Study and improvement of long range communication technologies for wireless sensor networks

Samira Abboud

Université Clermont Auvergne

Doctoral school n°70 : Engineering Sciences (EDSPI)

THESIS

presented by

Samira ABBOUD

to obtain the rank of

UNIVERSITY DOCTOR

Specialty: Computer Science

STUDY AND IMPROVEMENT OF LONG RANGE COMMUNICATIONS TECHNOLOGIES FOR WIRELESS SENSOR NETWORKS

publicly defended on 12 October 2020 with the following jury:

Prof. Andrzej DUDA	Professor, Grenoble INP	Reviewer
Dr. Nancy EL RACHKIDY	Assistant professor, Université Clermont Auvergne	Supervisor
Prof. Alexandre GUITTON	Professor, Université Clermont Auvergne	Co-director
Dr. Oana IOVA	Assistant professor, INSA Lyon	Examiner
Prof. Nicolas MONTAVONT	Professor, IMT Atlantique	Reviewer
Prof. Haidar SAFA	Professor, American University of Beirut	Co-director

ABSTRACT

The progress in low-energy, low-cost communication technologies have revolutionized remote sensing and monitoring applications. Internet of Things (IoT) has promised an ecosystem of connected devices across a wide range of applications such as in smart cities. Currently, many competing standards and technologies are attempting to seize the IoT, particularly in the area of remote sensing and communication technologies. LoRa (Long Range) is one of those technologies that is gaining popularity and attraction in the Wireless Sensor Networks (WSN) applications. The ability to make long-distance communications with relatively simple nodes, minimal infrastructure, reduced power requirements, and the use of unlicensed ISM bands provides a significant competitive advantage. Although the communication range in LoRa can exceed 15 kilometers in line of sight, the maximum bit rate that can be achieved is limited to few kilobits per second. Additionally, when a collision occurs in LoRa, the throughput is further reduced due to frame losses and retransmissions. The work of this thesis deals with the problem of collisions in LoRa that may occur under heavy load, and which degrade the performance of the network.

First, we consider the context for LoRaWAN uplink communications. We study the context of fully synchronized colliding LoRa signals, where each end-device has to retransmit its entire colliding frame after a collision occurs in LoRa. This behaviour decreases the overall throughput, and increases the energy consumption of the end-devices, and the delay of the frames. Therefore, in order to mitigate the damaging effects of collisions, we proposed a decoding algorithm to resolve synchronized colliding LoRa signals, in a saturated and confirmed network traffic. We substituted the conventional retransmission model of LoRa by having end-devices transmitting bitmaps instead of retransmitting whole frames to determine the correct symbols of each colliding frame. Our algorithm was able to significantly improve the overall throughput of the LoRaWAN MAC layer based on LoRa, and to decrease the energy consumption of the transmitters and the delay of the frames.

Second, we consider the context for LoRaWAN downlink communications. We noticed that the downlink in LoRa is a bottleneck. Hence, we worked on the gateway selection by the network server and its impact on the throughput, the energy consumption and the delay. We studied three types of gateway deployment and we show that the system

performance depends on this deployment. We showed that balancing the number of end-devices per gateway (also known as load) improves the throughput compared to choosing the gateway with the highest signal quality. Moreover, we showed that combining load and signal quality does not further improve the throughput. In addition, we showed that choosing the gateway with the highest signal quality decreases the delay and energy consumption compared to choosing the gateway with the lowest load.

Keywords: Internet of Things (IoT), Low Power Wide Area Networks (LPWAN), LoRa, LoRaWAN, Gateway Selection, Slot, Collision Cancellation, Synchronized Signals.

RÉSUMÉ

Les progrès des technologies de communication à faible consommation d'énergie et à faible coût ont révolutionné les applications de télédétection et de surveillance. L'Internet des objets (IoT) a promis la création d'un écosystème d'appareils connectés à travers un large éventail d'applications, telles que les villes intelligentes. À l'heure actuelle, de nombreuses normes et technologies concurrentes tentent de saisir l'IoT, en particulier dans le domaine des technologies de télédétection et de communication. LoRa (Long Range) est l'une de ces technologies qui gagne en popularité et en attraction dans les réseaux de capteurs sans fil (WSN). La possibilité d'établir des communications longue distance avec des nœuds relativement simples, une infrastructure minimale, des besoins en énergie réduits et l'utilisation de bandes ISM sans licence offre un avantage concurrentiel significatif. Bien que la portée de communication dans LoRa puisse dépasser 15 kilomètres en visibilité directe, le débit binaire maximal pouvant être atteint est limité à quelques kilobits par seconde. De plus, lorsqu'une collision se produit dans LoRa, le débit est encore réduit en raison de pertes de trames et de retransmissions. Les travaux de cette thèse traitent le problème des collisions dans LoRa qui peuvent survenir sous une charge importante et qui dégradent les performances du réseau.

Premièrement, nous considérons le contexte des communications en liaison montante dans LoRaWAN. Nous étudions le contexte des signaux LoRa en collision synchronisée, où chaque appareil terminal doit retransmettre toute sa trame en collision après qu'une collision se produit dans LoRa. Ce comportement diminue le débit global et augmente la consommation d'énergie des terminaux et le délai des trames. Pour cette raison, afin d'atténuer les effets néfastes des collisions, nous avons proposé un algorithme de décodage pour résoudre les signaux LoRa en collision synchronisée, dans un trafic réseau saturé et confirmé. Nous avons remplacé le modèle de retransmission conventionnel de LoRa en un modèle faisant en sorte que les dispositifs terminaux transmettent des bitmaps au lieu de retransmettre des trames entières pour déterminer les symboles corrects de chaque trame en collision. Notre algorithme a pu améliorer significativement le débit global de la couche LoRaWAN MAC à base de LoRa, et diminuer la consommation d'énergie des émetteurs et le délai des trames.

Deuxièmement, nous considérons le contexte des communications en liaison descen-

dante dans LoRaWAN. Nous avons remarqué que la liaison descendante dans LoRa est un goulot d'étranglement. Nous avons donc travaillé sur la sélection de la passerelle par le serveur de réseau et son impact sur le débit, la consommation d'énergie et le délai. Nous avons étudié trois types de déploiement de passerelle et nous avons montré que les performances du système dépendent de ce déploiement. Nous avons montré que l'équilibrage du nombre de terminaux par passerelle (également connu sous le nom de charge) améliore le débit par rapport au choix de la passerelle avec la meilleure qualité de signal. En outre, nous avons montré que la combinaison de la charge et de la qualité du signal n'améliore pas davantage le débit. De plus, nous avons montré que le choix de la passerelle avec la meilleure qualité de signal diminue le délai des trames et la consommation d'énergie des terminaux par rapport au choix de la passerelle avec la charge la plus faible.

Mots clés : IoT, LPWAN, LoRa, LoRaWAN, Sélection de Passerelle , Slot, Annulation de Collision, Signaux Synchronisés.

Table of Contents

List of Figures

List of Tables

Part I

INTRODUCTION

Chapter 1

Introduction

Contents

1.1 From the Internet to the Internet of Things

The Internet is a network of billions of interconnected computers, mainly connected through high-speed cables. These computers host a large variety of servers (web servers, storage servers, email servers, etc.) and clients. The Internet is linked by a wide range of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing. Hence, connecting things to the Internet yields many benefits. The world has seen these benefits with our cellphones, laptops, and tablets, but this is true for everything else too. This proliferation of things becomes a new trend, and continues to grow in organizations and enterprises.

While estimates on the numbers of smart and connected devices can vary, it is recognised that each year we can expect tens of millions of new connected devices accessing the Internet on a daily basis. The Internet has influenced almost all spheres of daily life, and then it was transformed by embedded systems and wireless networks and became the Internet of Things. The Internet of Things means taking everything from the world and connecting it to the Internet.

The Internet of Things, or IoT, refers to the billions of physical devices and objects around the world that are now connected to the Internet, all collecting and sharing data. The IoT is a giant network of connected "things" (which also includes people). The relationship is between people-people, people-things, and things-things. With the IoT, we try to connect all the objects between them (not only computers). This connection is not made mostly by wired networks, but mostly by wireless networks.

