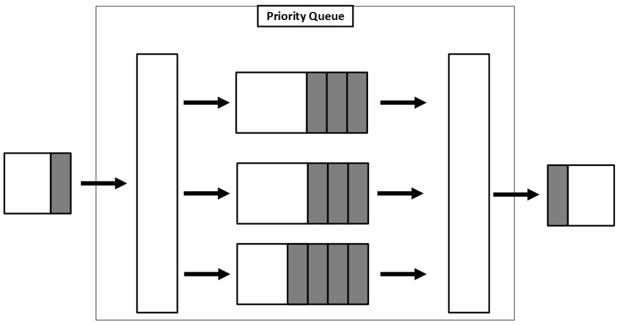


Figure 4.12: Priority Queue implementation

To store the data that is gathered, two priority queues are created: one for the sensors data and another for the camera images. The sensors data is stored with the final message already formed, including the header. Images gathered by the camera are saved in a folder at the node's mass storage, and in the image queue it is inserted the file's name along with the timestamp of when the image was taken.

4.7 Multi-Technology Connection Manager

The proposed *Connection Manager* is composed of three inner blocks, the *Network Manager*, *WiFi Manager* and *LoRa Manager*. Each one of these blocks is executed in a different thread, that are launched once the cluster formation ends. In order to maintain the thread synchronization and to signal the technologies state changes, two `Lock` objects are created.

LoRa Lock This primitive `Lock` is used to indicate the state of the LoRa communication channel. The LoRa `Lock` is locked when information is being sent through the LoRa radio. This `Lock` state is controlled by the *LoRa Manager* thread.

WiFi Lock The *Network Manager* thread is responsible for managing the `WiFi Lock` state. When locked, this `Lock` indicates that the node is connected to a WiFi network.

4.7.1 Network Manager

The *Network Manager* is responsible for managing the WiFi interface. This sub-module of the *Connection Manager* performs network scanning, as well as the connection/disconnection operations and the continuous network evaluation during the time the node is connected.

Figure 4.13 presents the *Network Manager* flow. The *Network Manager* is constantly scanning for available networks. Within the available networks it searches for specific Service Set Identifiers (SSIDs) that match the ones that can be used. If none of the available networks is a valid network, a new scan is performed within a certain period, in this case a scanning period of five seconds is adopted. When a valid network is found, the node gets its signal and if it is above a certain threshold, it will try to connect to that network; otherwise, the network is discarded.

If the node is able to connect to a network, that is, an IP address has been assigned by the AP, the `WiFi Lock` is acquired and the network is monitored with a certain periodicity. When the network is still in range, its signal is gathered and it is checked if it is above the defined threshold. Once the network's signal goes below the threshold or the network goes out of range, the node performs the disconnect action, releases the `WiFi Lock` and returns to the scanning state.

4.7.2 WiFi Manager

The *WiFi Manager* sub-module is responsible for managing how the waiting data should be handled when the node is connected to a WiFi network. In Figure 4.14 it is presented the *WiFi Manager* process flow. When the node is connected to a WiFi network, that is, the `WiFi Lock` is in the locked state, the node verifies which type of data it is acquiring. If the node is collecting camera images, it will dedicate the WiFi interface to send these images and leave the sensors data to be handled by the LoRa interface.

While the network is still available and there are images to send, the node sends the images waiting in the queue through a Transmission Control Protocol (TCP)/Internet Protocol (IP) connection. If an ACK message from the server side is not received, the node tries to resend the image while it is connected to the network. If eventually the node disconnects from the network in the meantime, the image is put back in the queue with the retry priority. Once an image is delivered to the server, it is deleted from the node's mass storage.

In case that sensors data is the only data being collected, the same procedure that is applied to the images is now applied to the sensors data. A TCP/IP connection is created between the node and the server as well.

While there are data packets to be sent, the node sends them until they are all dispatched or the `WiFi Lock` changes to the unlock state, meaning that the network is dis-

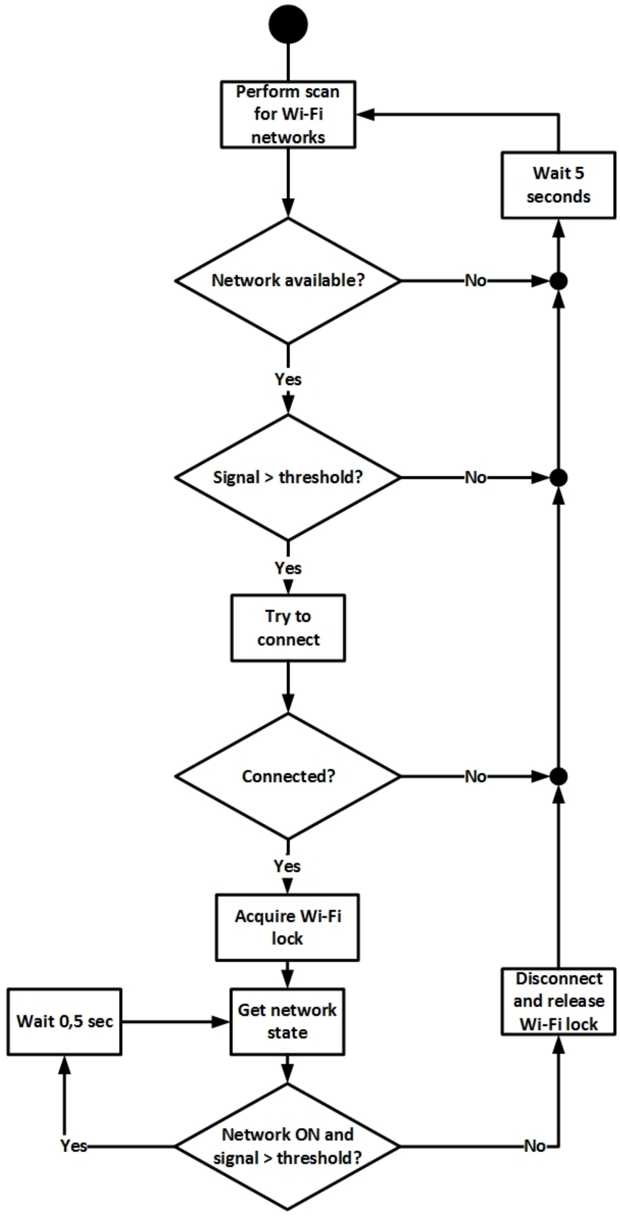


Figure 4.13: Network Manager Flow

connected.

4.7.3 LoRa Manager

The *LoRa Manager* is responsible to handle the LoRa radio interface. It is the *LoRa Manager* that defines if the interface should be in the listening or in the sending data mode. In Figure 4.15 it is presented the flow chart of the *LoRa Manager* behavior. By default, the LoRa radio is on the receiving state and listening for packets that come from

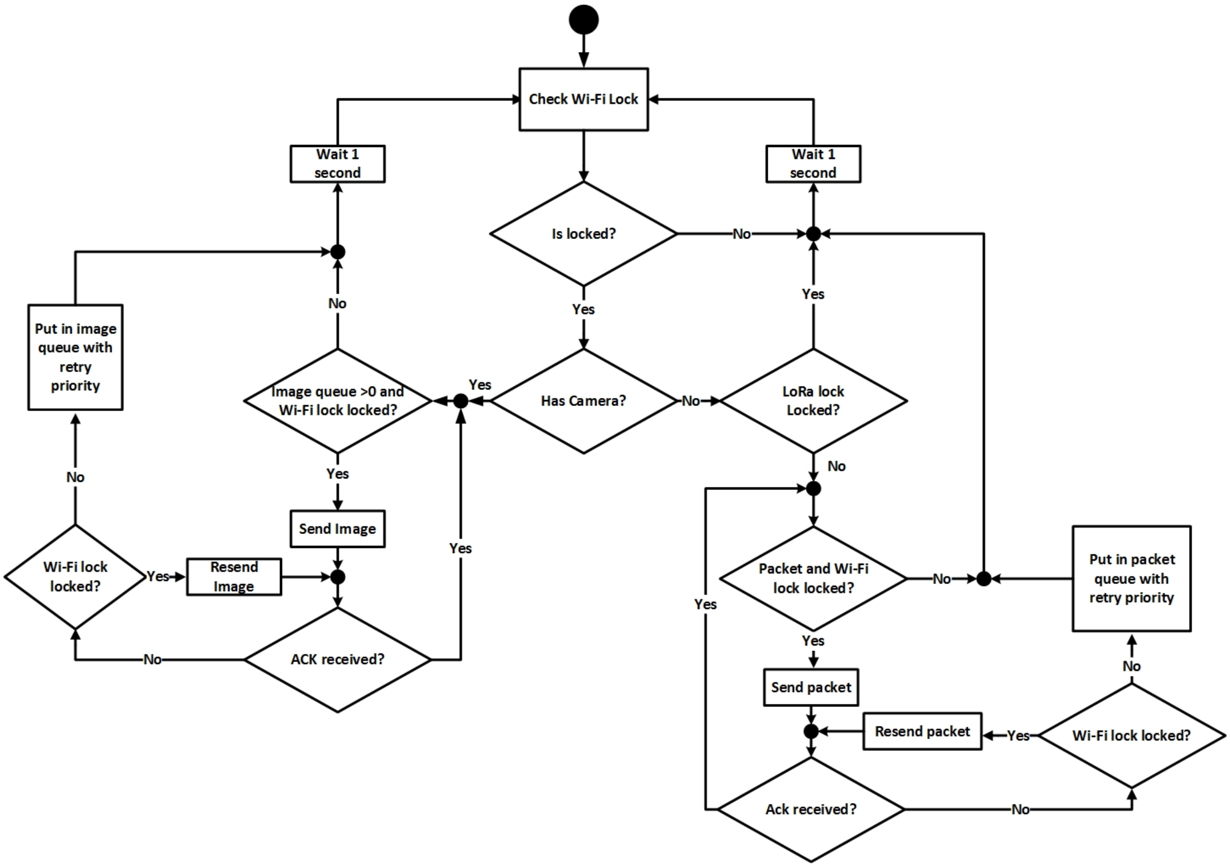


Figure 4.14: Wifi Manager Flow

nodes of its cluster. The node listens for an incoming packet during a defined time slot. If a packet arrives during this time slot, it will be handled by the MAC module since it will be one of the three possible MAC packets.

In case a timeout is given by the receiving action, the manager checks the `WiFi Lock` state. If it is locked and the node is not collecting images, then a new receiving time slot is initiated. When neither `WiFi Lock` is locked nor the node is getting image data, the node verifies if it is on a backoff state, going back to the receiving state in that case.

If the node has all the conditions to be able to communicate, it checks if the number of packets waiting in queue to be sent has already reached the defined threshold. While there are not enough packets to be sent, the node continues on receiving packets. Once the threshold has been reached, it may be necessary to change the communication mode to the long range mode depending on the node's role. In case the node is the CH, it will send its packets to the Sink, which communicates in the long range mode. Therefore it needs to configure the radio to the long range mode.

With the communication mode set, the node starts by trying to access the media. If it is not succeeded it then configures the short range mode and returns to listening for packets coming from cluster nodes. If the node is granted with the access to the media, it acquires the `LoRa Lock` and starts sending its data. While in the sending process, in case that it is not collecting image data, after sending a new packet the node checks if the `WiFi Lock` has changed its state to locked. When the `WiFi Lock` state changes during the LoRa sending time, the manager stops the LoRa from sending, releases the `LoRa Lock` and goes to the listening state. The data sending is then handled by the *WiFi Manager*.

In case that all packets were sent by the LoRa interface without any occurrence in between, the node sets back the short range, if it is necessary, and restarts to listen for packets.

4.8 Chapter Considerations

This chapter presented the implementation of the modules that are necessary to perform the proposed solution.

First, it was presented the proposed message format that serves as base for the different messages implemented. Then it was explained how the LoRa nodes organize themselves in clusters taking into account its link to the Sink. It was also presented which messages are responsible to exchange the necessary information between the involved nodes.

The data management was another module that was developed. Using as base a priority queue, it was explained how the data was managed and maintained during its period in the node.

The MAC protocol for LoRa was detailed, including the messages created and the information that each message carries. It was also showed how the nodes act according to the type of messages they receive and the role they play in the organization.

Finally, the Connection Manager has the responsibility to coordinate the different communication interfaces. For this purpose three sub-modules were developed so they can cooperate between them to decide which is the interface that should perform the data delivery, depending on the networks availability and the type of data.

The next chapter presents the evaluation of the implemented solution. It is also presented some tests on the coverage and range of the LoRa technology in different environment.

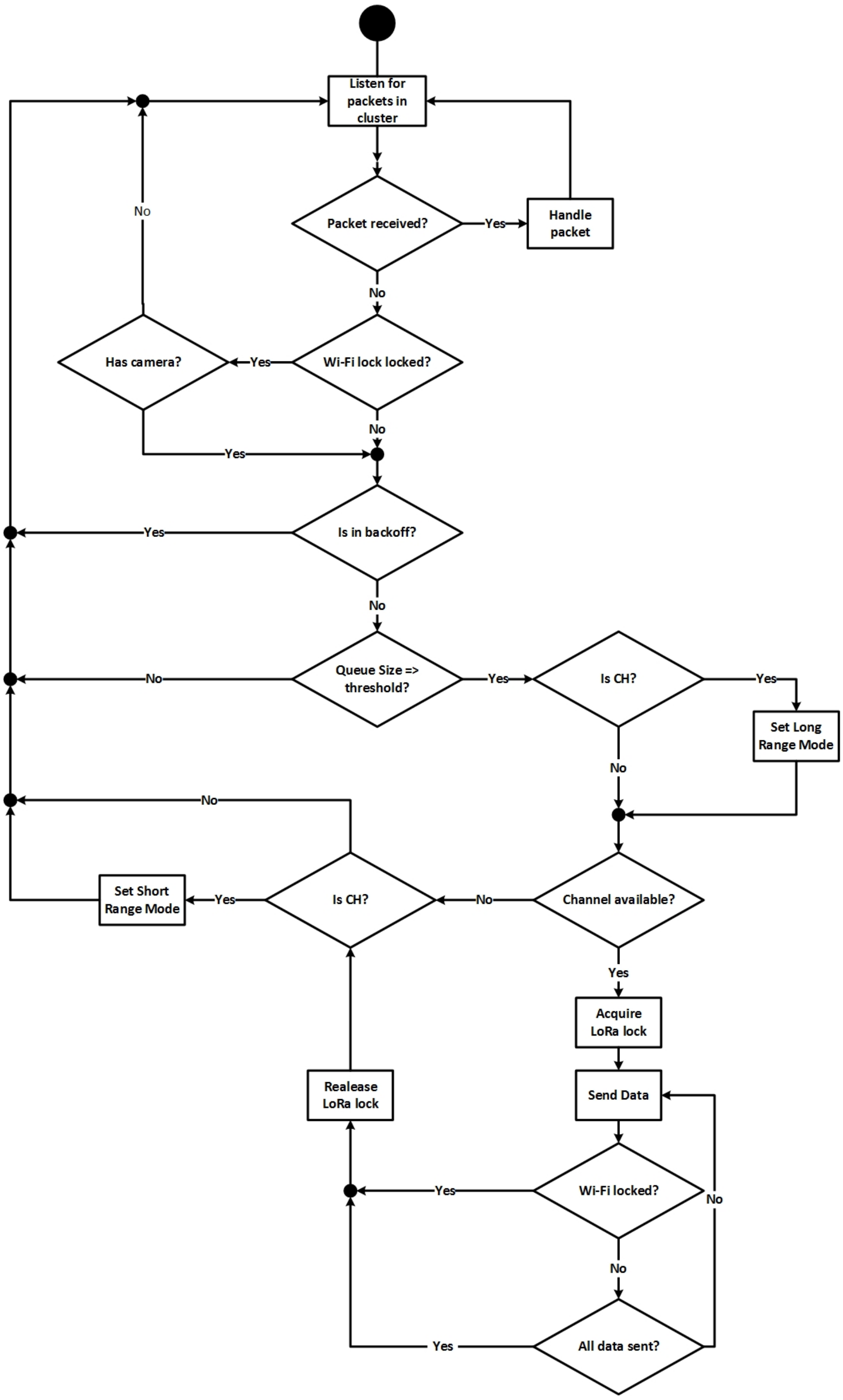


Figure 4.15: LoRa Manager Flow

Chapter 5

Integration and Evaluation

5.1 Introduction

This chapter describes the tests that have been performed to validate the proposed solution, as well as the obtained results. This chapter also describes the tests performed to the LoRa technology as well, in other to evaluate its range and quality in distinct scenarios.

Section 5.2 presents the results obtained for the different locations where LoRa has been used to transmit data.

Section 5.3 presents the different scenarios and tests performed in order to evaluate the proposed solution. Different situations were tested, such as, variation of the nodes and gathering period, presence of a WiFi network and image gathering. The obtained results are also presented for each scenario.

5.2 LoRa Range Tests

To evaluate the capabilities of LoRa in terms of its communication range and quality, several tests have been performed in different conditions.

The tests have been performed in three locations, one rural and two urban:

Alqueidão Located near Figueira da Foz, Alqueidão is a village with a rural environment, mainly a flatland. In this village it is possible to perform the tests without buildings obstructing the signal. Although direct LoS is not achieved, it is considered a good environment to test how the technology behaves in rural scenarios;

Coimbra It is one of the urban locations chosen to evaluate LoRa. In addition to being an urban scenario, Coimbra is also an uneven city with hills, which causes the testing points to be above or below the reference point, in some circumstances. The tests made were all in NLoS conditions;

Aveiro The second urban test location is Aveiro,which is, in contrast with Coimbra, a more flat city. This gives a different type of environment even though it is also a urban scenario. As in Coimbra, the tests have been made in NLoS conditions.

In each one of the three locations, three LoRa modes were tested, a short range mode, a middle-long range mode and a long range mode. The specifications of the modes used are presented in Table 5.1. At each testing point, it were transmitted 100 packets of 75 bytes from the fixed reference point. The number of transmitted packets is considered enough to get a viable mean RSSI and SNR; moreover, conclusions about the packet losses can be taken as well. The size of the packets is considered suitable for the applications that the solution approaches. The packets RSSI and SNR were gathered at the reception. The transmission power has been set to its maximum (14dBm), and a 4.5 dBi omni-directional antenna has been used.

Table 5.1: LoRa modes tested

Mode	**BW (Hz)**	**CR**	**SF**	**Sensitivity (dB)**
1	125	4/5	12	-134
3	125	4/5	10	-129
10	500	4/5	7	-114

5.2.1 Alqueidão Range Tests

To evaluate the LoRa range and quality in a rural scenario, it were defined seven testing points with increasing distance.

In Figure 5.1 it is shown the map of the region where the tests were made, as well as the points where packets were collected. Table 5.2 presents information about the distance from each point to the fixed reference. In Table 5.3 it is also shown the elevation of each testing point.

Table 5.2: Test points distance to fixed reference (Alqueidão)

	Points						
	P1	P2	P3	P4	P5	P6	P7
Distance to reference point (meters)	572	1210	1470	2890	4030	5240	5660

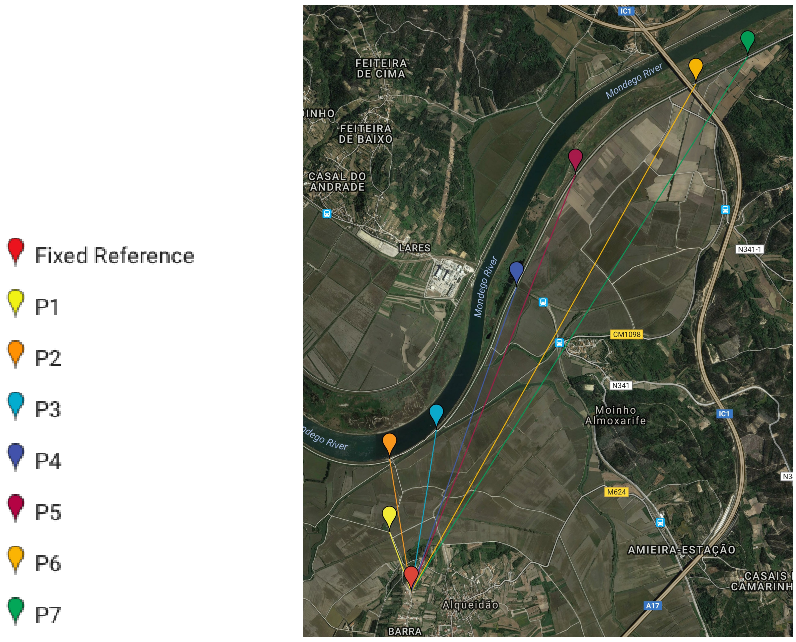


Figure 5.1: Alqueidão Range Test Points

Table 5.3: Test points elevation and difference to fixed reference (Alqueidão)

	Points							
	Fixed Ref.	P1	P2	P3	P4	P5	P6	P7
Elevation (meters)	34.0	0.9	4.1	4.4	2.3	5.9	2.4	2.8
Difference to reference point (meters)	-	-33.1	-29.9	-29.6	-31.7	-28.1	-31.6	-31.2

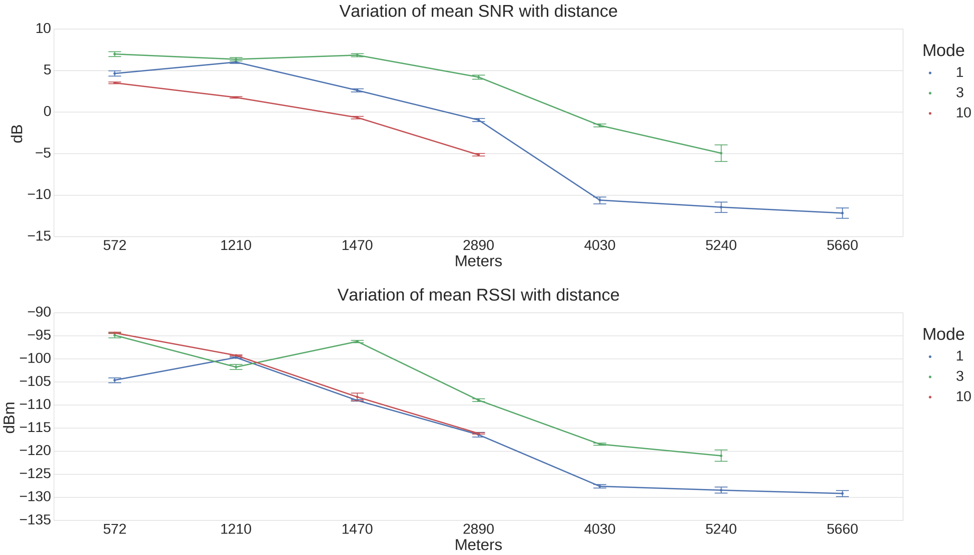


Figure 5.2: LoRa SNR and RSSI Variation (Alqueidão)

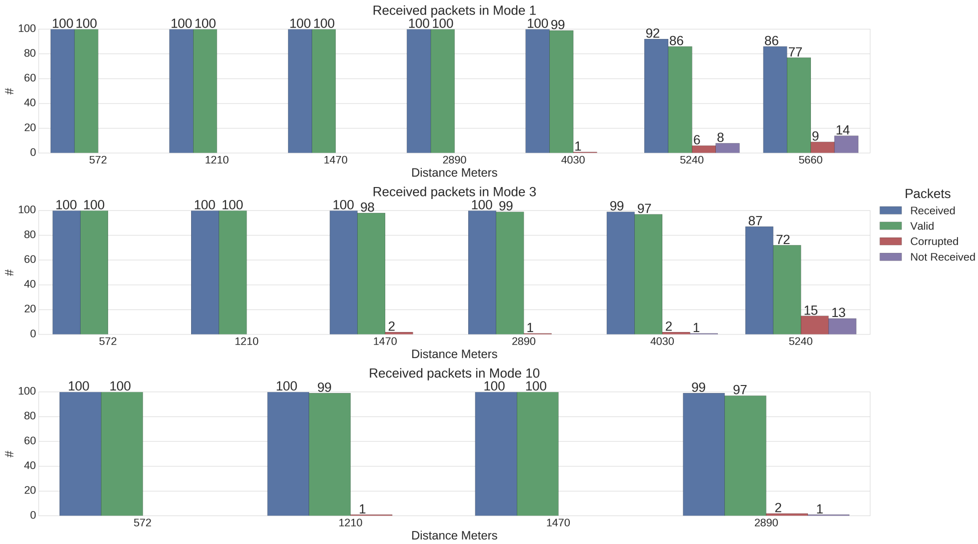


Figure 5.3: LoRa Received Packets (Alqueidão)

Figure 5.2 depicts the LoRa SNR and RSSI results for different ranges and transmission modes. After analyzing the Figure 5.2, it is noticeable that the three modes showed different limits in their communication range. This happens due to their different sensitivity, since mode 1 presents the best sensitivity. It was expected that mode 1 showed the highest range and mode 10 the lowest, as shown in Figure 5.2.

It is possible to achieve a maximum range of 5660 meters with mode 1, although, at the extreme distances, some packets were not received or were corrupted, as it is shown in Figure 5.3. Mode 3 also gives good range coverage, with a maximum distance of 5240 meters, even though only 72% of the packets were received correctly at that distance. Mode 10 presents the lowest range, but it still is a decent range taking into account that it is the mode that permits a higher throughput.

With these tests it was possible to conclude that, even without LoS conditions and with some obstacles in the way (such as trees and at some points bridges), LoRa can be a good technology for long range communications in a rural scenario.

5.2.2 Coimbra Range Tests

For testing LoRa in the first urban location, it were defined 12 testing points. Each point can have distinct characteristics, such as: number of buildings that the signal goes through; height of the buildings that are in the way of the signal, that could interfere, or not, with the signal; elevation difference for the reference point, that could be above or below.

In Figure 5.4 it is shown the map of the region where the tests were made, as well as the testing points. Table 5.4 presents information about the distance from each point to the fixed reference. In Table 5.5 it is also shown the elevation of each testing point. It is important to refer that the LoRa module at the reference point, was placed at a height of ten meters above ground, which is already taken into account for the elevation value of the reference point.

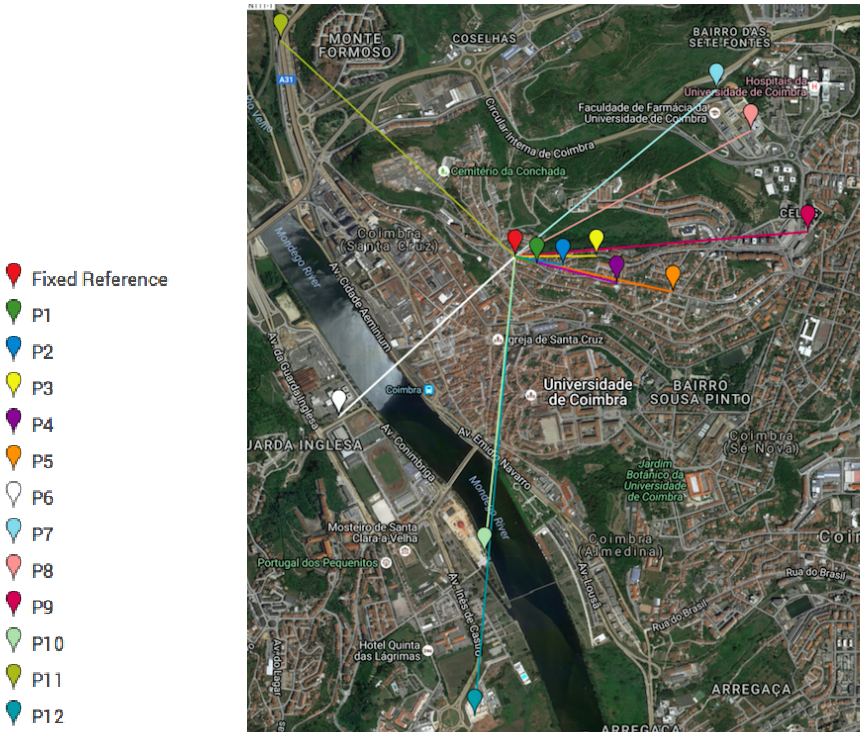


Figure 5.4: Coimbra Range Test Points

Table 5.4: Test points distance to fixed reference (Coimbra)

	Points											
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Distance to reference point (meters)	100	210	351	452	698	1030	1130	1150	1270	1290	1370	1990

Table 5.5: Test points elevation and difference to fixed reference (Coimbra)

	Points												
	Fixed Ref.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Elevation (meters)	96.7	92.6	97.8	99.8	105.5	117.9	21.0	75.0	98.8	112.3	20.6	23.7	22.7
Difference to reference point (meters)	-	-4.1	1.1	3.1	8.8	21.2	-75.7	-21.7	2.1	15.6	-76.1	-73.0	-74.0

Once analyzed the results for this scenario presented in Figure 5.5 and 5.6, the first conclusion to take is that distance does not mean a worst signal, as happened in the rural scenario.