1.2 Wireless technologies and limitations

The Internet of Things (IoT) refers to a future where everyday physical objects (such as smart devices, smart objects, sensors, actuators, embedded computers, etc.) are connected to the Internet [7]. The IoT has recently gained significant importance and consideration in academia and industry due the offers and capabilities that IoT provides. With IoT, the world becomes a network of smart objects, communicating with each other with least inputs from human beings. Smart objects would be around us, knowing about our likes and

needs. IoT facilitates connectivity not only among computers, but between actual, daily things and even among people. The IoT can affect every aspect of life and business, for instance the introduction of the IoT into education monitoring, such as student healthcare, access control in classroom and improving teaching and learning [8, 9].

The term IoT carries a large meaning. Indeed, according to [10], *"the semantic origin of the IoT term is composed by two words and concepts: Internet and Thing, where Internet can be defined as the world-wide network of interconnected computer networks, based on a standard communication protocol, the Internet suite (TCP/IP), while Thing is an object not precisely identifiable. Therefore, semantically, Internet of Things means a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols."* According to [11], *"The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service."*

The IoT is gaining rapid popularity and it is being deployed to realize smart cities, smart healthcare, smart homes, surveillance systems, environmental and animal monitoring, smart agriculture, smart farming, or smart metering applications [12–17]. More devices are connected to the network every year, and IoT still has a long way to go and grow. The total number of connected devices was estimated at around 50 billion by 2020 [9, 18–20], and around 125 billion by 2030 [1, 2] as shown in Fig. 1.1.

Current wireless technologies used to support IoT applications can be divided into short-range and long-range technologies. The main features and limitations of these solutions are the network management costs, the scalability of the network, the energy efficiency of the peripheral nodes, and the coverage area. Indeed, the legacy wireless technologies can not address diverse requirements of IoT applications, such as long range connectivity for low power and low data rate devices [21].

In the following, we present three wireless communication technologies in wireless networks with their features and limitations.

1.2.1 WiFi

Wireless fidelity (WiFi) [22–24] is a technology designed for connecting electronic devices in a wireless local area network (WLAN). WiFi is based on the IEEE 802.11 family of

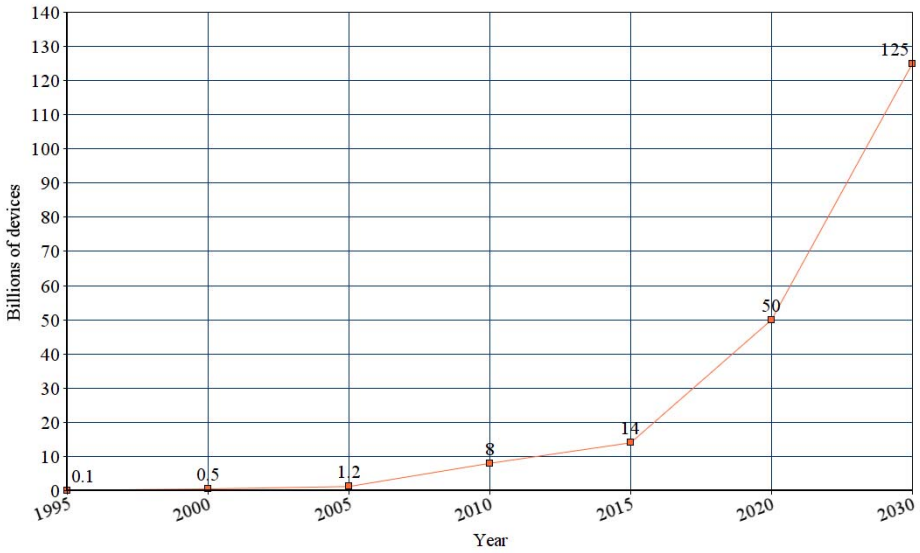


Figure 1.1: Estimation of IoT Growth by 2030 [1, 2].

standards which operate in the 2.4GHz and 5GHz unlicensed bands available worldwide. The WiFi includes IEEE 802.11a/b/g standards for WLAN [25].

WiFi has a massive bandwidth of 22 MHz, and, as a result, allows to achieve very fast data rates. The data rate is 54 Mb/s. It can even reach 800 Mb/s [26] with a bandwidth equal to 40 MHz. WiFi uses carrier sense multiple access with collision avoidance (CSMA/CA) channel access protocol, and, optionally, a request to send/clear to send (RTS/CTS) mechanism. Nowadays, there are more than 7 billion devices with WiFi technology in use [27–29].

In the IoT world, WiFi is used for many applications such as remote wireless monitoring and management of lights, power outlets, surveillance, alarms, appliances, climate control (like temperature and humidity control), metering, manufacturing control and diagnostics, medical equipment, etc. [26]. WiFi is a key technology in the development of IoT, and it provides a vast field for several IoT solutions.

Despite WiFi being the most widespread and generally known wireless communication

protocol, its broad usage across the IoT world is mainly limited by higher power consumption resulting from the need of retaining high signal strength, and fast data transfer for better connectivity and reliability. The principal drawback of WiFi in the smart home scenario is relatively higher power consumption [26].

1.2.2 Bluetooth

Bluetooth [30,31], also known as the IEEE 802.15.1 standard, is a short-range connectivity technology. It is considered to be a key solution for the future of the wearable electronics market such as wireless headphones or geolocation sensors, especially given its widespread integration with smartphones.

Bluetooth is based on a wireless radio system designed for short-range and inexpensive devices to replace cables for computer peripherals, such as mices, keyboards, printers, etc. [25]. This range of peripherals are used in a type of network known as wireless personal area network (WPAN). Designed with cost-effectiveness and reduced power consumption, the Bluetooth Low-Energy (BLE) protocol [32] requires very little power from the device. Yet, this comes with a compromise: when transferring frequently large amounts of data, BLE is not an effective solution since more power is consumed.

The data rate is 3 Mb/s. Bluetooth operates in the 2.4-GHz ISM (industrial, scientific, and medical) Radio Frequency band [26] which is available for license-free use in the whole world.

A set of Bluetooth devices sharing a common channel is called a piconet. A piconet is a star-shaped configuration in which the device at the center performs the role of master, and all other devices operate as slaves. Up to seven slaves can be active and served simultaneously by the master. If the master needs to communicate with more than seven devices, it can do this by asking the active slave devices to switch to low-power park mode, and then inviting other stationed slaves to become active in the piconet. This behaviour can be repeated, hence allowing a master to serve a large number of slaves [31].

Bluetooth specifies that devices must be able to achieve a minimum receiver sensitivity of -70 dBm. However, Bluetooth implementations typically achieve much higher receiver sensitivity levels of -95 dBm or better. For the same transmission power, the range of Bluetooth is shorter than it is for 802.11 WLAN [31].

1.2.3 IEEE 802.15.4/ZigBee

IEEE 802.15.4 [33–35] is a popular wireless mesh networking standard. It finds its most frequent applications in traffic management systems, household electronics, and machine industry.

IEEE 802.15.4 supports low data exchange rates, low power operation, security, and reliability. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power consumption allows longer battery life, and the mesh networking provides high reliability.

ZigBee [36] is a standard of the ZigBee Alliance which is based on IEEE 802.15.4 for data communications. It is designed for low-power consumption and allows batteries to last from months to years. ZigBee standard provides network, security, and application support services operating on top of the IEEE 802.15.4 Medium Access Control (MAC) and Physical Layer (PHY) wireless standard. It employs a suite of technologies to enable scalable, self-organizing, self-healing networks that can manage various data traffic patterns.

The main characteristics of IEEE 802.15.4 are as follows:

- Low power consumption, with battery life ranging from months to years.
- Three license-free bands: 2.4-2.4835 GHz, 902-928 MHz and 868-870 MHz. The number of channels allotted to each frequency band is fixed at sixteen, ten and one respectively. The higher frequency band is usable worldwide, and the lower two bands in the areas of North America and Europe as shown in Table 1.1 [37].
- Maximum data rates allowed for each of these frequency bands are fixed as 250 kbps at 2.4 GHz, 40 kbps at 915 MHz, and 20 kbps at 868 MHz.
- Low data rate (250 kbps, 40 kbps, and 20 kbps) with low latency devices such as joysticks for low duty cycle applications (<0.1%).
- Channel access using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to access to the shared medium.
- 50m typical range.