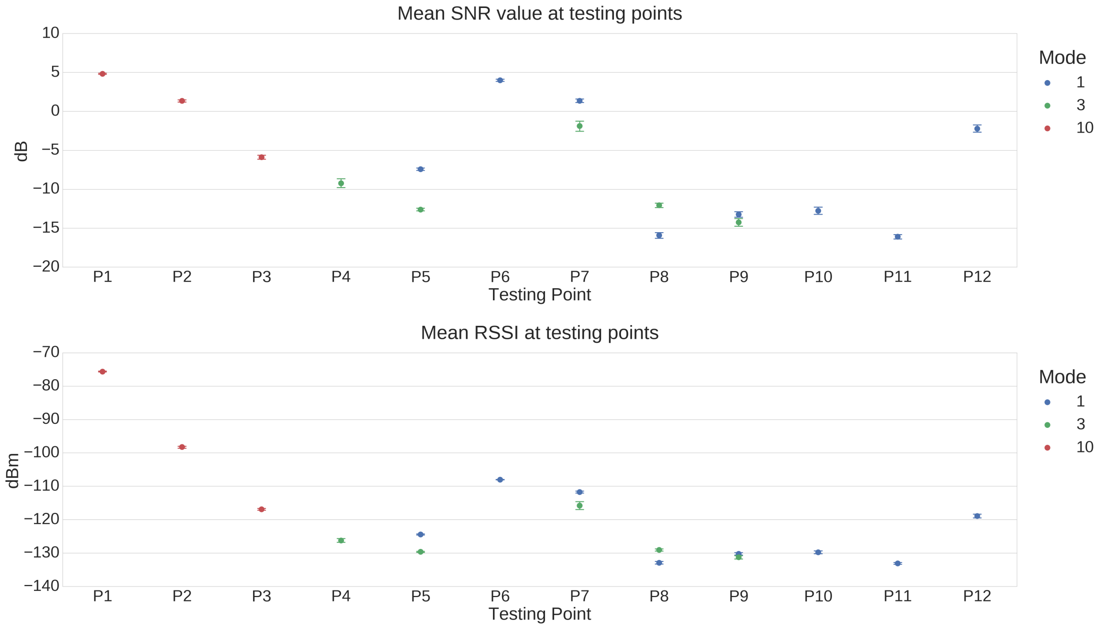


Figure 5.5: LoRa SNR and RSSI Variation (Coimbra)

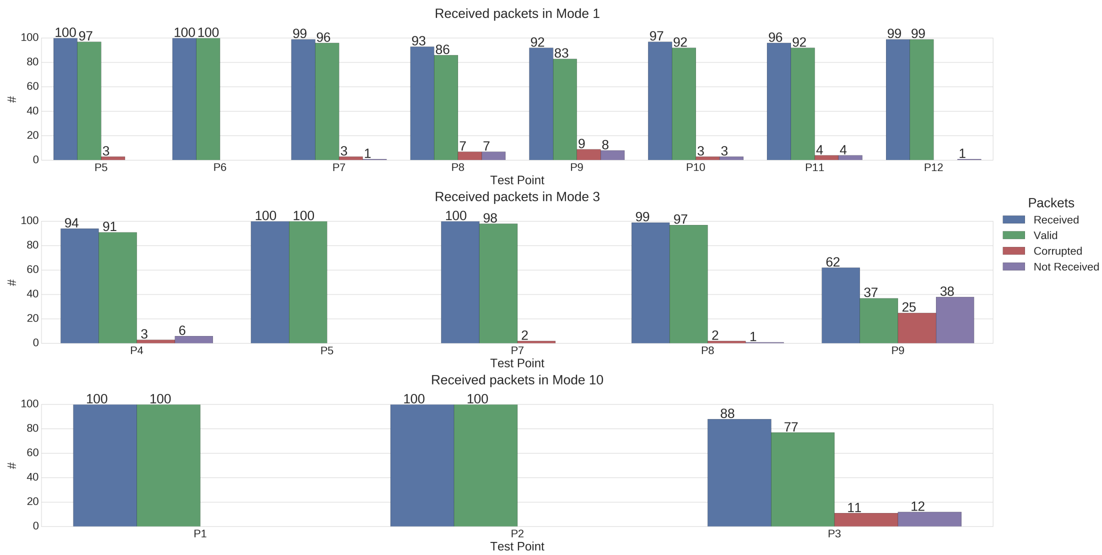


Figure 5.6: LoRa Received Packtes (Coimbra)

Starting with the shorter range mode, a maximum distance of 513 meters was achieved, which compared to the 2890 achieved in the rural scenario is a considerable decrease. However, this can be explained due to the fact that the testing locations are surrounded by buildings, which causes a major attenuation to the signal. Excluding P1, the other two testing points have a considerable number of buildings in the way, which allied to the low elevation difference, causes that forcibly the signal has to pass through all the building to reach the receiver.

For mode 3, distinct results were achieved for testing points that were practically at the same distance to the reference point. This can be explained by the conditions of each testing point. Looking at the P7 and P8 results it is visible a clear difference in the signal quality. Analyzing the conditions of each point by looking into the map, it is noticeable that P8 is surrounded by buildings while P7 is in a more open space. The elevation difference could be an influence as well, since P7 is some meters below the reference point and P8 is at almost the same elevation. This could cause a significant difference in the quality of how the signal is received, leading to distinct results for the same distance.

Testing point P9 was the point where poorer results were achieved, by receiving 62 of the 100 packets, however, from this 62 only 37 were valid, being the remaining 25 corrupted. This could be caused by the higher elevation of this point allied to the number and height of the buildings that are in the path of the signal.

Communication mode 1 was the mode that provided the highest range, with communication at 1990 meters distance from the reference point. With mode 1 there are some results that may seem contradictory, such as, the results for P10 and P12. Even though these points are almost in the same path line and P12 is farther from reference than P10, P12 presents better quality in the packets received. This could be caused by P10 being closer to the reference point, which allied to the elevation difference causes that the signal path to the receiver is more bent. One point that was identified during the range tests is that, for points in similar conditions, even a slightly difference on the elevation of the testing point can make a difference in the quality that the signal is received. This is another reason that can justify the difference of results from P10 to P12.

From these tests it is possible to conclude that not only the distance has impact in the signal's attenuation, but also the different conditions of the testing points have a significant influence in the final results. Also, LoRa can provide a good coverage in urban areas, even with all constraints associated.

5.2.3 Aveiro Range Tests

The third testing location is Aveiro, which by being a more plane city does not present the same constraints as Coimbra. To evaluate the range and quality of LoRa, 9 testing points were defined.

In Figure 5.7 it is shown the map of the region where the tests were made, as well as the testing points. Table 5.6 presents information about the distance from each point to the fixed reference. In Table 5.7 it is also shown the elevation of each testing point. It is important to refer that the LoRa module at the reference point was placed at a height of six meters above ground, which is already taken into account for the elevation value of the reference point.

Figure 5.8 and 5.9 shows that a range of 2110 meters can be achieved with both mode 1 and 3, and a range of 773 meters is achieved with mode 10. Also, it is possible to see that in this urban scenario, the results follow the tendency of the quality decreasing with the increasing of the distance.

Figure 5.7: Aveiro Range Test Points

Table 5.6: Test points distance to fixed reference (Aveiro)

	Points								
	P1	P2	P3	P4	P5	P6	P7	P8	P9
Distance to reference point (meters)	456	472	513	681	773	1180	1380	1460	2110

Table 5.7: Test points elevation and difference to fixed reference (Aveiro)

	Points									
	Fixed Ref.	P1	P2	P3	P4	P5	P6	P7	P8	P9
Elevation (meters)	12.7	3.8	16.7	12.7	16.0	10.9	19.8	23.4	3.4	14.6
Difference to reference point (meters)	-	-8.9	4.0	0.0	3.3	-1.8	7.1	10.7	-9.3	1.9

For mode 10, P3 is the point that presents the best quality, even though it is not the closest point. This can be explained by the fact that this point has a clear surrounding area and few obstacles in the signal path.

As for the modes 3 and 10, P6 shows a clear discrepancy to the tendency of the other points. This could happen due to the location where the test was made, since right in front of the testing point, there is a tall building in the signal path. Also, this is the testing point that presents a higher difference in terms of elevation to the reference point, and the fact that this point is at a higher elevation can also contribute to the obtained results.

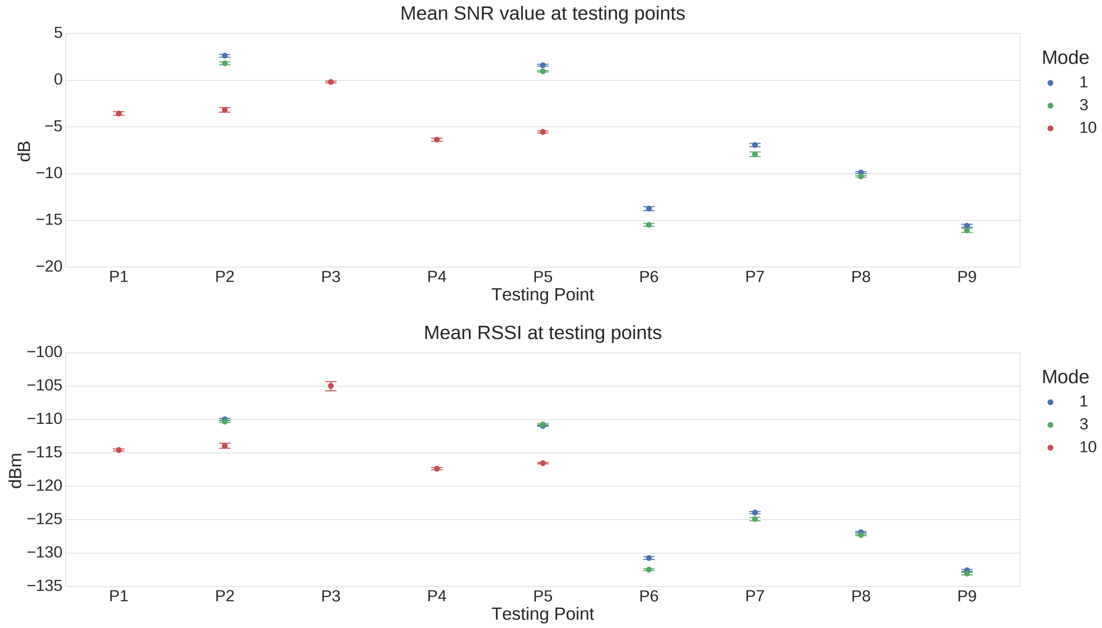


Figure 5.8: LoRa SNR and RSSI Variation (Aveiro)

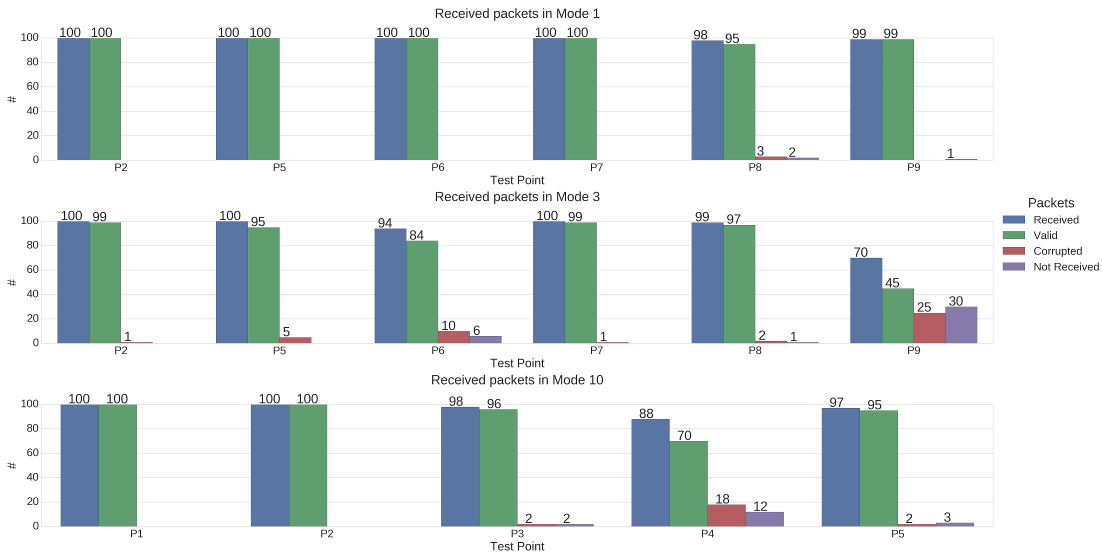


Figure 5.9: LoRa Received Packtes (Aveiro)

With the achieved range of 2 kilometers, it is possible for LoRa to cover almost all the center of Aveiro, which being an urban area with a considerable number of buildings shows the efficient signal penetration of the LoRa modulation. Also, it is possible to say that with these results a network could be deployed with a reduced number of gateways to cover all the desired area.

5.3 Data Gathering Tests

To validate the proposed solution for a data gathering implementation with communication technologies such as LoRa and WiFi, different scenarios were defined. These scenarios include: only LoRa communication between the nodes, and an evaluation of how the number of nodes and the data gathering period can influence the proposed solution; the changes that the presence of a WiFi network can make in the whole process and the consequent results; image data collection and consequent traffic differentiation that can lead to simultaneous data transmissions from the different technologies.

5.3.1 Testbed Description

This section presents the testbed organization and the important details during the analysis of the results.

The testbed used to perform the evaluation tests is presented in Figure 5.10. It is composed of two DCUs, one will be the CH and the other will be the member node, and a Raspberry Pi 2 with a SX1272 LoRa module that will have the Sink role. Each module is running Raspbian 7 and the DCUs have an IEEE 802.11n WiFi dongle. The Sink is connected to the Instituto de Telecomunicações (IT) intranetwork through Ethernet. In the testbed it is also included a laptop that is used as a server and will act as the data end point. This laptop is also connected to the IT intranetwork, making it possible for the Sink to deliver the received packets to the server through a TCP/IP connection.

All the Raspberry Pis have Python 3.5 installed in its Raspbian distribution to execute the developed modules.

At some specific tests, one of the DCUs will have an USB camera connected to provide image data gathering. There will also be a device that will act as a WiFi AP in some of the tests.

5.3.1.1 Node Simulation

As it was presented in Sub-Section 5.3.1, there are only two DCUs available, which makes the scenarios very limited. In order to surpass that constrain, it was implemented a node simulator that will be used by the CH and the Sink when it is desired. This module is used to simulate nodes inside a cluster when it is used by the CH, and simulates clusters when it is used by the Sink.

It is important to have this type of possibilities of increased complexity, in order to evaluate the functionalities of the MAC process, as well as the impact of the number of nodes per cluster and the number of clusters.

The number of nodes per cluster is defined in the DCUs as an argument of the program that is going to be executed; the same applies for the Sink. With the number of nodes and cluster defined, the MAC process takes into account the total number of nodes. Thus, when a node in a cluster tries to send data to the CH, it sends an RTS. When an RTS

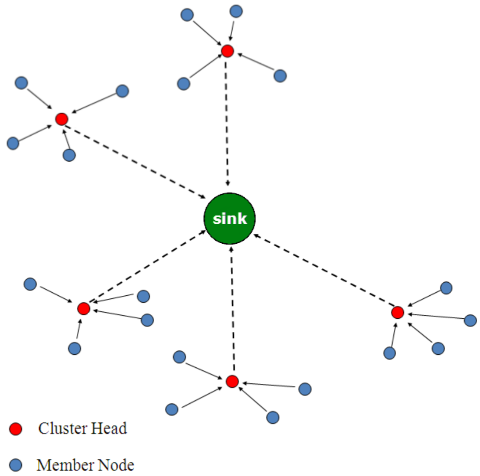


Figure 5.10: Testbed used in the experiments

is received by the CH, it generates a random number between 1 and 100. The generated number will be compared to a threshold that is defined by Equation 5.1.

If the random number is larger than the threshold, it is considered that the media will be granted to a simulated node, otherwise the CH responds with a CTS to the real node and receives its data. When the media is granted to a simulated node, the CH simulates that it is busy receiving data, by using the time that comes in the RTS message. It also generates random data packets with the same size of the real ones and in equal number of the received from the cluster member, that is needed in order to achieve the sending threshold of the CH.

$$CH_{threshold} = nodesAssocToCH \times \frac{100}{(totalNodes - 1) - servedNodes} \tag{5.1}$$

When a simulated node is served, it is updated the number of nodes already served. When the real node gains the media access and there are simulated nodes yet to be served, they are served when the real node leaves the media free.

The same procedure is applied to the Sink and the threshold is defined by Equation 5.2. The only difference is that in the Sink there is no need to generate random packets; however, the time spent on handling the incoming data is needed to be simulated.

$$Sink_{threshold} = existingCHs \times \frac{100}{totalCHs - servedCHs} \tag{5.2}$$

5.3.1.2 Scenarios common considerations

Several scenarios were tested, however, there are some characteristics that are common in most of the scenarios.

- Each scenario is tested ten times;
- The confidence interval is 95%;
- Each test had the duration specified by the *Test Duration* field in the table that contains the information about the scenario;
- After the cluster formation process ends (when applied), the data acquisition begins at the same time in the CH and the member node;
- The member node sending threshold is defined as two packets, which means that after a data acquisition, the member node will send its data to the CH;
- The CH sending threshold is defined by the number of nodes in the cluster multiplied by the member node sending threshold.

5.3.2 Influence of the Gathering Period

This test is performed in order to evaluate the impact that the data gathering period has in the behavior of the proposed solution. Two experiments have been made where the gathering period is changed. Table 5.8 presents the scenarios relevant information.

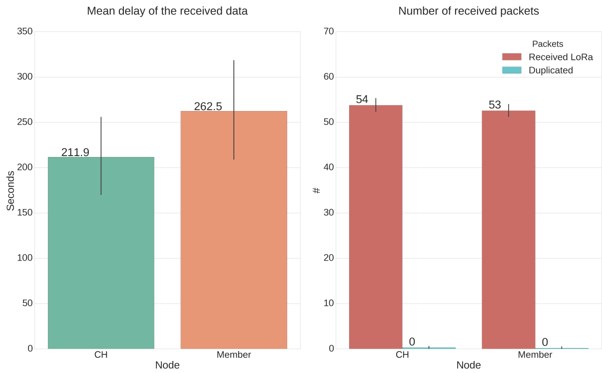
Table 5.8: Gathering period influence scenarios configuration

	Long Range Mode	**Short Range Mode**	**Gathering Period (s)**	**Nodes per Cluster**	**Number of Clusters**	**Total Nodes**	**Node Send Threshold**	**CH Send Threshold**	**Test Duration (min)**
120 seconds period	3	10	120	5	5	25	2	10	60
180 seconds period			180						90

The obtained results for each scenario are presented in Figure 5.11 to 5.14.

As it can be seen in Figure 5.11, the experiment with a 120 seconds gathering period shows a higher mean delay for the packets received at the server. The same occurs for the packet delay distribution (Figure 5.12), since for the same percentage of received packets, the value of the delay is higher in the 120 seconds case. In Figure 5.11 it is also presented the number of duplicated packets, which can occur, for example, when a CH is sending its data packets to Sink and the Sink receives a packet but the ACK message does not reach the CH.

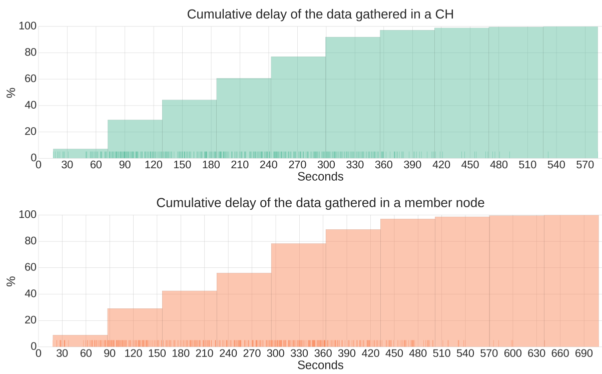
These results can be explained by the fact that the CH can not dispatch all its packets before a new data acquisition. In consequence, the packets will accumulate in the queue and packet delay grows linearly. That is proven by the results presented in Figure 5.14a.



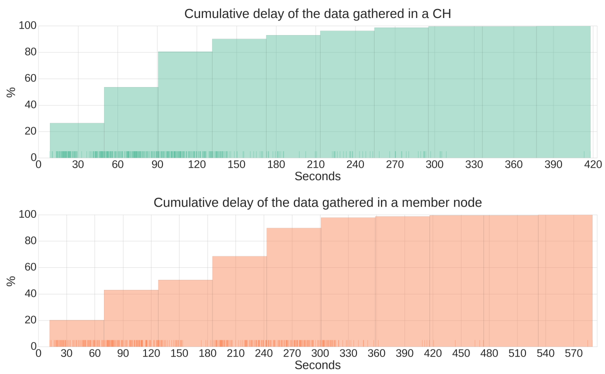
(a) Mean delay and packets received with 120 seconds period

(b) Mean delay and packets received with 180 seconds period

Figure 5.11: Mean delay and packets received with gathering period variation

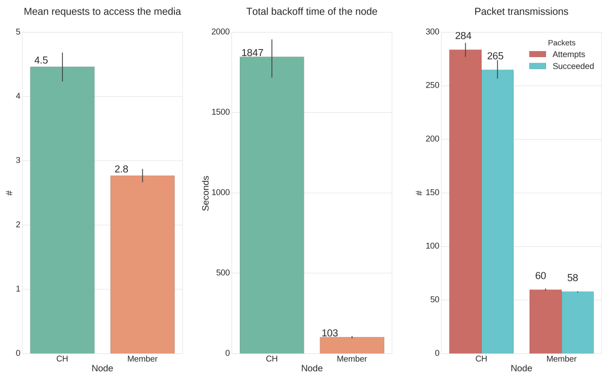


(a) Delay distribution with 120 seconds period

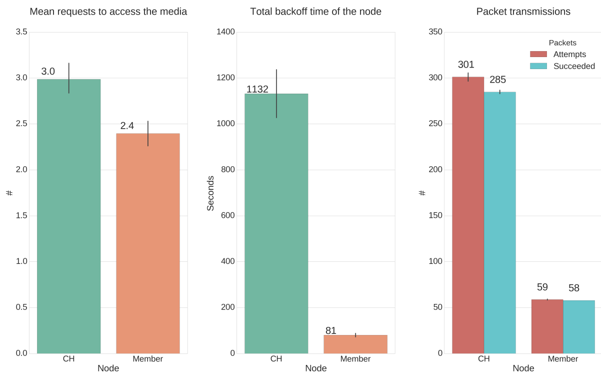


(b) Delay distribution with 180 seconds period

Figure 5.12: Delay distribution with gathering period variation

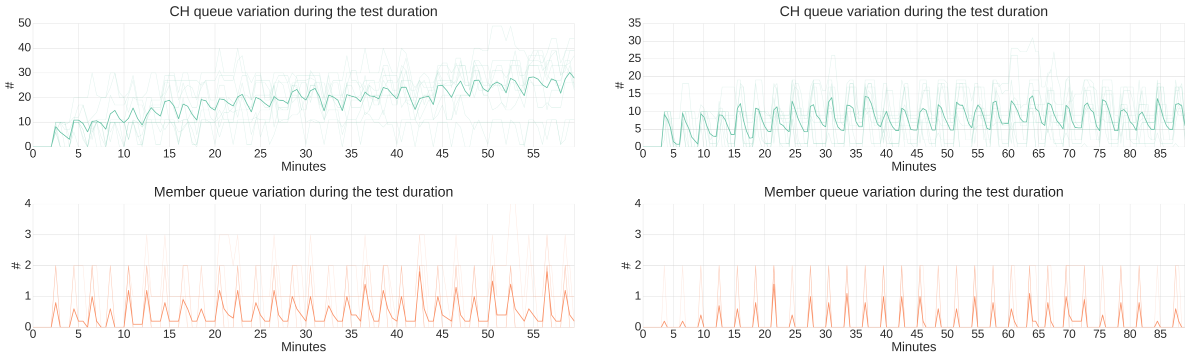


(a) MAC data with 120 seconds period



(b) MAC data with 180 seconds period

Figure 5.13: MAC data with gathering period variation



(a) Queue variation with 120 seconds period (b) Queue variation with 180 seconds period

Figure 5.14: Queue variation with gathering period variation

The reason why the CH can not deliver its packets before a new data acquisition can be explained by the actions that are taken between two consecutive data acquisitions. In Table 5.9 it is presented the actions duration, which helps to understand how this problem arises.

Table 5.9: Actions duration between data acquisitions

	Action			
	Access media and send data from member nodes	Access media to send to sink	Send data to sink (LoRa Mode 3)	Receiver slot timeout
Duration (s)	2	1.5	19	8

When the data acquisition ends, the CH starts to receive media access requests from the member node, in order to receive its acquired data. Since the cluster is constituted of five nodes, in which one is the CH, there are still four nodes that need to send their data to the CH. Since a member node can take two seconds to send its data, then, there are eight seconds required for the CH to receive the data from all nodes. Then, the CH tries to send its data to the Sink, but first it needs to try to access the media, which can take 1.5 seconds. Since there are a total of five clusters, the CH can require five tries to gain the media access, which translates to 4.5 seconds. A CH node can take up to 19 seconds to deliver its data packets to the Sink, taking into account that there are five clusters, a total of 95 seconds is needed for all the CHs to be served by the Sink.

Summing all the times of the actions taken, it gives a total of 107.5 seconds. Although that time is lower than the data gathering period, it is needed to take into account that some timeouts can occur during the packet reception slots, such as, when a node sends an RTS, and then waits for the correspondent CTS. If for some reason the CTS does not reach the requester, it will give a timeout on the requester side, which, in this case, will add eight seconds for each timeout that is given to the total time. With that been said, it is probable that at some points, new data arrives at the CH before the previous data has been delivered, leading to an accumulation of delayed transmissions and provoking the increase of the overall delay of the packets.

Another consequence of the constant queue increasing is the mean number of media access tries before the CH is granted with the media access (Figure 5.13). This happens because, when the CH sends a data aggregate and there are already enough packets in queue to exceed the defined sending threshold, the CH will try to access the media again. This can lead to a situation that, when the CH is trying to access the media, the Sink is serving the remaining simulated nodes from the previous data sending, and will only decide if the CH can access the media after serving all nodes from a previous sending.

To overcome the constraints, a higher gathering period should be used. As it is shown in the results for a gathering period of 180 seconds, the mean delay decreases substantially, the number of media access tries decreases as well and the number of packets in queue does not tend to grow constantly.

With this analysis it can be stated that the gathering period can not be arbitrary, but a well defined value taking into account the scenario characteristics.

5.3.3 Variation of Total Nodes

After evaluating the influence of the data gathering period in the performance of the solution, this section analyzes the impact that the total number of nodes have in the designed solution. For that purpose, four scenarios were tested, with 5, 10, 25 and 50 total number of nodes. For each scenario an appropriate data acquisition period was chosen. In Table 5.10 is presented the scenarios information.

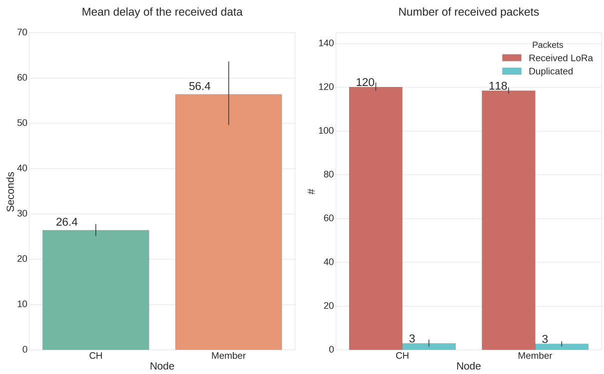
Table 5.10: Total nodes variation scenarios configuration

<table>
<tr><th></th><th>Long Range Mode</th><th>Short Range Mode</th><th>Gathering Period (s)</th><th>Nodes per Cluster</th><th>Number of Clusters</th><th>Total Nodes</th><th>Member Node Send Threshold</th><th>CH Send Threshold</th><th>Test Duration (min)</th></tr>
<tr><td>5 Nodes Scenario</td><td rowspan="4">3</td><td rowspan="4">10</td><td>60</td><td rowspan="4">5</td><td>1</td><td>5</td><td rowspan="4">2</td><td rowspan="4">10</td><td>60</td></tr>
<tr><td>10 Nodes Scenario</td><td>60</td><td>2</td><td>10</td><td>60</td></tr>
<tr><td>25 Nodes Scenario</td><td>180</td><td>5</td><td>25</td><td>90</td></tr>
<tr><td>50 Nodes Scenario</td><td>300</td><td>10</td><td>50</td><td>120</td></tr>
</table>

The obtained results for each scenario are presented in Figure 5.15 to 5.18.

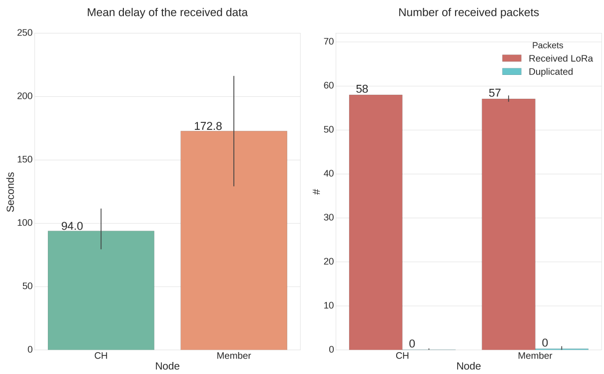
The results obtained with these experiments (Figure 5.15 to 5.18) show that the mean delay of the received packets in the server increases with the number of nodes, as it was expected. This happens because the number of clusters increases, which causes the CH to need a higher mean number of tries to gain the media access and send its data to the Sink (Figure 5.17). With the increasing of the tries to access the media, comes also the increasing of the time that the nodes spend in backoff.

The queue size variation (Figure 5.18) shows that in none of the scenarios occurs an increasing tendency of the queue size. This was prevented by choosing an appropriate data gathering period taking into account the total number of nodes. The total transmissions of data packets tend to decrease with the increasing of the number of nodes. This happens because of the gathering period adjustment, which leads to a lower number of data packets generated by the nodes (Figure 5.17 and 5.15).

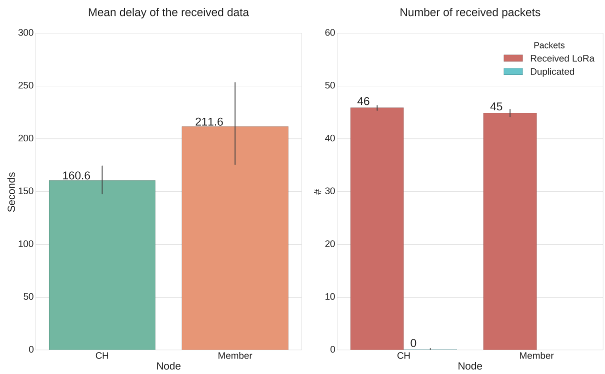


(a) Mean delay and packets received with 5 nodes

(b) Mean delay and packets received with 10 nodes



(c) Mean delay and packets received with 25 nodes



(d) Mean delay and packets received with 50 nodes

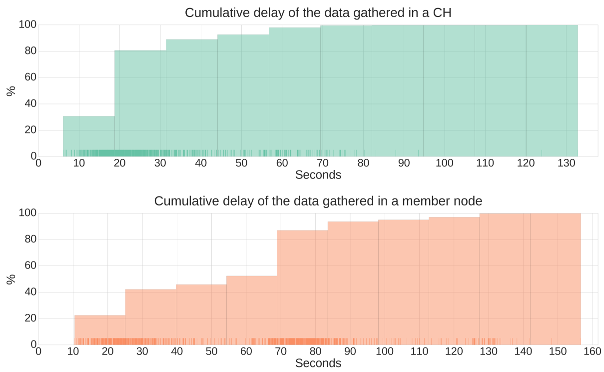
Figure 5.15: Mean delay and packets received with the number of nodes variation

---

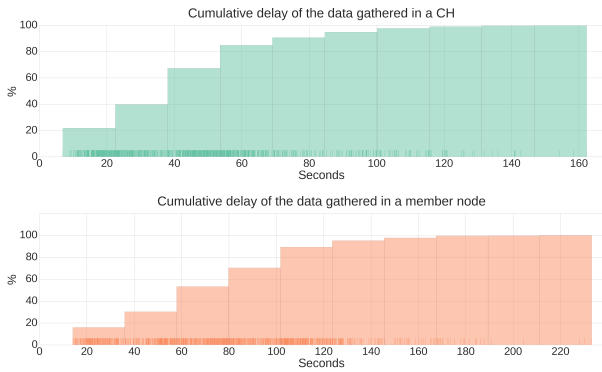
One result that is noticeable is the difference on the delay measured on the server for the data that are gathered by the CH and the data that are gathered by the member node. This can be explained by the fact that the sending thresholds are fixed. When for some reason the CH or the member node can not deliver all packets that were supposed to, these packets will only be sent when the threshold is reached again. That only occurs in the next data gathering action, which will cause that the delay of these packets is at least equal to a gathering period.