Table 1.1: IEEE 802.15.4 Operating Conditions [4].

Frequency band	Number of channels	Datarate (kbps)	Applicability
2.4 GHz	16	250	WorldWide
915 MHz	10	40	USA
868 MHz	1	20	Europe

1.2.4 Summary and limitations

The huge growth in the number of devices requires features such as low cost with low power consumption of the devices, and an extended radio coverage.

However, these features are not adapted for the legacy wireless protocols such as WiFi [22], Bluetooth [30, 31] and ZigBee [38, 39]. These technologies are limited in that they cannot easily provide long range communication for devices that must operate at low power.

Regarding the power consumption, both WiFi and Cellular technologies deplete and damage the battery quickly, although cellular does it more quickly. This is one of the main reasons that neither of these technologies are suitable for IoT applications where it is difficult to recharge the battery easily for instance when measuring temperature at the surface of the sea [40].

Table 1.2 displays a brief comparison between the three aforementioned technologies. This table also presents the benefits and drawbacks of these technologies.

The limitations presented in the aforementioned technologies have led to the development of new wireless technologies designed for long distance, low power devices, and low cost connectivity to meet the requirements of the IoT applications. These wireless technologies have been designated as Low Power Wide Area Network (LPWAN) such as SigFox, Weightless and LoRaWAN.

1.3 Objectives

LoRaWAN [5] is a Low Power Wide Area Network (LPWAN) technology that has attracted much attention from the community in recent years. LoRaWAN has raised up as an important protocol for long-range communication of ultra low-powered devices. Nonetheless,

Table 1.2: Brief comparison of Bluetooth, 802.15.4, and WiFi protocols.

Standard	**Bluetooth**	**802.15.4/ZigBee**	**WiFi**
IEEE specification	802.15.1	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max Data Rate	1 Mb/s (slow data)	250 kb/s (slow data)	54 Mb/s (fast/high data)
Channel bandwidth	1 MHz	0.3/0.6 MHz; 2 MHz	22 MHz
Nominal range	10 m (short range)	10-100 m (short range)	100 m (short range)
Energy Consumption	Low	Low	High
Battery Drain	No	No	Yes
Applications	Not suitable for applications that need to send large files, live video, or have other bandwidth intensive data requirements	Not suitable for networks where high data rate and high mobility are needed	Not suitable for small battery powered devices that need to run for long periods of time

some challenges regarding the network performance still need to be addressed.

LoRaWAN performance depends on many factors such as the density of the devices in the network and the parameter settings. The existing research works have been studying and improving the overall LoRaWAN performance. Thus, during the five past years, different studies have analyzed the technology limits, and addressed metrics such as network throughput, energy consumption, and delay [41–50].

Throughput: LoRaWAN uses a pure ALOHA MAC layer. This behaviour may trigger many collisions, which negatively impacts the very limited LoRaWAN throughput. On the other hand, when two transmissions overlap at the receiver, the stronger signal may survive the collision thanks to the capture effect feature. However, collisions can not be totally avoided even when considering the capture effect.

Energy consumption: Many LoRaWAN devices, such as sensors or actuators, are not powered by the electricity grid. Therefore, it is crucial to investigate the energy consumption in LoRaWAN, in order to reduce the device energy consumption, lifetime and energy cost of data delivery. The goal of an IoT deployment is to conserve the energy without sacrificing the throughput.

Delay: The delay for the correct decoding of a frame in LoRaWAN is critical since it impacts the overall network throughput, as well as the energy consumption of the devices. The delay is defined as the difference between the reception of the frame by the network

server and the first time it is sent by the transmitter. An increase in the delay leads to a decrease in the performance of the whole network.

In this thesis, we make the following contributions:

i) We notice that the pure ALOHA used by LoRaWAN at the medium access control (MAC) layer is a performance bottleneck as the network size scales up. Therefore, it is important to study the performance of other MAC scheme in the context of LoRaWAN. Hence, we propose in [51] a method to separate fully synchronized LoRa collisions by relying on sending short frames instead of complete frames. In other words, we propose an efficient decoding of synchronized colliding LoRa signals based on layer 2 frame decoding. We also evaluate the performance of LoRaWAN under such setting in terms of throughput, delay experienced by transmitted uplink messages, and energy consumption of the transmitters.

ii) We find that for downlink communications, the impact of the duty cycle restriction at the gateway increases more when confirmed traffic is required by the transmitters. This duty cycle restriction represents a severe bottleneck in terms of confirmed frame success rate since successfully received uplink frames may not be acknowledged by the network server in due time. Furthermore, the missed acknowledgments exacerbate the uplink traffic load, triggering retransmissions of otherwise successfully delivered uplink frames. For these reasons, we study in [52] the selection of the gateway in downlink communication in order to evaluate and improve the performance of LoRaWAN in terms of throughput, delay and energy consumption. We evaluated the network performance under three different scenarios of gateway deployment (i.e. Urban, Environmental and Hybrid). Moreover, we studied three different algorithms and their impact with the three aforementioned scenarios on this selection.

Eventually, our work (summarized by the previous two points) in this thesis aims to enhance the following metrics in LoRaWAN in order to increase the overall network performance:

1. We aim to reduce frame collisions in order to increase the total throughput since it is small in LoRaWAN.

2. We aim to reduce the energy consumption of the transmitters in order to save their battery, as this latter is not always rechargeable.

3. We aim to reduce the delay of the frames in order to decrease data latency.

1.4 Thesis plan

The remaining of this thesis is divided into the following parts.

In **Part II**, composed of Chapters 2, 3 and 4, we present a state of the art of LoRaWAN performance and limitations.

In Chapter 2, we give an overview of the most known LPWAN protocols in the literature. First, we present LPWAN protocols such as SigFox and Weightless. Then, we focus on LoRaWAN.

In Chapter 3, we present some existing works done for LoRaWAN in uplink traffic. We classify the works in the literature into two categories, namely: works developed on LoRaWAN performance regarding the capacity, scalability and CSMA; and works developed on LoRaWAN collisions with the solutions proposed to increase the throughput in uplink communications while mitigating the negative impact of collisions. We set out to describe them in detail.

In Chapter 4, we present some existing works done for LoRaWAN in downlink traffic. We classify the works in the literature into two categories, namely: works elaborated on LoRaWAN performance regarding the downlink communications, scalability and network capacity; and works elaborated on the gateway selection in LoRaWAN with the solutions proposed to increase the throughput in downlink communications.

In **Part III**, composed of Chapter 5, we introduce a new MAC protocol to decode fully synchronized colliding LoRa signals and then we present the results.

In Chapter 5, we propose an improvement to the conventional LoRaWAN by including a slotted-backoff solution in order to mitigate LoRa collisions. We show that collisions are decreased but still exist, and the collided frames can not be decoded by the receiver. Therefore, we propose a new MAC protocol which aims to decode fully synchronized colliding LoRa signals by sending short bitmap frames to the receiver instead of retransmitting the complete collided frames. We show that our proposed MAC protocol brings a

significant improvement to the overall network performance. It improves the throughput of the network, and decreases the energy consumption of the transmitters, as well as the delay of the frames.

In **Part IV**, composed of Chapter 6, we work on the selection of the gateway for LoRaWAN downlink communications and then we present the results.

In Chapter 6, we present three scenarios of gateway deployments named Urban, Environmental and Hybrid. Then, we study the impact of each scenario on the selection of the gateway for three different algorithms. The first algorithm is based on balancing the load between the gateways, the second is based on choosing the gateway having the highest receive power, and the third is a combination of the two previous algorithms. We show that the throughput depends on the scenario of gateway deployment. Moreover, we show that balancing the load is more efficient for the throughput than choosing the gateway with the highest receive power. On the other hand, we show that balancing the load is less efficient for the delay and energy consumption than choosing the gateway with the highest receive power.