That is shown, for example, in Figure 5.16a, where it is visible that there are two regions where the delay of the packets is concentrated, around 20 and 40 seconds, and, 70 and 90 seconds: the packets that arrived within the last mentioned interval of delay, the ones that were sent in the next gathering action. This is a pattern that happens in all the results from here on, but do not have a relevant effective effect.

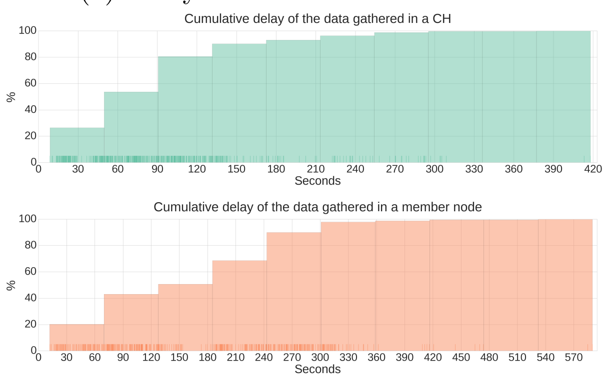
Analyzing the overall performance of the solution for the different scenarios, it can be said that, if the increase of the nodes is accompanied with a right adjustment of the data gathering period, the solution can handle and give a good response to different scenarios that it may be exposed to.



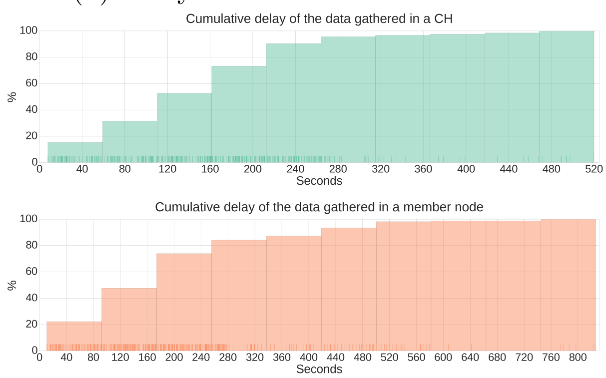
(a) Delay distribution with 5 nodes



(b) Delay distribution with 10 nodes

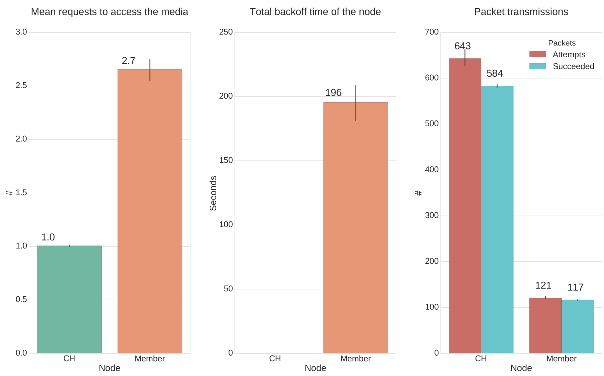


(c) Delay distribution with 25 nodes

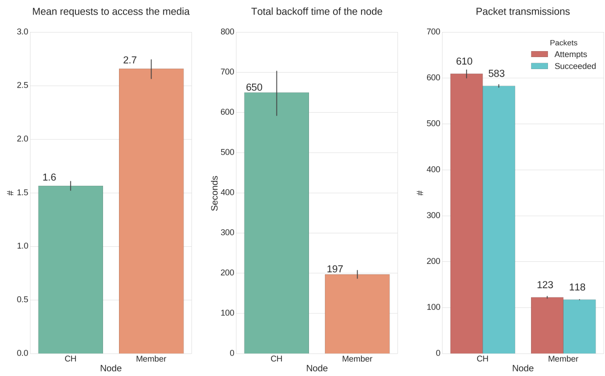


(d) Delay distribution with 50 nodes

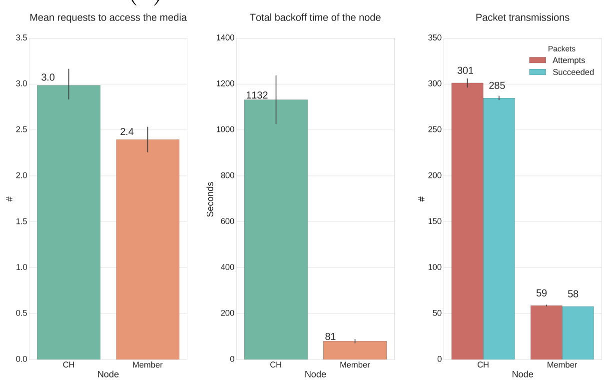
Figure 5.16: Delay distribution with the number of nodes variation



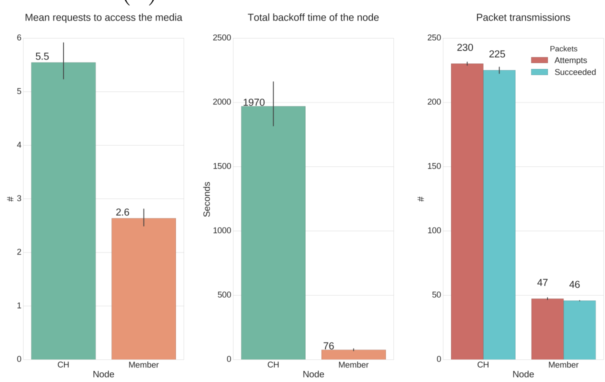
(a) MAC data with 5 nodes



(b) MAC data with 10 nodes



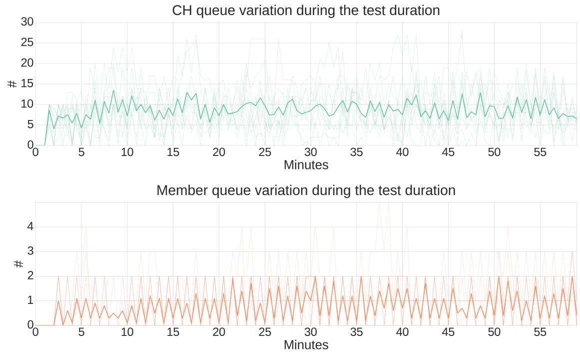
(c) MAC data with 25 nodes



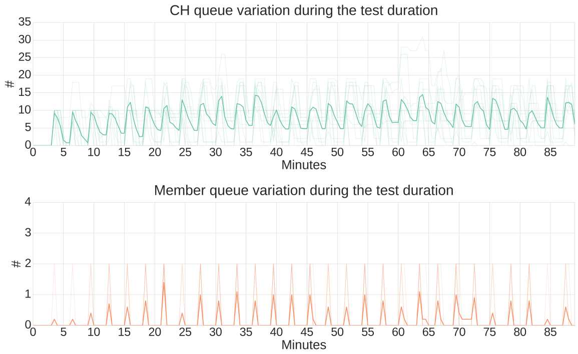
(d) MAC data with 50 nodes

Figure 5.17: MAC data with the number of nodes variation

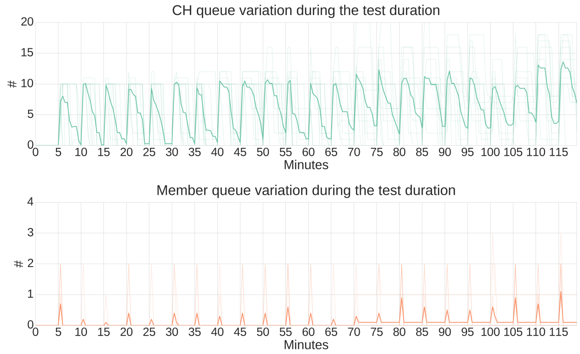
(a) Queue variation with 5 nodes



(b) Queue variation with 10 nodes



(c) Queue variation with 25 nodes



(d) Queue variation with 50 nodes

Figure 5.18: Queue variation with the number of nodes variation

5.3.4 Clustering Influence

In the scenarios presented next, it is tested how the proposed solution performs without the nodes being organized in clusters and how advantageous the cluster organization can be to the overall data gathering process. A total of 50 nodes were used to perform this test, then there were defined three scenarios, one without the nodes organized in clusters and two with different clusters organization. The information relative to the scenarios organization is presented in Table 5.11.

Table 5.11: Clustering scenarios configuration

<table>
<tr><th></th><th>Long Range Mode</th><th>Short Range Mode</th><th>Gathering Period (s)</th><th>Nodes per Cluster</th><th>Number of Clusters</th><th>Total Nodes</th><th>Member Node Send Threshold</th><th>CH Send Threshold</th><th>Test Duration (min)</th></tr>
<tr><th>No Clustering Scenario</th><td rowspan="3">3</td><td>-</td><td rowspan="3">300</td><td>-</td><td>-</td><td rowspan="3">50</td><td rowspan="3">2</td><td>-</td><td rowspan="3">120</td></tr>
<tr><th>5x10 Clustering Scenario</th><td rowspan="2">10</td><td>10</td><td>5</td><td>20</td></tr>
<tr><th>10x5 Clustering Scenario</th><td>5</td><td>10</td><td>10</td></tr>
</table>

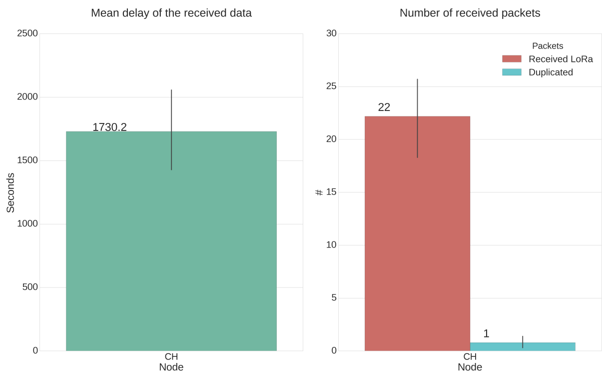
The results for the three scenarios are presented in Figure 5.19 to 5.22.

From the analysis of the obtained results, it is clearly visible that the scenario with no cluster organization presents the worst results when compared to the two scenarios where the nodes were organized in clusters. First the mean arriving delay of the packets at the server (Figure 5.20) is approximately 10 times higher than the delay achieved in the organized scenarios. That can be explained by the evolution of the queue size in the scenario without clustering (Figure 5.22a), which grows constantly as the test time goes on.

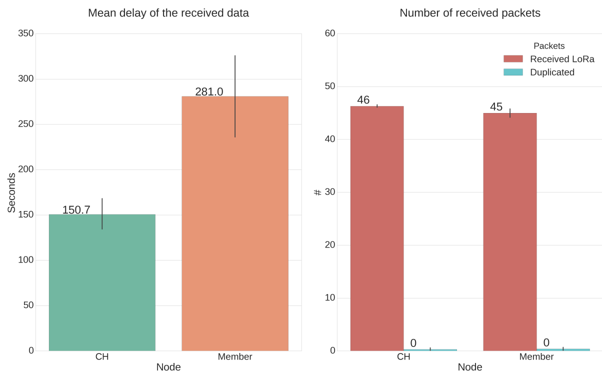
Also, the mean number of media access tries are larger in the flat organization than in the cluster organized scenarios. In fact, it takes nearly the same mean number of messages that are sent for one node in the unorganized scenario, that it takes in a cluster (including the access to the media of the CH to send data to the Sink) in the 10x5 scenario.

When comparing the two organized scenarios, there is no major difference between them. The values of delay are quite similar, in both cases the queue size does not increase constantly, being the one difference in the media access. This was expected, since in one scenario there are five clusters with ten nodes each, and, in the other scenario there are ten clusters with five nodes each.

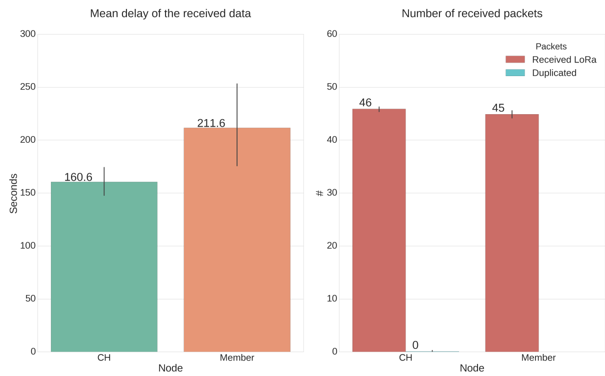
With this experiment, it is clarified that an organized node architecture brings advantages in some critical points, such as, message exchange minimization, packet delivery with lower delays, and uncontrolled queue size increase is resolved.



(a) Mean delay and packets received without clustering

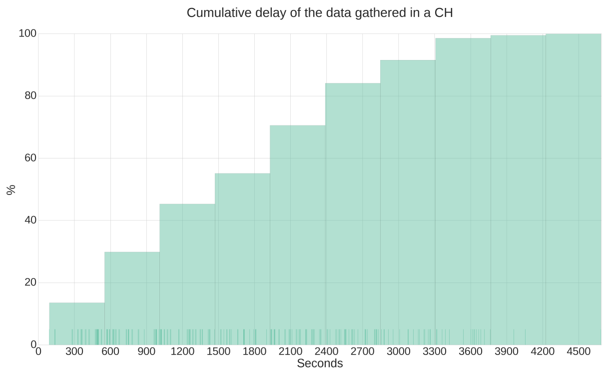


(b) Mean delay and packets received with 5x10 clustering

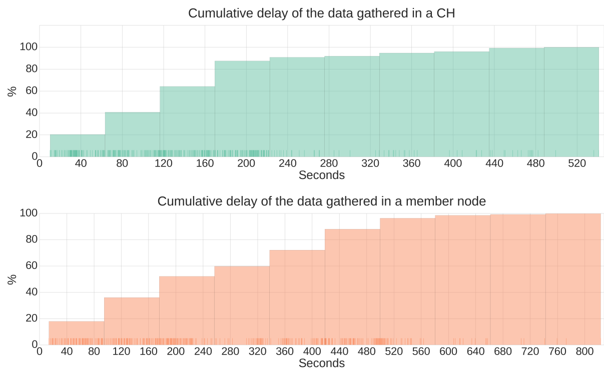


(c) Mean delay and packets received with 10x5 clustering

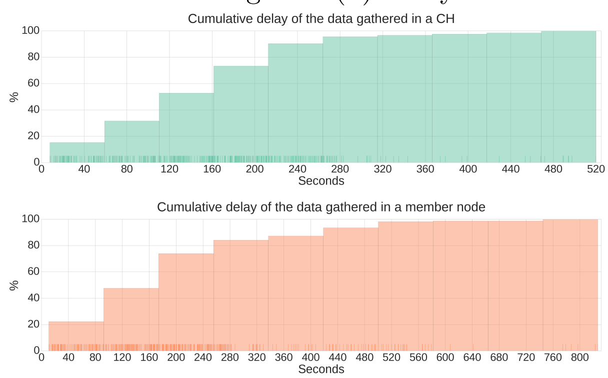
Figure 5.19: Mean delay and packets received with different cluster organizations



(a) Delay distribution without clustering

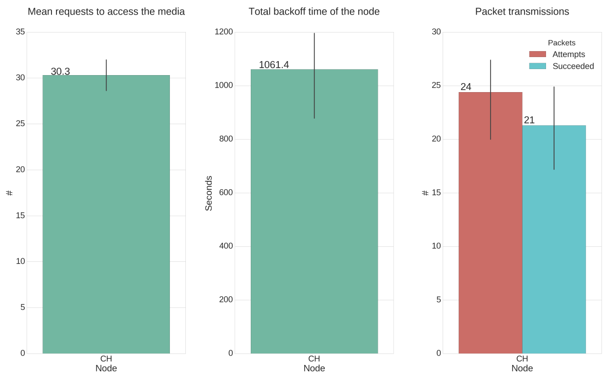


(b) Delay distribution with 5x10 Clustering

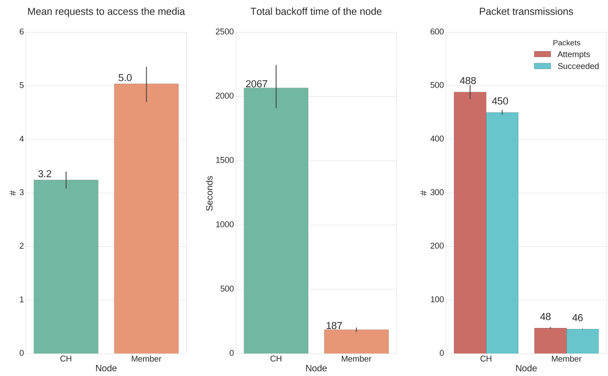


(c) Delay distribution with 10x5 Clustering

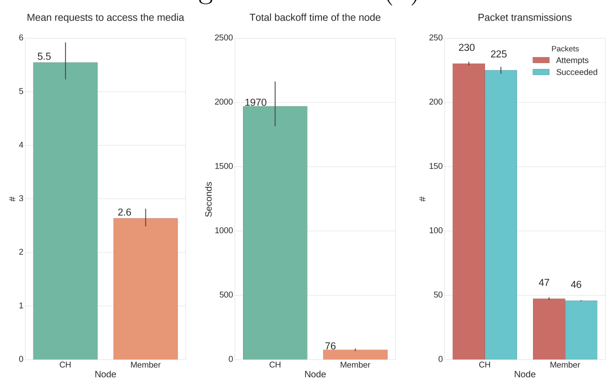
Figure 5.20: Delay distribution with different cluster organizations



(a) MAC data without clustering

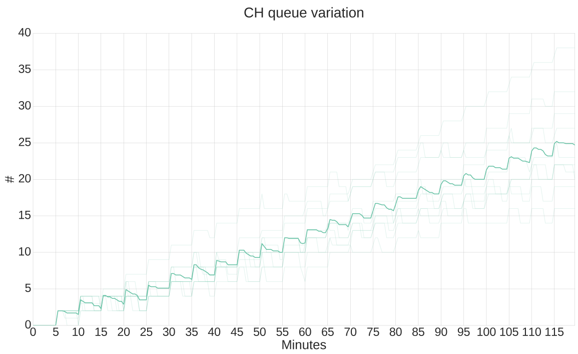


(b) MAC data with 5x10 Clustering

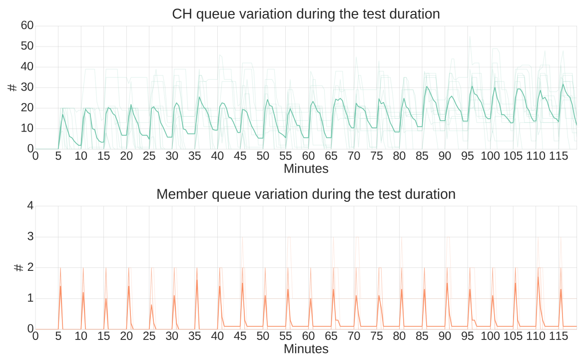


(c) MAC data with 10x5 Clustering

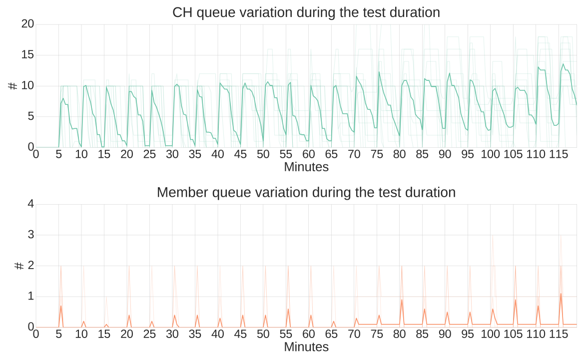
Figure 5.21: MAC data with different cluster organizations



(a) Queue variation without Clustering



(b) Queue variation with 5x10 Clustering



(c) Queue variation with 10x5 Clustering

Figure 5.22: Queue variation with different cluster organizations

5.3.5 LoRa along with WiFi

The impact of the multi-technology capability is another evaluation parameter that is taken into consideration. For that, two scenarios were defined where each one was further tested in two different conditions, one with only LoRa to send data and the other one with the presence of a WiFi network in range of the CH node at some instants.

The WiFi network is provided by an AP, that within a random period between 3 and 10 minutes, is broadcasting a valid SSID during a random time between 20 and 34 seconds.

The Scenario 1 is characterized by the fact that the chosen gathering period is not appropriate for the case when there is only LoRa available. As a consequence it is expected that the performance of the solution is not the desirable. Scenario 2 is the one used previously, more exactly, in the evaluation of the total nodes variation (Sub-Section 5.3.3). As it was mentioned, both scenarios were tested in two situations, single-technology (LoRa) and multi-technology (LoRa and WiFi). In Table 5.12 it is presented the scenarios relevant information.

Table 5.12: Multi-Technology scenarios configuration

	Long Range Mode	Short Range Mode	Gathering Period (s)	Nodes per Cluster	Number of Clusters	Total Nodes	Node Send Threshold	CH Send Threshold	Test Duration (min)
Scenario 1	1	10	120	5	5	25	2	10	60
Scenario 2	3		300	10		50		20	120

Figures 5.23 to 5.28 show the obtained results for the different situations in the two testing scenarios.

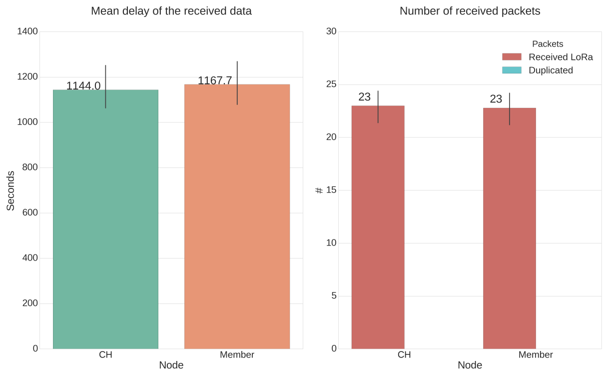
Starting with Scenario 1, with only LoRa available, the results show that the delay of the received packets in the server is considerably high, comparing with the values achieved in the already analyzed scenarios. This is explained by the intentional choosing of an inappropriate value of data gathering period, that being too short leads to a constant accumulation of packets in the sending queue, as it is shown in Figure 5.27a.

When a WiFi network presence is added to the Scenario 1, the impact in terms of the improvement of the results is significant, from the delay of the received packets, that decreased nearly 8 times, to the queue increase control. Due to the fact that a significant number of packets accumulate in the queue, the majority of the received packets in the server were transmitted through WiFi from the CH (Figure 5.23b and Figure 5.26a).

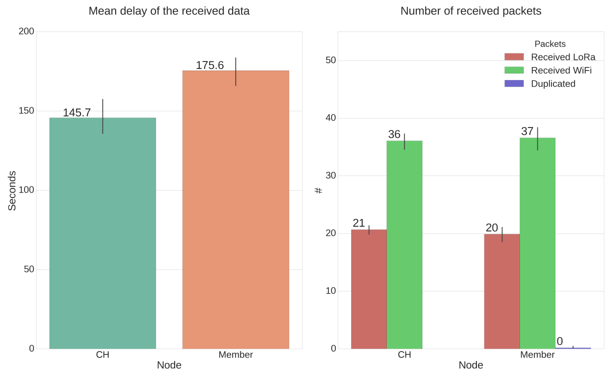
In Figure 5.26a it is also presented the information relative to the WiFi connection, that shows that for Scenario 1 a mean number of 8 contacts is made (approximately, 1 contact at every 7 minutes and 30 seconds) with a mean total duration of 24.4 seconds.

In Scenario 2, the impact of the WiFi network presence is not so noticeable since this scenario was defined considering the constraints that the solution has to be able to have a good performance. Nevertheless, the network presence contributes to a decreasing of the delay and the need for using LoRa so frequently.

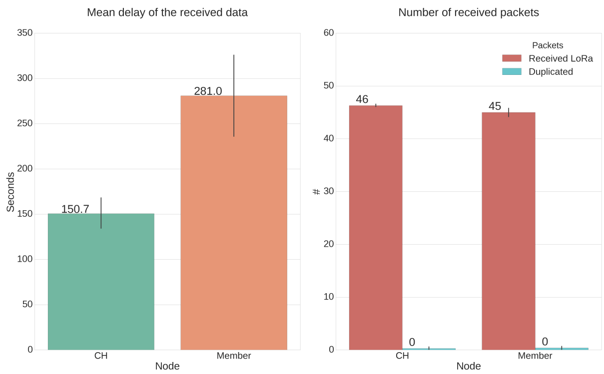
Figure 5.23 shows the impact of the WiFi presence in the packets delay distribution. This impact is more noticeable in Scenario 1 where 80% of the packets are received with a delay equal or minor than 250 seconds when a WiFi network is presented; when only LoRa



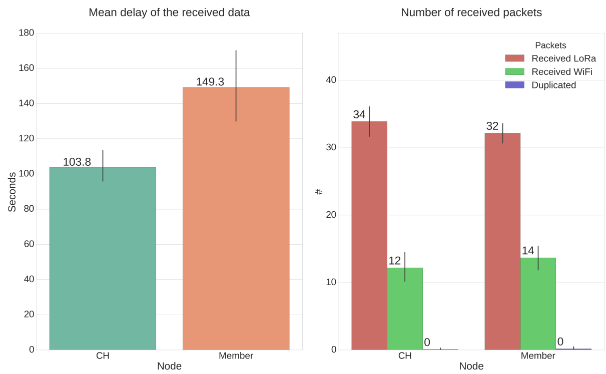
(a) Mean delay and packets received with only LoRa (Scenario 1)



(b) Mean delay and packets received with LoRa and WiFi (Scenario 1)



(c) Mean delay and packets received with LoRa (Scenario 2)



(d) Mean delay and packets received with LoRa and WiFi (Scenario 2)

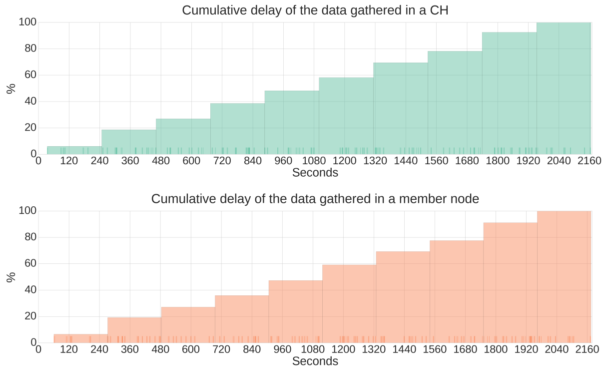
Figure 5.23: Mean delay and packets received with and without WiFi presence

is used, less than 20% of the packets are received with equal delay.

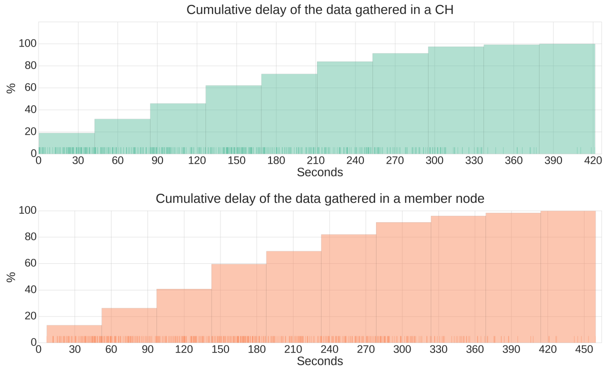
In terms of the media access attempts (Figure 5.25), it does not suffer any significant change since, when the nodes are sending data through LoRa they will need the same mean number of access attempts. In the case of the number of data packets sent through LoRa, a major difference is noticed in Scenario 2, since in Scenario 1 it is overloaded of data packets and the majority of them are sent through WiFi. Once that Scenario 2 has appropriate configurations, the number of packets sent through LoRa decreases with the addition of a WiFi network.

As it was explained in Section 3.4, when a node is sending data through LoRa and a valid WiFi network becomes available to forward the node's data, the preferred technology is changed during the time that the WiFi network is available. In Figure 5.28 it is presented the technologies sending state during one of the tests done. As it can be seen, near the minute 30, the CH is sending data through LoRa when a WiFi network becomes available for it to forward data, and the change is performed.

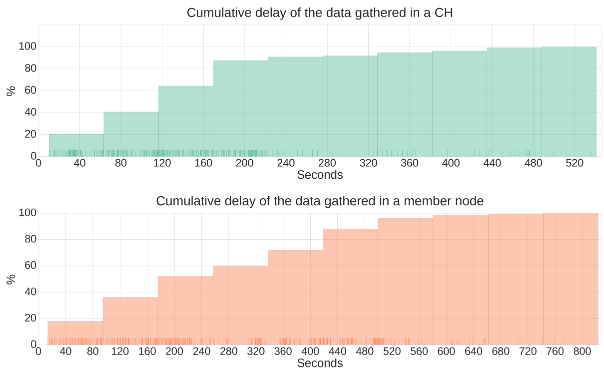
As it was mentioned during the explanation of the multi-technology approach in Section 3.4, there is no time wasted during the technology change, since it only occurs when the



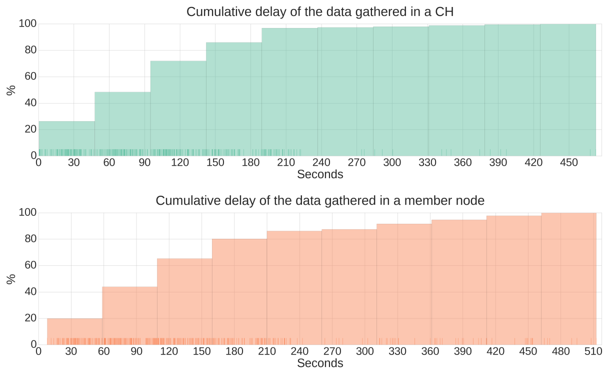
(a) Delay distribution with only LoRa (Scenario 1)



(b) Delay distribution with LoRa and WiFi (Scenario 1)



(c) Delay distribution with only LoRa (Scenario 2)

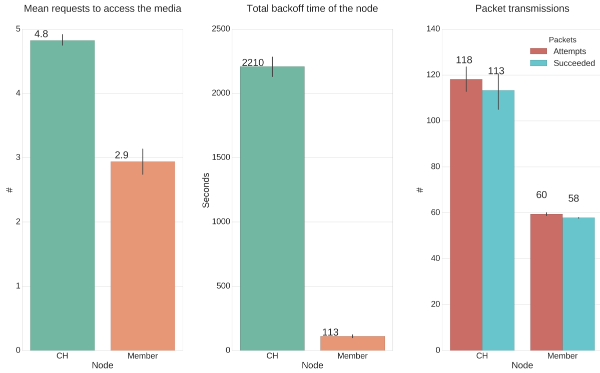


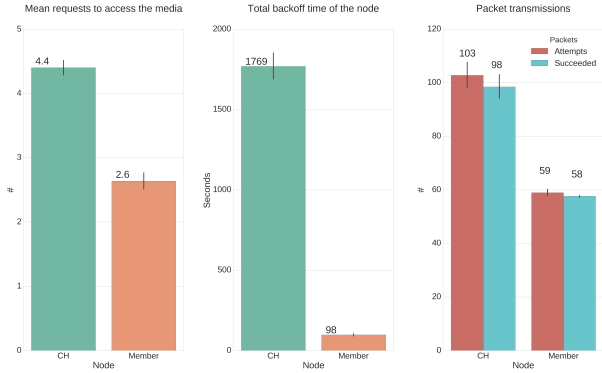
(d) Delay distribution with LoRa and WiFi (Scenario 2)

Figure 5.24: Delay distribution with and without WiFi presence

WiFi is completely ready. That is highlighted in the technology change action pointed in Figure 5.28.