In **Part V**, we conclude this thesis work by summarizing our different contributions, and by giving the perspectives of our work.

Part II

STATE OF THE ART

Chapter 2

LPWAN Solutions

Contents

In the future, it is expected that network technologies will become progressively integrated into the human environment. Subsequently, massive volumes of data need to be processed, stored, and presented in an easily interpretable, efficient, and transparent form [7].

The limitations presented in the WiFi, Bluetooth and IEEE 802.15.4 technologies have motivated the development of new wireless protocols designed for long distance, low power, and low cost connectivity. These new wireless protocols have been designated as low power wide area network (LPWAN). LPWAN has become one of the fastest growing areas in the IoT [53]. LPWAN technologies aim to wirelessly connect large numbers of geographically dispersed devices at a low cost over wide area. Hence, LPWAN is designed to allow long range communications among low power consumption devices [15, 21, 54]. However, a low bit rate is achieved using LPWAN. It is well suited for IoT applications that need to transmit only small amounts of data over a long range. Most of LPWAN technologies can be separated into either wideband or ultra-narrow-band technologies [21, 55].

Many LPWAN technologies are being developed such as LoRaWAN [5], Sigfox [56], DASH7 [57, 58], Weightless [59], RPMA Ingenu [60] and 5G [61]. In the following, we provide a brief review about the most prominent LPWAN technologies.

2.1 SigFox

SigFox [56] uses Ultra-Narrow Band (UNB) modulation with Differential Binary Phase-Shift Keying at 100 bps (DBPSK). In SigFox, the device initiates a transmission by sending three uplink in sequence on three random carrier frequencies. The bandwidth that is assigned to SigFox communication in Europe is of about 192kHz. The size of a channel is of about 100Hz, which determines a total number of 1920 of channels [62]. The base station successfully receives the uplink even if two of the transmissions are lost due to collision with other transmissions or interference from other systems using the same frequency.

SigFox is a proprietary technology [63] which operates on sub-GHz frequencies on Industrial, Scientific, and Medical (ISM) radio bands: 868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia [64]. It is an LPWAN technology, founded and delivered in 2009 by the French company Sigfox.

SigFox protocol was designed for the transmission of small messages and is not suitable

for multimedia and permanent broadcast applications, which require a wide band. In addition, the UNB technique provides a very high reception sensitivity because less noise is added to the band. Each base station can handle a large number of connected objects [65], with a coverage area of about 20 to 25 km in rural areas, and 3 to 10 km in urban areas [66,67]. The duty cycle restrictions of the utilized subband in the 868 MHz EU ISM band is 1%. Therefore, a SigFox device may only transmit 36 seconds per hour. The time on air is 6 seconds [68] per message and thus the maximum is 6 messages per hour with a payload of 4, 8, or 12 bytes.

SigFox imposes a number of constraints on the messages transferred over the network. First, each end-device can send up to 140 messages per day [15]. Second, the payload of each message cannot exceed 12 bytes long at a data rate of up to 100 bps, which is sufficient for devices that transmit an alarm, a location, an environmental state (temperature) and a measure of energy consumption.

SigFox has a star topology similar to a cellular architecture, with a wide deployment of base stations aimed at covering entire countries. This topology permits nodes to upload the gathered data directly to SigFox servers, making it accessible to subscribers through a web-based Application Program Interface. The use of ISM bands together with SigFox medium access strategy, namely without collision-avoidance techniques, leads to a stringent bandwidth-occupancy limitation suffered by nodes [69].

Even though originally designed as a unidirectional system, SigFox has recently included a limited downlink window (four messages of eight bytes per end-user per day). Since the number of messages over the uplink is limited to 140 messages per day, and the number of messages over the downlink is limited to four messages per day, this means that the acknowledgment of every uplink message is not supported [64].

2.2 Weightless

Weightless [59] is the name of a set of three LPWAN open wireless technology standards for exchanging data between a base station and thousands of machines around it. These three standards are Weightless-N, Weightless-P, and Weightless-W.

Weightless-N supports a star network topology and operates in sub-GHz spectrum using UNB technology, with a range of several kilometers even in challenging urban en-

vironments [65]. Weightless-N uses a class of low-cost technology, very similar to that employed by SigFox. Thereby, ultra-narrow band (the Differential Binary Phase Shift Keying or DBPSK) modulation is adopted in order to provide unidirectional-only connectivity of up to 100 bps, exploiting ISM bands. Weightless-N allows a battery duration of up to 10 years, very low cost terminals, and a long connection range [69].

Weightless-P supports narrowband channels of 12.5 kHz, with Frequency Division and Time Division Multiple Access modes, bi-directional communication with an adaptive data rate from 200 bps to 100 kbps, time-synchronized aggregators, and low-cost highly energy-efficient modulations [65]. Weightless-P includes characteristics such as acknowledged transmissions, auto-retransmission, frequency and time synchronization. Compared with Weightless-N, Weightless-P provides a smaller range of 2 km, and its advanced features has a shorter battery lifetime of three years [69].

Weightless-W is a system with star topology operating in TV white space spectrum [65]. Weightless-W achieves two-way data rates from 1 kbps to 10 Mbps with very low overhead. Due to the extensive feature set provided by Weightless-W, the battery lifetime of nodes is limited to three years, and the terminal cost is high. The communication between the nodes and the base station can be established up to 5 km, depending on the environmental conditions [69].

2.3 LoRa Technology

Long Range (LoRa) is a PHY layer for a low-power wide-area network as shown in Fig 2.1. LoRa enables long-range transmissions (3 to 8 km in urban scenarios and 15 to 20 km in rural scenarios) with low power consumption (up to 20 years battery lifetime depending on the use).

2.3.1 LoRa

LoRa [65, 70, 71] is a physical layer (PHY) technology and a proprietary modulation technique developed by Semtech [72]. It permits long-range, low-power and low-throughput communications. It operates on the 433, 868 or 915 MHz ISM bands, depending on the region in which it is deployed (433 MHz in Asia, 433 MHz and 863 to 870 MHz in Europe,

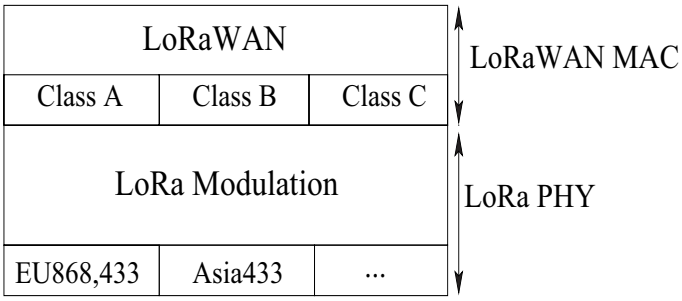


Figure 2.1: LoRa/LoRaWAN protocol stack.

902 to 928 MHz in United States).

LoRa technology uses *Chirp Spread Spectrum* (CSS) modulation [65] where symbols are encoded into signals of increasing (up chirp) or decreasing (down chirp) radio frequencies [73]. Figure 2.2 (a) shows a single LoRa up chirp, and Fig. 2.2 (b) shows a single LoRa down chirp. LoRa uses CSS making it robust to channel noise. The use of this modulation offers high performance in terms of range, by increasing the robustness of the signal and the sensitivity of the receiver while maintaining low power consumption. Therefore LoRa is suitable for long range and low bandwidth communications.

LoRa parameters and transmission options:
LoRa throughput and range depend on five main parameters: Transmission Power (TP), Carrier Frequency (CF), Bandwidth (BW), Spreading Factor (SF), and Coding Rate (CR).

1. **Transmission Power.** The TP on a LoRa radio can be adjusted from -4 dBm to 20 dBm, in 1 dBm steps. The transmission power of the end-device is set by default to 14 dBm [5]. The signal-to-noise ratio is increased by increasing the transmission power at the cost of energy expenditure[74].

2. **Carrier Frequency.** The CF represents the central transmission frequency used in a band. LoRa uses license-free sub-gigahertz radio frequency bands such as 868 MHz for Europe, and 915 MHz for Australia and North America. Different communication channels are used by LoRa devices.

3. **Bandwidth.** The BW is set to 125 kHz or 250 kHz in Europe. Larger bandwidth allows higher data rate, thus reducing transmission time at the expense of reduced sensitivity. A lower BW gives a higher sensitivity, but a lower data rate.

4. **Spreading Factor.** The SF (from 7 to 12), determines the duration of the symbol T_{sym} according to the following formula [75]:

$$T_{sym} = \frac{2^{SF}}{BW}$$

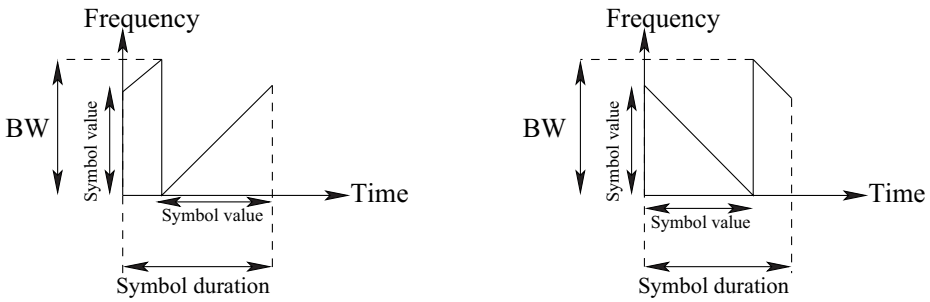
SF defines the number of bits encoded into each symbol. Each LoRa chirp consists of a linear frequency sweep and can encode 2^{SF} possible values [76]. It is an offset on the initial transmitted frequency compared to the minimum frequency of the channel. Actually, each chirp represents a symbol. For example, with SF=7, there are $2^{SF} = 128$ possible values per symbol. The duration of the sweep is called symbol duration, and depends on the SF as well as on the bandwidth as shown in Fig. 2.2. The SF has an influence on the transmission duration, the energy consumption, the robustness and the communication range. The lower the SF, the higher the data rate transmission but the lower the immunity to interference, thus the smaller the range. A large SF increases the Signal to Noise Ratio (SNR) and therefore the receiver sensitivity and the range of the signal. However, it reduces the transmission rate and thus increases the transmission duration and the energy consumption.

5. **Coding Rate.** The CR determines the rate of the Forward Error Correction code (FEC) as $4/(4+n)$ with $n \in [1;4]$. CR offers protection against bursts of interference: a higher CR gives more protection (i.e., the transmission is more robust), but increases the time on air [77].

The near orthogonality of the SFs allows the reception of several signals in parallel on the same channel, as long as they use different SFs. Consequently, concurrent transmissions with different SFs do not interfere with each other, and can be successfully decoded.

LoRa PHY frame format:

A LoRa frame begins with a preamble, which is used to keep the receiver synchronized with the transmitter. The preamble has a number of $n_{preamble}$ up chirp symbols with 4.25 LoRa symbols as frame delimiters for synchronization. In these 4.25 end symbols, the preamble starts with 2 up chirps, and ends with 2.25 down chirps. Therefore, the preamble is $(n_{preamble} + 4.25)$ symbols long. Typically, the preamble duration is 12.25 T_{sym}. After the preamble, there is an optional PHY header. When it is present, this



(a) LoRa up chirp. (b) LoRa down chirp.

Figure 2.2: Examples of LoRa chirps.

header is transmitted with a code rate CR of 4/8. The header includes a Header cyclic redundancy check (CRC) to allow the receiver to discard packets with invalid headers. Also, the header contains the length of the data information. The rest of the frame is encoded with the code rate specified in the PHY Header. The payload is sent after the header. The payload size is limited to 255 bytes. At the end of the frame there is an optional payload CRC as illustrated in Fig 2.3.

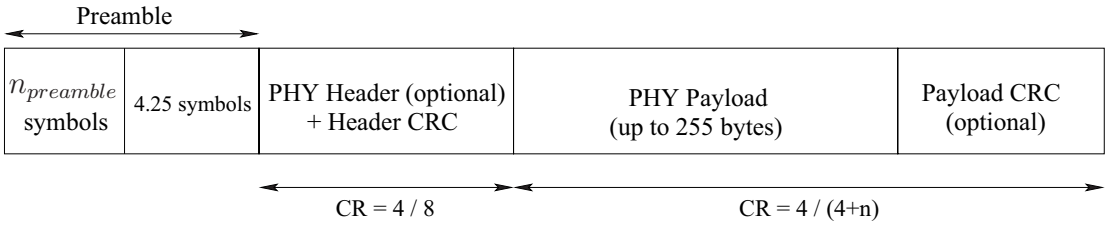


Figure 2.3: Structure of a LoRa frame.

For LoRa physical header and payload, LoRa chirps imposes another set of parameters including Header (H) and low data rate optimization enabled (DE).

Given BW, SF and CR, the time required to transmit a LoRa frame from an end-device to the gateway is the sum of the transmission time of the preamble $T_{preamble}$ and the payload $T_{payload}$ [72] as follows:

$$T_{frame} = T_{preamble} + T_{payload}$$

$T_{preamble}$ depends on T_{sym} and the number of preamble symbols $n_{preamble}$ as follows:

$$T_{preamble} = (n_{preamble} + 4.25) * T_{sym}$$

$T_{payload}$ depends on T_{sym} and the number of payload symbols $n_{payload}$ as follows:

$$T_{payload} = n_{payload} * T_{sym}$$

with

$$n_{payload} = max(\beta(CR + 4), 0) + 8$$

and

$$\beta = \left\lceil \frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)} \right\rceil$$

where PL denotes the size of the payload in bytes; $H = 0$ if the header is enabled, and 1 if it is not; $DE = 1$ if low-data rate optimization is enabled, and 0 otherwise.

2.3.2 LoRaWAN

LoRaWAN [5] is a MAC protocol built on top of LoRa technology, and developed by LoRa Alliance [78]. It enables communications between end-devices and a network server through gateways. While LoRa is a proprietary modulation from Semtech, LoRaWAN is an open standard. Besides LoRa modulation, LoRaWAN also supports the Frequency Shift Keying (FSK) modulation in the physical layer as an option.

Data rates for Europe range from 0.3 kbps to 50 kbps, and correspond to a parameter called DR. For DR0 to DR5, the bandwidth of the channel is equal to 125 kHz, and for DR6, it is equal to 250 kHz. SF12 is used for DR0, and the SF is decreased by one for each increase of the DR, until DR5 (included). For DR6, SF7 is used. This is depicted in Table 2.1 [5, 6].

The network channels can be freely attributed by the network operator. However the three default channels given in Table 2.2 must be implemented in every EU868MHz end-device. Those first three channels are the minimum set that all network gateways should always be listening on and must be implemented in every end-device. According to

Table 2.1: Data rates and related configuration for LoRaWAN 868 MHz EU band channel [5, 6].

Data rate	Configuration (Modulation/BW)	Indicative physical bit rate [bit/s]	Max. payload (bytes)	On-air time for 8-byte packet (ms)
DR0	LoRa: SF12 / 125 kHz	250	59	1581.056
DR1	LoRa: SF11 / 125 kHz	440	59	790.528
DR2	LoRa: SF10 / 125 kHz	980	59	452.608
DR3	LoRa: SF9 / 125 kHz	1760	123	226.30
DR4	LoRa: SF8 / 125 kHz	3125	230	127.9
DR5	LoRa: SF7 / 125 kHz	5470	230	70.91
DR6	LoRa: SF7 / 250 kHz	11000	230	35.46
DR7	FSK	50000	230	5

LoRaWAN specifications, the three aforementioned channels can be used both for uplink (UL) and downlink (DL) transmissions.

Table 2.2: EU863-870 default channels [5].

Modulation	Bandwidth [kHz]	Channel Frequency [MHz]	LoRa DR	Number of Channels	Duty cycle
LoRa	125	868.10 868.30 868.50	DR0 to DR5	3	<1%

In addition, the channels are regulated by different limitations on transmission power and duty cycle. In particular, the three bidirectional channels belong to the same sub-band and, hence, are subject to a common duty cycle limitation of 1%. An UL (respectively DL) transmission in any of such channels consumes the UL (respectively DL) duty cycle budget of all three channels. Instead, the DL-only channel at 869.525 MHz belongs to a different sub-band that permits a duty cycle of 10% and a larger transmission power.