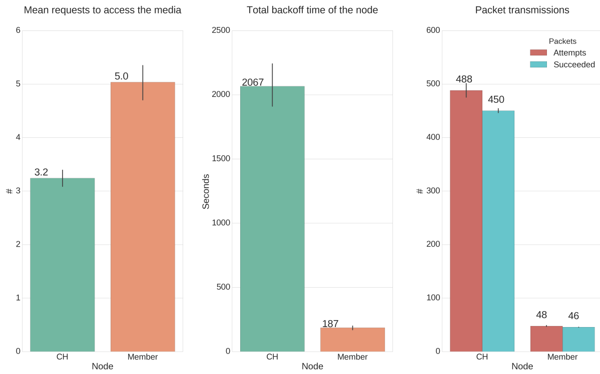
With this experiment it can be concluded that the WiFi presence, even if it is sporadic, is an improvement to the overall solution performance. The WiFi has a major contribution in scenarios that are not properly designed, that is, with a high frequency data gathering, which makes it difficult to be handled only by LoRa or with a numerous number of nodes.

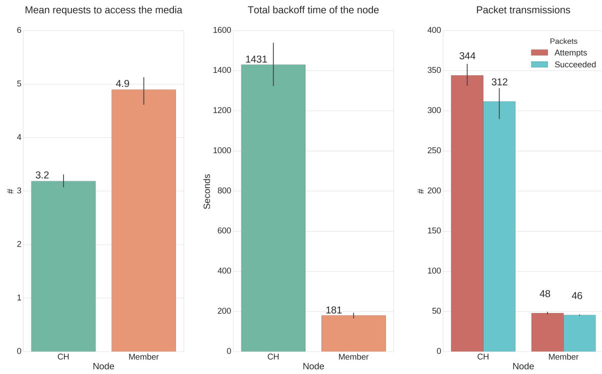




(a) MAC data with only LoRa (Scenario 1)

(b) MAC data with LoRa and WiFi (Scenario 1)

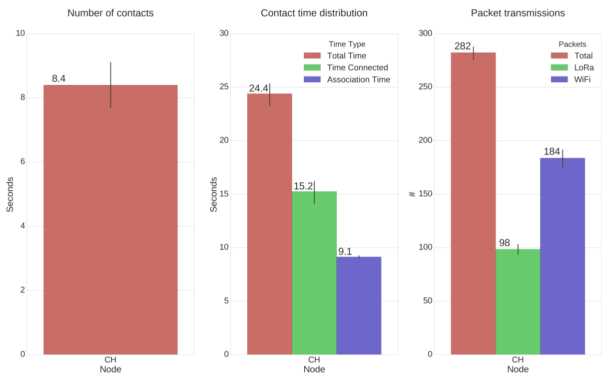


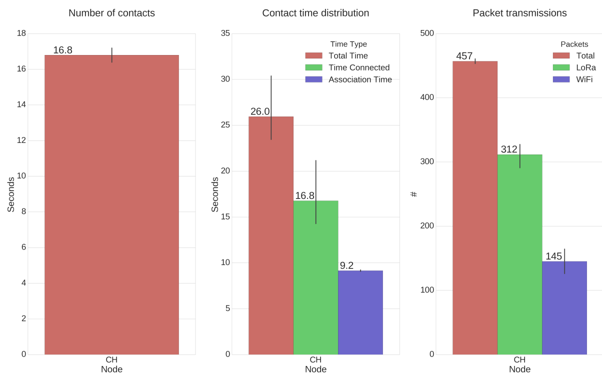


(c) MAC data with only LoRa (Scenario 2)

(d) MAC data with LoRa and WiFi (Scenario 2)

Figure 5.25: MAC data with and without WiFi presence

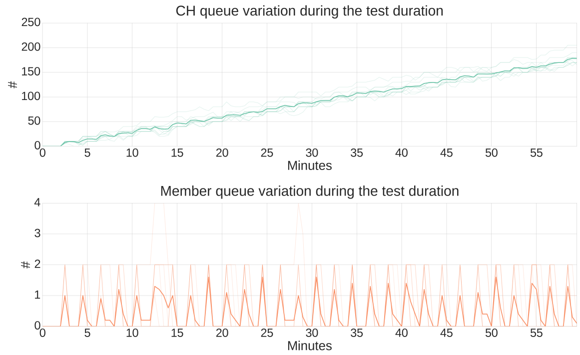




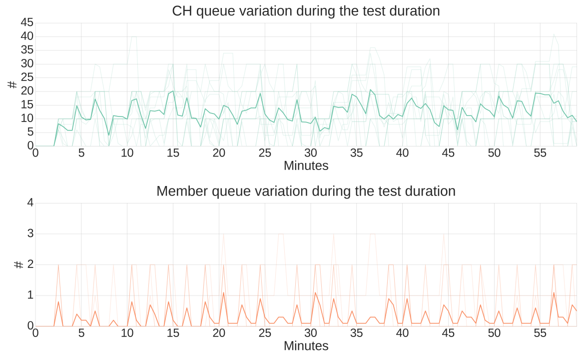
(a) WiFi data (Scenario 1)

(b) WiFi data (Scenario 2)

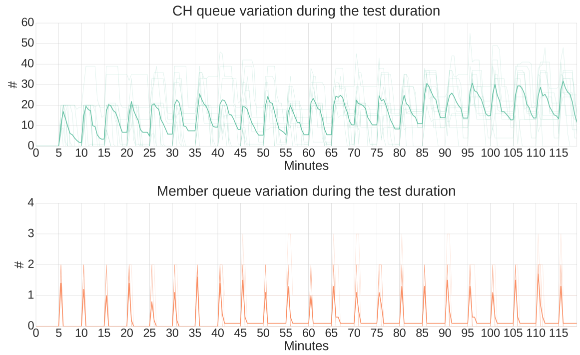
Figure 5.26: WiFi data



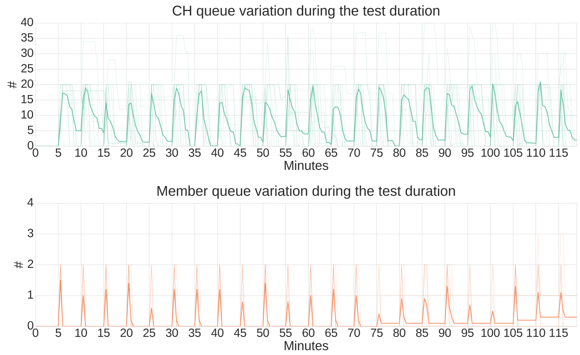
(a) Queue variation with only LoRa (Scenario 1)



(b) Queue variation with LoRa and WiFi (Scenario 1)



(c) Queue variation with only LoRa (Scenario 2)



(d) Queue variation with LoRa and WiFi (Scenario 2)

Figure 5.27: Queue variation with and without WiFi presence

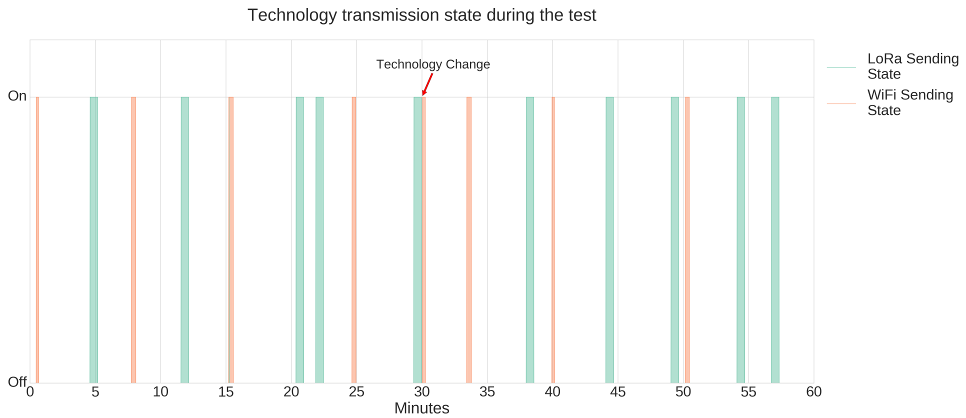


Figure 5.28: Technology state

5.3.6 DCU with image gathering

Another scenario where the multi-technology communication is presented is when, simultaneous with the sensors data gathering, there is also image gathering. As it was defined, when there is image gathering, the technologies are dedicated to a specific type of data. In this case the WiFi interface will only send image data and LoRa will send the sensors data, each can lead to situations when the transmission is simultaneous from the different technologies. The scenario used is the 25 nodes scenario previously used in the nodes variation experiment, with the addition of the image gathering with a periodicity of 90 seconds.

The WiFi AP is the same as the one explained in Sub-Section 5.3.5. The scenario configuration is presented in Table 5.13.

Table 5.13: Image gathering scenario configuration

	Long Range Mode	Short Range Mode	Gathering Period (s)	Image Period (s)	Nodes per Cluster	Number of Clusters	Total Nodes	Node Send Threshold	CH Send Threshold	Test Duration (min)
Image Gathering Scenario	3	10	180	90	5	5	25	2	10	90

In Figure 5.29 it is presented the obtained results for the image gathering scenario, and in Figure 5.30 it is shown the technologies transmissions in one of the tests.

Comparing the results presented from Figure 5.29a to Figure 5.29d, to the ones obtained in Section 5.3.3 for the same scenario, it is verified that the results are similar. This is expected, taking into account that in this scenario the WiFi network does not forward sensor data packets. Thus, the LoRa related results are not supposed to suffer significant modifications.

Relatively to the image gathering, a mean of 57 images were received with a mean delay of 212 seconds. The image data is sent in the 13 contacts that were made during the test time, with a mean contact time of 24.3 seconds.

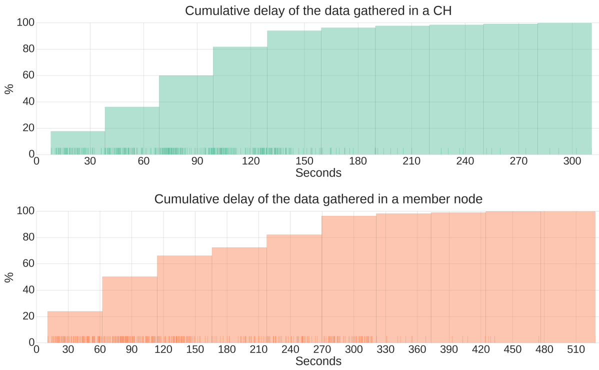
The key feature and main advantage of this traffic differentiation is the fact that the two technologies can send data at the same time without interfering with each other. That is evidenced by the two points annotated in Figure 5.30, where sensors data is sent through LoRa and image data is sent through WiFi at the same time.

5.3.7 Gathered Data Example

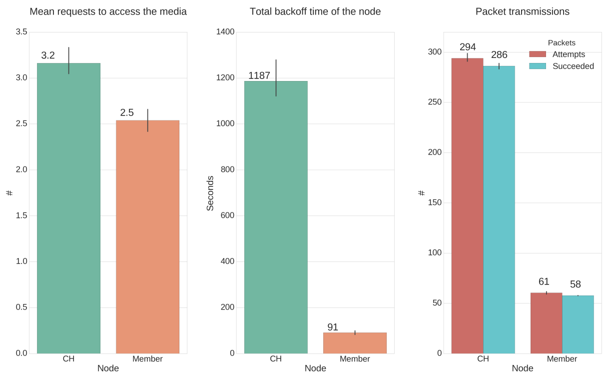
During the tests of the developed solution, real data was gathered by the sensors.

As an example, in Figure 5.31 it is presented the information of the temperature, humidity and luminosity, inside of a room during the day of 22 of September 2016.

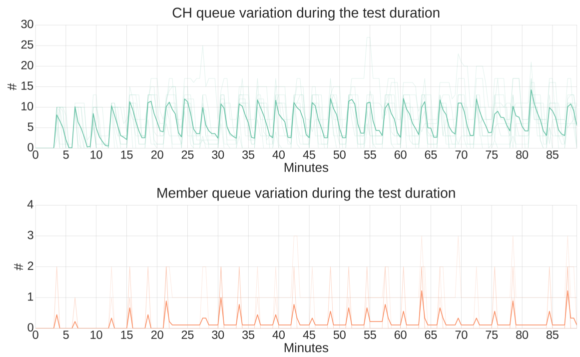
(a) Mean delay and packets received with image gathering



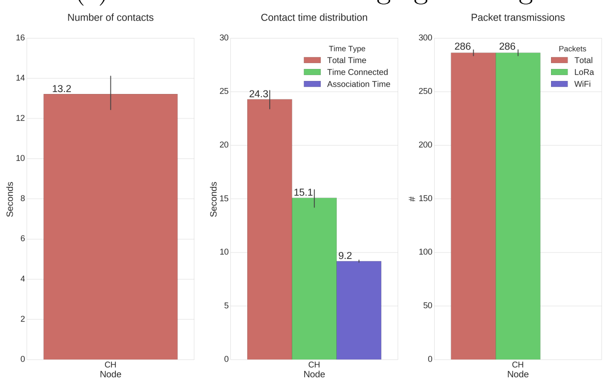
(b) Delay distribution with image gathering



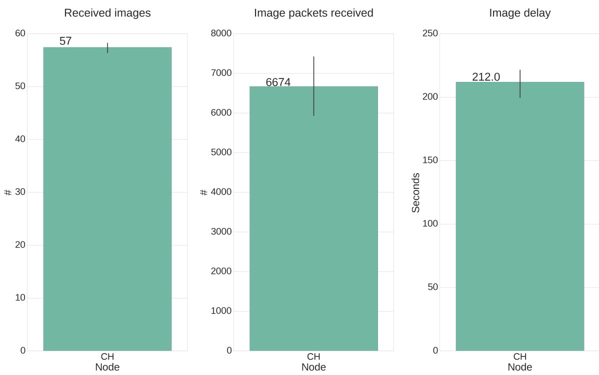
(c) MAC data with image gathering



(d) Queue distribution with image gathering



(e) WiFi data



(f) Image data

Figure 5.29: Results with image gathering

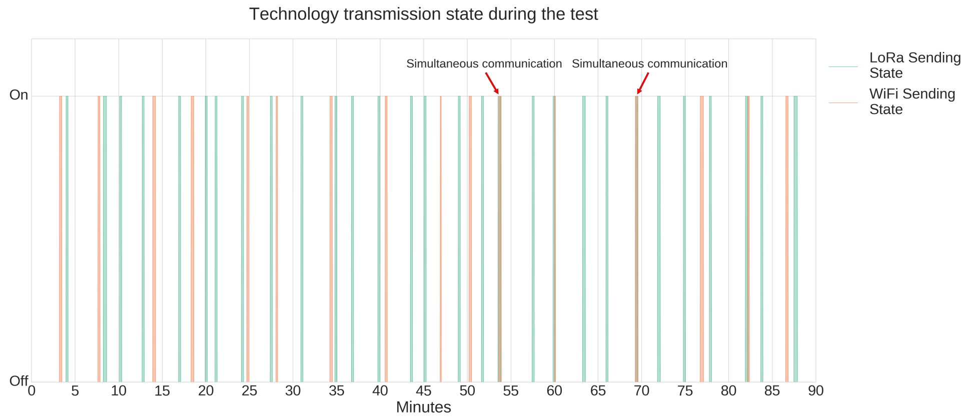


Figure 5.30: Technology state with image gathering

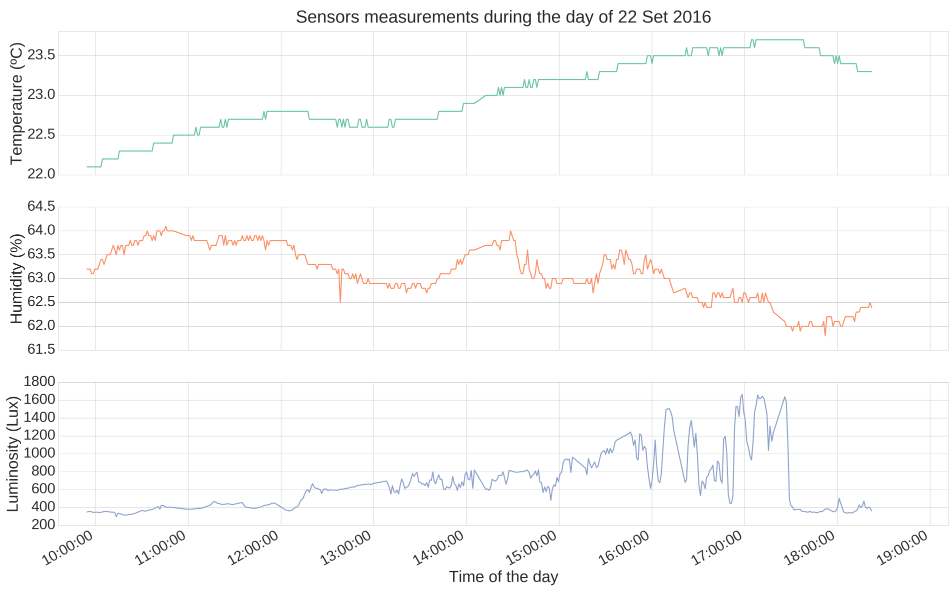


Figure 5.31: Sensors gathered data

5.4 Chapter Considerations

This chapter presented the evaluation of the LoRa technology in terms of its capabilities, such as range and signal quality. Also, the integration of the LoRa technology in a data gathering solution was evaluated.