LoRaWAN Architecture

The network topology of LoRaWAN is considered as star-of-stars and consists of three kinds of devices: end-devices, gateways, and network server [5] as shown in Fig. 6.1.

1. End-devices, which are basic nodes, typically consist in sensors or actuators that can transmit data to the network server through gateways. End-device to gateway communications can be either LoRa or FSK modulation with different data rates and channels.

2. Gateways receive frames transmitted by end-devices and forward them through a reliable connection (typically IP) to the network server. They also receive the net-

work server's acknowledgements (ACK) or MAC commands and forward them to the intended end-devices through LoRa.

3. The network server is the central network controller. It manages the gateways through standard IP technology. The network server provides authentication and authorization of end-devices, network encryption and decryption, data transmission, adapting data rates, elimination of duplicate packets, and interface with applications.

All communications can be bi-directional, although uplink communications from end-devices to the network server are expected to be predominant.

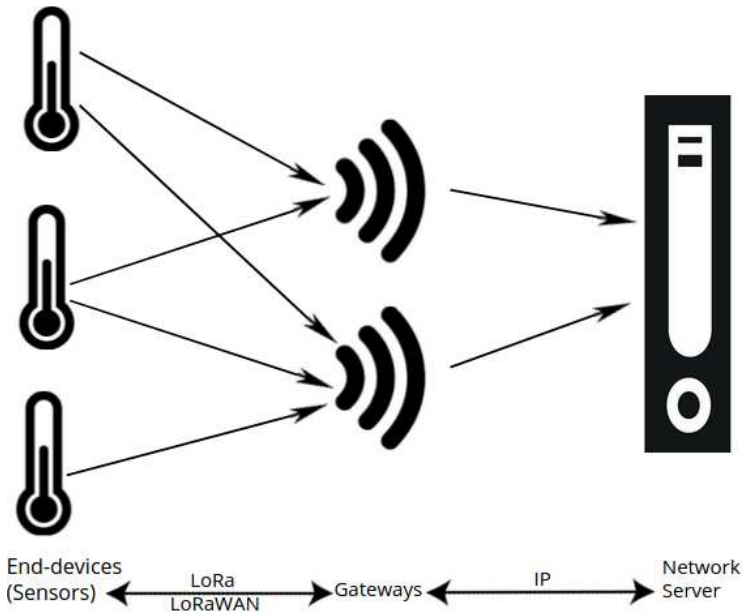


Figure 2.4: LoRaWAN architecture.

LoRaWAN Duty Cycle

In order to access the physical medium, the European Telecommunications Standards Institute (ETSI) regulations impose some restrictions on the maximum transmission power, on the duty cycle and on the maximum time a transmitter can transmit per hour. These restrictions differ by country/region, but we concentrate here on the European region. The ETSI regulations allow the choice of using either a duty cycle limitation or a Listen

Before Talk (LBT) transmissions management.

Based on the specifications [5], LoRaWAN only uses duty-cycled transmissions which limits the rate at which the end-device can transmit messages. In that sense, let d be the duty cycle in a given sub-band. Then the time required to transmit a packet in this sub-band, known as time on air T_a must be followed by a minimum off-period equal to

$$T_{off} = T_a(\frac{1}{d} - 1),$$

during which the channel is unavailable for transmission by this node. Thus the maximum duty cycle is the maximum percentage of time during which an end-device can occupy a channel.

To illustrate how a duty cycle limitation translates to a certain maximum time on air and minimum waiting time between consecutive packet transmissions, we consider a device transmitting on a channel with a 1% duty cycle. This device can perform 10 transmissions of 3.6 seconds within one hour. During the unavailable time of a given sub-band, the device may still be able to transmit on another sub-band. If all sub-bands are unavailable, the device has to wait before any further transmission. Table 2.3 shows LoRaWAN default channels and duty cycle limitations in Europe.

Table 2.3: LoRaWAN default channels and duty cycle limitations in Europe.

Frequency (MHz)	Direction	Duty cycle	Max transmission power (dBm)
868.1	DL, UL	1%	14
868.3	DL, UL	1%	14
868.5	DL, UL	1%	14
869.525	DL	10%	27

LoRaWAN Classes

End-devices in LoRaWAN can be configured to operate according to one of three different classes depending on how they schedule the reception of downlink traffic. LoRaWAN enables three classes of operation for end-devices [79]: class A (for All), B (for Beacon), and C (for Continuously listening), as shown in Fig. 2.5 [19, 80].

- **Class A:** in Class A which is the mandatory class, end-devices choose a random channel and send data when data is available (i.e., they use pure ALOHA access

for the uplink phase). Each uplink transmission is followed by two short downlink receive windows called Rx1 and Rx2 for the acknowledgments (ACK). The start time of Rx1 begins after a fixed amount of time following the end of the uplink transmission. By default, this delay is one second. Rx2 begins by default two seconds after the end of the uplink transmission. The end-device listens for possible ACKs during Rx1. If no ACK is received during Rx1, the end-device listens for possible ACKs during Rx2. After these listening periods, the end-devices switch to sleep mode to save energy until the next transmission. Downlink transmissions from the network server at any other time have to wait until the receive windows of the next uplink transmission. The delay between two transmissions has to be larger than or equal to 99 times the duration of the frame transmission in order to respect the duty cycle of 1%.
The end-device does not open the second receive window if it successfully receives a frame during the first receive window.

- **Class B:** in Class B, which is optional, end-devices open extra receive windows (ping slots) at scheduled times by receiving a time synchronized beacon from the network server via the gateways. This allows the network server to know when the end-device is listening. Any of these ping slots may be used by the network to initiate a downlink communication. In this class, end-devices must also implement Class A in parallel.

- **Class C:** in Class C, which is also optional, end-devices are always active and have almost continuous receive windows, only closed when transmitting. This is the lowest energy efficient class of devices.

Class B and Class C need more power than Class A, but they offer lower latency for communications between the network server and the end-devices.

Receive window parameters

The DR to be used in Rx1 is set as the uplink DR minus an offset called *RX1DROffset*. *RX1DROffset* can take values in the range of 0 to 5. Since *RX1DROffset* has a default value of zero, the DR for the first receive window is by default the same one used in the last uplink transmission as shown in Table 2.4. The frequency channel used in the first receive window is the same as the one used for the preceding uplink transmission.

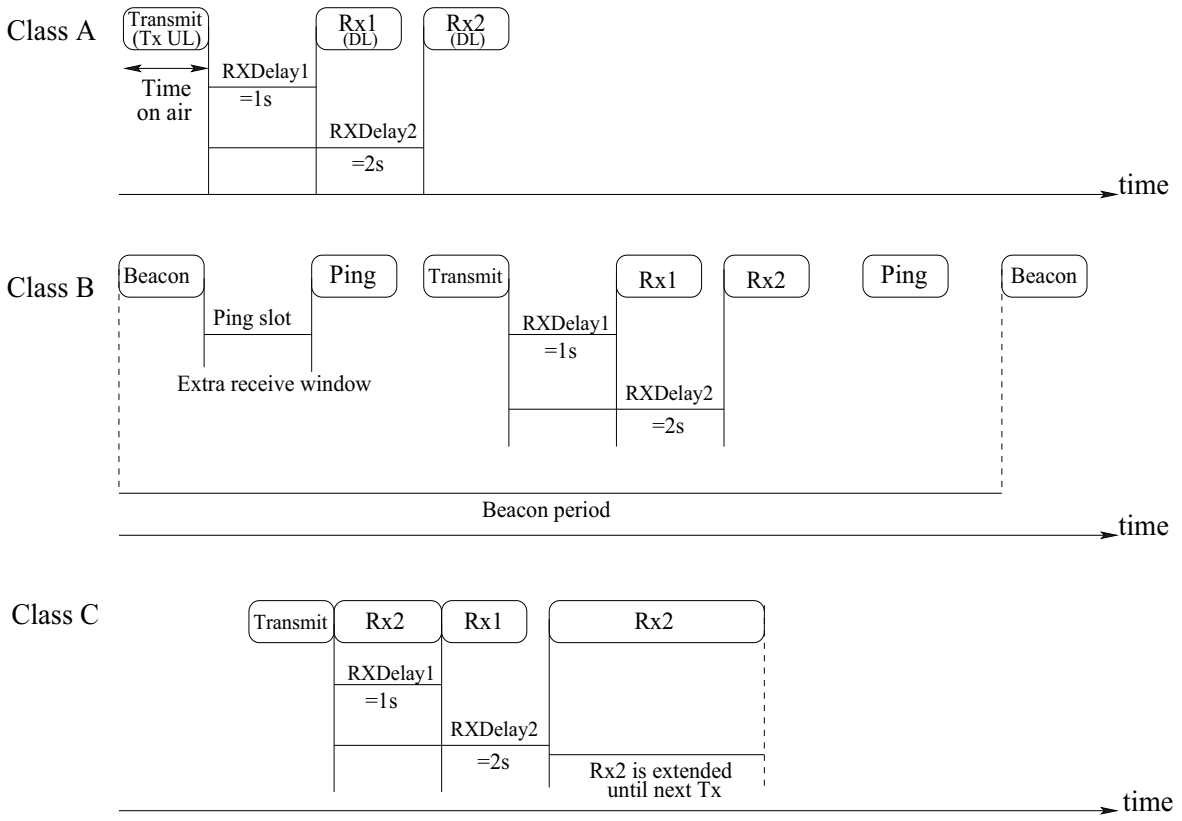


Figure 2.5: LoRaWAN class device communication mechanisms.

The second receive window uses a fixed data rate and frequency configuration by default DR0 and frequency 869.525 MHz.

Table 2.4: Basic physical layer parameters and their default values.

Parameters	Description	Default Value
DR_Rx1	Downlink data rate, 1st receive window	max(DR_Tx - *RX1DROffset*, DR0)
DR_Rx2	Downlink data rate, 2nd receive window	DR0
RX1DROffset	Data rate offset for the 1st receive window	0
RECEIVE_DELAY1	Delay from end of uplink transmission to start of 1st receive window	1s
RECEIVE_DELAY2	Delay from end of uplink transmission to start of 2nd receive window	2s

Adaptive Data Rate

The *Adaptive Data Rate* (ADR)[5, 81] allows the network server to adapt the transmit data rate of an end-device by changing the SF, in order to find the best trade off between energy efficiency and link robustness. In other words, the ADR automatically adapts to

the needs of the end-devices in order to limit the use of the bandwidth and therefore the energy consumption. In addition, ADR ensures good network performance and better scalability in terms of capacity. It is used for optimizing data rates, airtime and energy consumption in LoRaWAN networks.

For this purpose, end-devices located near the gateways should typically use higher data rates (i.e., use a lower SF) and the farthest end-devices should use the lowest data rates. In this way, the performance of the network is improved by implementing ADR.

Using ADR, the data rate of the end-device can be changed and controlled by the network server. The end-device sends a message up through the gateway, which simply passes the message along without acting on the data. This message includes information about the reception time and the signal strength. Based upon the strength of the received signal, the network server determines the optimal data rate for the end-device (that is, the spreading factor) by incrementing its data rate.

Retransmission

Uplink transmissions can be either unconfirmed or confirmed. Unconfirmed frames are transmitted only once and are not expected to be acknowledged by the network server. This means that if an unconfirmed frame is lost, the end-device can send the next frame independently on the reception of the previous frame.

Confirmed frames are expected to be acknowledged by the network server. The end-device expects to receive a downlink acknowledgment (ACK) during one of the two receive windows that immediately follow the transmission. If the ACK is not received, the end-device retransmits the same message until an ACK is received or until a maximum number of transmission attempts for the message is reached (8 by default). If the sender is an end-device, the network sends the ACK through one of the gateways in range using one of the receive windows opened by the end-device after the send operation. If the sender is a gateway, the end-device transmits an ACK at its own discretion. Note that ACKs are only sent in response to the latest message received and are never retransmitted. Furthermore, when confirmed traffic is employed, end-devices must wait ACK_TIMEOUT seconds before performing a retransmission, as defined in the LoRaWAN standard [5]. The value of ACK_TIMEOUT is a random delay, from 1 to 3 seconds. Moreover, the retransmission must obey the duty cycle limitation as any other normal transmission.

Regarding the ACK data rate, the LoRaWAN specifications recommend that ACKs

transmitted on Rx1 should use the same SF as the UL transmission, while ACKs transmitted on Rx2 should use the lowest available data rate (i.e. [SF=12, DR0]).

For confirmed frames and when no ACK is received, the end-device tries to retransmit the same data again. This retransmission happens on a new frequency channel, but can also happen at a different data rate (preferable lower) than the previous one as shown in Table 2.5. The DR to be used is recommended to follow the next rules. The first and second transmission attempts of a confirmed message are done by using the same DR, the third and fourth attempts use the next lower data rate (or DR0 if it was the DR previously used), and so on, until the 8th transmission attempt. After 8 transmission attempts of the same confirmed message without an ACK, the MAC layer should return an error code to the upper layer. Each retransmission is started after an ACK timeout (ACK_TIMEOUT) period, which is initiated at the start time of the last 2nd receive window as shown in Fig. 2.6. Thus, the retransmission starts after the transmission is done, and it is between RXDelay2 + 1 and RXDelay2 + 3 seconds. Any further transmission uses the last DR used.

Table 2.5: Data-Rate Adaptation during message retransmissions.

Transmission attempt	**Data Rate**
1 (first)	DR
2	DR
3	max(DR-1,DR0)
4	max(DR-1,DR0)
5	max(DR-2,DR0)
6	max(DR-2,DR0)
7	max(DR-3,DR0)
8	max(DR-3,DR0)

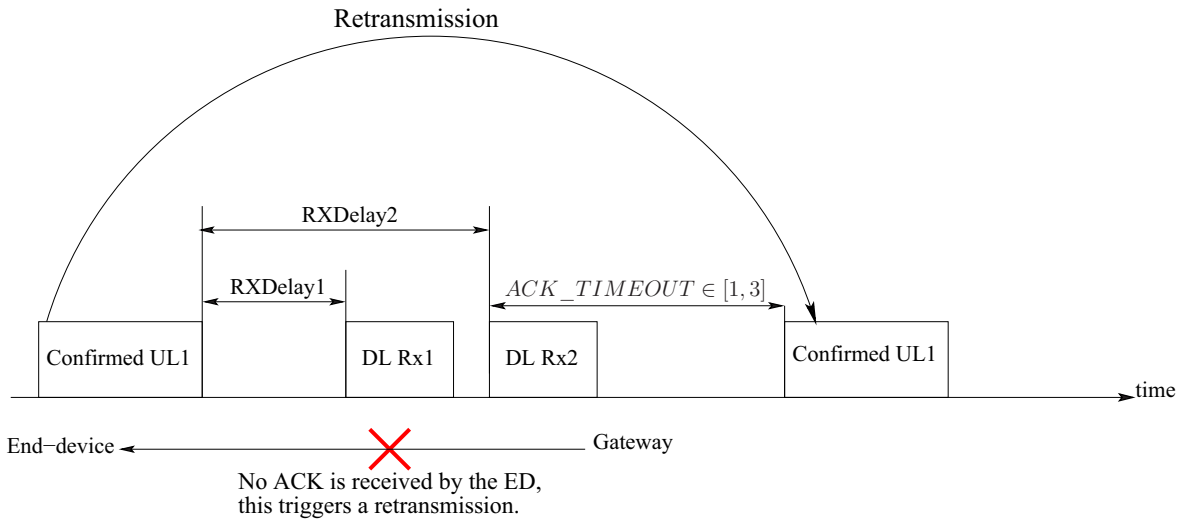


Figure 2.6: LoRaWAN Retransmission Procedure.