First, the LoRa technology was tested in different scenarios in order to study how it behaves in different conditions. The first scenario was rural, where a maximum range above 5 km was achieved. Then, two urban scenarios which presented distinct conditions were tested, and a maximum range of nearly 2 km was achieved in both cases. Comparing the two urban scenarios lead to the conclusion that, the nature of the region in test influences the performance of the LoRa technology.

Then, the proposed data gathering solution was evaluated. With the diverse scenarios tested, it is possible to conclude that several parameters influence the performance of the solution, such as the nodes organization, total number of nodes, gathering period, traffic type and WiFi presence.

It can be stated that, with the appropriate network configurations (gathering period adapted to the number of nodes), the proposed solution has a good performance and can deliver the information with a delay that does not affect the type of data being handled. When the parameters are not well defined for the network, it presents larger delays and a worst performance, which can be surpassed if a WiFi network is presented from time to time.

With the analyses of the capabilities of LoRa and its integration in a data gathering solution, it can be said that LoRa is suitable to act as the main technology in some specific conditions, or as a complement to other technologies, in a data gathering network.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

As it was presented in the beginning of this dissertation, LPWAN technologies are the rising stars in the IoT networks. One of the most evolved LPWAN technologies is LoRa, which presents different trade-offs between coverage and data rate, making it a suitable and versatile technology for a data gathering implementation.

The proposed data gathering solution presented in this work aims to extend the technologies in the existing data gathering platform that has only WiFi as a communication channel to deliver the acquired data by sensors. The inclusion of LoRa in the platform aims to provide an alternative communication technology for the sensors to deliver their data. In order for this addition to the network to be feasible and reliable, some key points needed to be taken into account and further developed.

In the developed modules for the data gathering solution are included:

- **Cluster Organization** A simple clustering algorithm was proposed to organize the nodes with LoRa radio communication. This was found a necessary issue due to the multiple concurrent transmissions that may occur in case that a flat organization was adopted. The usage of different LoRa communication modes for the intra-cluster and cluster-sink communication are a property of this proposed scheme.

- **MAC Protocol for LoRa** The proposed MAC protocol aims to minimize the concurrent transmission occurrences by the overhearing of RTS and CTS packets by the non-transmitting nodes.

- **Connection Manager** An entity responsible for managing the different technologies and to make decisions taking into account the data being handled was proposed. This Connection Manager is able to select the best technology between LoRa and WiFi when both are available, as well as to perform communications from both technologies at the same time without interfering with each other.

An evaluation of the LoRa radio capabilities was also performed, with interesting results in terms of the provided trade-offs between range, data rate and adaptation to different environments. LoRa presented a very good range for the different scenarios and a versatility to develop diverse applications in different environments.

With this work it is possible to conclude why LoRa is considered one of the most promising LPWAN technologies. It has proven that it can be used in many types of applications and increase its performance due to its unique characteristics. Moreover, the multi-technology approach is ideal to get the most out of all the technologies available and enable the heterogeneity of data and its requirements.

6.2 Future Work

Throughout the dissertation, it was possible to detect that there are still points that need to be improved or developed. Noteworthy:

Energy consumption awareness In order to make the solution available for battery-driven devices, it is important to take the power consumption in consideration and adapt the proposed protocols to be energy efficient;

Multi-hop inside cluster Adapt the cluster formation to include nodes that cannot reach the sink, but are in range of other nodes, to be included in a cluster either by direct contact to the CH or by multi-hop via member nodes;

Integration with network protocols Develop an upper layer framework to abstract the network complexity in order to use common Internet protocols such as CoAP, DNS, HTTP(s), over LoRa;

LoRa Evaluation Perform another range of tests to evaluate LoRa connectivity in scenarios not yet evaluated, such as in a water environment, with the help of aquatic drones or boats; and mobility scenarios to evaluate how the LoRa reception can handle moving nodes.

LoRa applications Analyse the possibility of implementing LoRa communications in other areas of interest, such as control of swarms of aquatic drones, smart bicycles and human well-being.

Bibliography

[1] Wetync. [Online]. Available: http://www.wetync.com/home-nl/

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] Ericsson. (2016, 04) White paper: Cellular networks for massive iot – enabling low power wide area applications. [Online]. Available: https://www.ericsson.com/news/160106-cellular-networks-massive-iot_244039856_c

[4] Gartner. (2013, 12) Gartner says the internet of things installed base will grow to 26 billion units by 2020. [Online]. Available: http://www.gartner.com/newsroom/id/2636073

[5] D. Lund, C. MacGillivray, V. Turner, and M. Morales, "Worldwide and regional internet of things (iot) 2014–2020 forecast: A virtuous circle of proven value and demand," 2014.

[6] H. Schmidbauer. (2016, 09) White paper: Nb-iot vs lora™ technology - which could take gold? LoRa Alliance. [Online]. Available: https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRa-Alliance-Whitepaper_NBIoT_vs_LoRa.pdf

[7] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, "The applications of wifi-based wireless sensor network in internet of things and smart grid," in *2011 6th IEEE Conference on Industrial Electronics and Applications*. IEEE, 2011, pp. 789–793.

[8] J.-P. Bardyn, T. Melly, O. Seller, and N. Sornin, "Iot: The era of lpwan is starting now," in *European Solid-State Circuits Conference, ESSCIRC Conference 2016: 42nd*. IEEE, 2016, pp. 25–30.

[9] R. Sanchez-Iborra and M.-D. Cano, "State of the art in lp-wan solutions for industrial iot services," *Sensors*, vol. 16, no. 5, p. 708, 2016.

[10] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: A survey," *arXiv preprint arXiv:1606.07360*, 2016.

[11] A.-A. A. Bouleogeorgos, P. D. Diamantoulakis, and G. K. Karagiannidis, "Low power wide area networks (lpwans) for internet of things (iot) applications: Research challenges and future trends," *arXiv preprint arXiv:1611.07449*, 2016.

[12] S. Barrachina, B. Bellalta, T. Adame, and A. Bel, "Multi-hop communication in the uplink for lpwans," *arXiv preprint arXiv:1611.08703*, 2016.

[13] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the iot: Lessons learned from industrial implementations," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on.* IEEE, 2015, pp. 181–186.

[14] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range iot technologies: The dawn of lora™," in *Future Access Enablers of Ubiquitous and Intelligent Infrastructures.* Springer, 2015, pp. 51–58.

[15] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios," *arXiv preprint arXiv:1510.00620*, 2015.

[16] Sigfox. [Online]. Available: https://www.sigfox.com

[17] Weightless. [Online]. Available: http://www.weightless.org

[18] Ingenu. [Online]. Available: http://www.ingenu.com

[19] T. J. Myers, D. T. Werner, K. C. Sinsuan, J. R. Wilson, S. L. Reuland, P. M. Singler, and M. J. Huovila, "Light monitoring system using a random phase multiple access system," Jul. 2 2013, uS Patent 8,477,830.

[20] IEEE-SA. (2013, 06) Ieee sa - 802.15.4k-2013 - ieee standard for local and metropolitan area networks— part 15.4: Low-rate wireless personal area networks (lr-wpans)—amendment 5: Physical layer specifications for low energy, critical infrastructure monitoring networks. [Online]. Available: https://standards.ieee.org/findstds/standard/802.15.4k-2013.html

[21] (2016) Rpma technology for the internet of things. Ingenu. [Online]. Available: http://theinternetofthings.report/Resources/Whitepapers/4cbc5e5e-6ef8-4455-b8cd-f6e3888624cb_RPMA%20Technology.pdf

[22] DASH7. [Online]. Available: http://www.dash7-alliance.org

[23] O. Cetinkaya and O. B. Akan, "A dash7-based power metering system," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC).* IEEE, 2015, pp. 406–411.

[24] W. Guibene, K. E. Nolan, and M. Y. Kelly, "Survey on clean slate cellular-iot standard proposals," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1596–1599.

[25] M. Elkhodr, S. Shahrestani, and H. Cheung, "Emerging wireless technologies in the internet of things: a comparative study," *arXiv preprint arXiv:1611.00861*, 2016.

[26] LoRa. [Online]. Available: https://www.lora-alliance.org

[27] A. Berni and W. Gregg, "On the utility of chirp modulation for digital signaling," *IEEE Transactions on Communications*, vol. 21, no. 6, pp. 748–751, 1973.

[28] F. Sforza, "Communications system," Mar. 26 2013, uS Patent 8,406,275. [Online]. Available: http://www.google.com/patents/US8406275

[29] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[30] Semtech, "Sx1272/73 datasheet, rev 3," Tech. Rep., 03 2015. [Online]. Available: http://www.semtech.com/images/datasheet/sx1272.pdf

[31] ——, "An1200.13 sx1272/3/6/7/8: Lora modem – designer's guide, revision 1," Tech. Rep. [Online]. Available: https://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf

[32] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan™ specifications," Tech. Rep. [Online]. Available: https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf

[33] L. Alliance. (2015) White paper: A technical overview of lora and lorawan. LoRa Alliance. [Online]. Available: https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf

[34] M. Aref and A. Sikora, "Free space range measurements with semtech lora™ technology," in *Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-SWS), 2014 2nd International Symposium on*. IEEE, 2014, pp. 19–23.

[35] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of lpwans: range evaluation and channel attenuation model for lora technology," in *ITS Telecommunications (ITST), 2015 14th International Conference on*. IEEE, 2015, pp. 55–59.

[36] C. A. Trasviña-Moreno, R. Blasco, R. Casas, and Á. Asensio, "A network performance analysis of lora modulation for lpwan sensor devices," in *Ubiquitous Computing and Ambient Intelligence: 10th International Conference, UCAmI 2016, San Bartolomé de Tirajana, Gran Canaria, Spain, November 29–December 2, 2016, Part II.* Springer, 2016, pp. 174–181.

[37] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, "Evaluation of lora lpwan technology for remote health and wellbeing monitoring," in *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2016, pp. 1–5.

[38] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis *et al.*, "Smartsantander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.

[39] T. Ojala, "Open urban testbed for ubiquitous computing," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 442–447.

[40] R. N. Murty, G. Mainland, I. Rose, A. R. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "Citysense: An urban-scale wireless sensor network and testbed," in *Technologies for Homeland Security, 2008 IEEE Conference on.* IEEE, 2008, pp. 583–588.

[41] Y. Luis, P. M. Santos, T. Lourenco, C. Pérez-Penichet, T. Calcada, and A. Aguiar, "Urbansense: An urban-scale sensing platform for the internet of things," in *Smart Cities Conference (ISC2), 2016 IEEE International.* IEEE, 2016, pp. 1–6.

[42] MIT. Clairity: Mit's air quality network. [Online]. Available: http://clairity.mit.edu/site/html/

[43] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on.* IEEE, 2014, pp. 69–77.

[44] W. Zamora, C. T. Calafate, J.-C. Cano, P. Manzoni *et al.*, "Ecosensor: Monitoring environmental pollution using mobile sensors," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A.* IEEE, 2016, pp. 1–6.

[45] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for iot smart city experiments," in *Smart Cities Conference (ISC2), 2016 IEEE International.* IEEE, 2016, pp. 1–8.

[46] T. Petrić, M. Goessens, L. Nuaymi, A. Pelov, and L. Toutain, "Measurements, performance and analysis of lora fabian, a real-world implementation of lpwan," 2016.

[47] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* ACM, 2002, pp. 88–97.

[48] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "Wireless sensor networks for environmental monitoring: the sensorscope experience," in *2008 IEEE International Zurich Seminar on Communications.* IEEE, 2008, pp. 98–101.

[49] E. S. Nadimi, R. N. Jørgensen, V. Blanes-Vidal, and S. Christensen, "Monitoring and classifying animal behavior using zigbee-based mobile ad hoc wireless sensor networks and artificial neural networks," *Computers and Electronics in Agriculture*, vol. 82, pp. 44–54, 2012.

[50] V. Toldov, M. Jp, R. Igual, R. Wolhuter, N. Mitton, and L. Clavier, "Performance evaluation of lora radio solution for prednet wildlife animal tracking project," in *LPWAN*, 2016.

[51] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "Mac essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[52] I. Demirkol, C. Ersoy, F. Alagoz *et al.*, "Mac protocols for wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.

[53] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of mac protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[54] M. Bor, J. Vidler, and U. Roedig, "Lora for the internet of things," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '16, 2016, pp. 361–366.

[55] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.

[56] D. Dechene, A. El Jardali, M. Luccini, and A. Sauer, "A survey of clustering algorithms for wireless sensor networks."

[57] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on.* IEEE, 2000, pp. 10–pp.

[58] H. Dhawan and S. Waraich, "A comparative study on leach routing protocol and its variants in wireless sensor networks: a survey," *International Journal of Computer Applications*, vol. 95, no. 8, 2014.

[59] K. Kredo and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer networks*, vol. 51, no. 4, pp. 961–994, 2007.

[60] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.

[61] P. Karn, "Maca-a new channel access method for packet radio," in *ARRL/CRRL Amateur radio 9th computer networking conference*, vol. 140, 1990, pp. 134–140.

[62] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: a media access protocol for wireless lan's," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 212–225, 1994.

[63] I. S. Association *et al.*, "802.11-2012-ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *Retrived from http://standards. ieee. org/about/get/802/802.11. html*, 2012.

[64] Extending python with c or c++ — python 3.5.2 documentation. [Online]. Available: https://docs.python.org/3.5/extending/extending.html

[65] C. Pham, "Qos for long-range wireless sensors under duty-cycle regulations with shared activity time usage," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 4, p. 33, 2016.