Summary LPWAN solutions

To summarize, Table 2.6 presents a brief comparison of the characteristics of SigFox, Weightless and LoRaWAN technologies.

Table 2.6: SigFox, Weightless and LoRaWAN technologies comparison.

	SigFox	Weightless-N	Weightless-P	Weightless-W	LoRaWAN
Band	868/915 MHz	433/868/915 MHz	433/868/915 MHz	TV whitespace	433/868/780/915 MHz
Max. data-rate	100 bps	100 bps	100 kbps	10 Mbps	50 Kbps
Range (urban)	10 km	5 km	2 km	5 km	5 km
Packet-size	12 B	20 B	10 B	10 B	256 B
Downlink	Yes	No	Yes	Yes	Yes
Topology	Star	Star	Star	Star	Star-of-stars

In this thesis, we are working on LoRa/LoRaWAN and not on other long-range LPWAN technologies because LoRaWAN is an open standard protocol.

Chapter 3

State of the Art for LoRa collisions in Uplink Communications

Contents

3.1 Collisions in LoRaWAN

The LoRaWAN specification allows end-devices to transmit at any time. There is no clear channel assessment to avoid collisions. Nodes can transmit on any available channel at any time using any available data rate, as long as they respect the duty cycle limitation (in Europe).

Even though multiple signals from different end-devices might arrive at the gateway at the same time, they can be successfully decoded as long as they use different channels or SFs.

Figure 3.1 illustrates simultaneous frame transmissions in LoRaWAN.

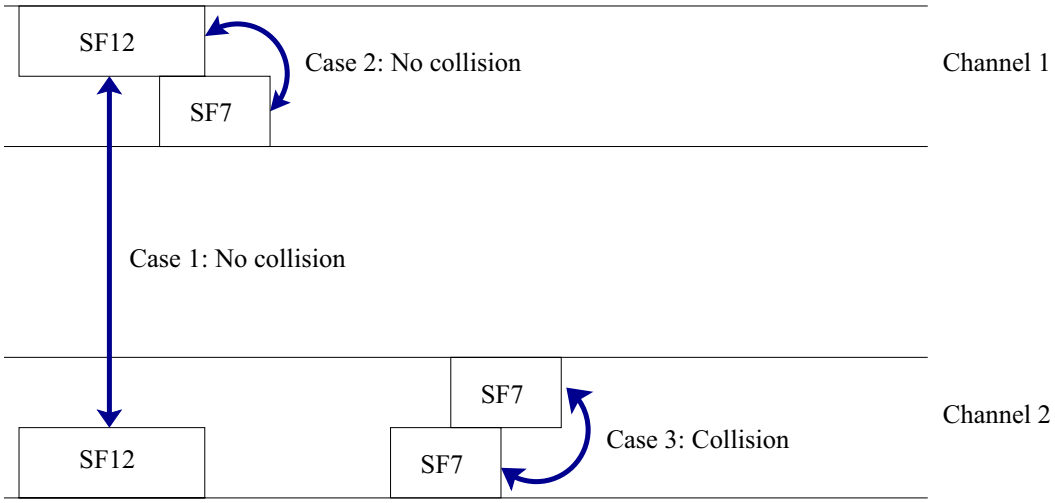


Figure 3.1: Simultaneous frame transmissions in LoRaWAN.

1. In case 1, although two signals generated with SF12 arrive at the same time, they can be successfully decoded because they use different channels.

2. In case 2, owing to the quasi-orthogonality of SFs, there is no collision, even though multiple devices use the same channel.

3. In case 3, a collision occurs since the two signals are generated with the same SF (i.e. SF7), arrive on the same channel and overlap in time.

Therefore, a collision occurs only when the multiple signals generated with the same SF overlap on the same channel.

However, despite these conditions which may cause the loss of the frames, one may survive if it satisfies the capture effect condition. When two LoRa transmissions overlap at the receiver, there are several conditions which determine whether the receiver can decode one or two frames, or none at all ([82],[83]). These conditions depend on power, timing, Carrier Frequency (CF), and Spreading Factor (SF).

Power: As LoRa is a form of frequency modulation, it presents the *capture effect* that occurs when two signals are present at the receiver and the weaker signal is suppressed by the stronger one. Therefore, frame x collides with frame y when

$$|P_x - P_y| < P_{Threshold}$$

where P_x is the received signal strength of transmission x, P_y is the received signal strength of transmission y, and $P_{Threshold}$ is a power threshold equal to 6 dBm ([82],[83]). If the condition holds, both frame x and frame y get corrupted.

Timing: Two frames overlap in time when their reception time intervals overlap. In other words, two frames x and y do not overlap with each other if

$$b_x <= a_y \;||\; a_x >= b_y$$

where the frame reception starts at time a and ends at time b.

Carrier Frequency (CF): When two frames overlap in time, but not in CF, they do not interfere which each other and can both be decoded. Therefore, we can define the condition when two frames collide as

$$f_x = f_y$$

where f_x and f_x are the frequencies of frames x and y.

Spreading Factor: The spreading factors used in LoRa are considered quasi orthogonal. In other words, signals modulated with different SFs are almost orthogonal: even if overlapping in time and frequency, two or more signals transmitted with different SFs

can be successfully decoded, provided that their received powers satisfy the condition of capture effect. Therefore, we define the condition on when two frames collide on SF as:

$$SF_x = SF_y$$

Therefore, if two or multiple frames are sent with the same SF, on the same channel, and they overlap in time, there is a need to have at least one receiver where the capture effect is satisfied (i.e., $|P_x - P_y| > P_{Threshold}$) in order to ensure that a frame can eventually be received as shown in Fig. 3.2. Here, two frames x and y are generated with the same SF (i.e. SF7), arrive on the same channel and overlap in time. Hence, they collide with each other. But the strongest frame may survive the collision if its signal is greater than the signal of the other frame by a minimum of 6 dBm.

The capture effect has a significant impact on the achievable throughput. In many situations, at least one of the colliding transmissions can be received successfully and survive the collision after taking into account the capture effect.

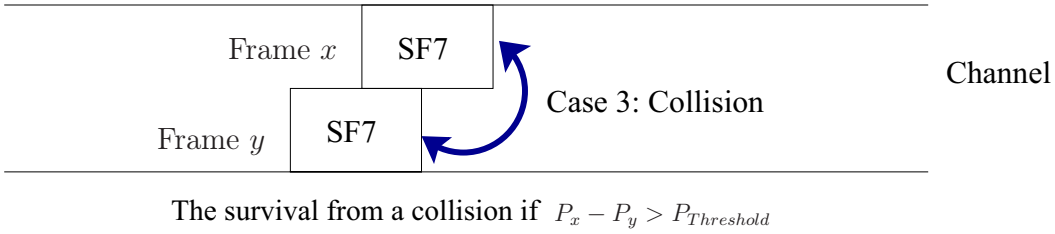


Figure 3.2: Condition for a frame to survive a collision.

3.2 Related work on uplink LoRa communications

Advanced receivers in LoRa (typically LoRaWAN gateways) are able to decode superposed signals when they are sent on different SFs or on different channels. When signals are sent on the same channel and with the same SF, they risk to collide, unless the strongest signal is captured by the receiver.

There have been a few works dealing with LoRa collisions. Some researchers such as [3, 6, 41, 77, 82, 84–92] have studied the collisions in LoRa and their impact on the throughput.

In [77], the authors model a LoRa network consisting of nodes with different communication settings in terms of bandwidth and spreading factor. They compute the average success probability per configuration as a function of density taking into account both intra and inter-SF collisions. They also formulate and solve an optimization problem to maximize the node capacity for a given deployment area by optimizing the number of nodes having different SF configurations. They present numerical results and show that solutions close to the optimal can increase the maximum number of nodes by more than 700% compared to the case where an equal number of users per SF is considered. Indeed, on one hand, the time on air of frames increases significantly by increasing the number of users having higher SFs and, thus, there is a higher probability of intra and inter-SF collisions. On the other hand, the number of intra-SF collisions may be also high if a high number of nodes with the same SF is deployed.

In [6], the authors address the problem of improving the network scalability for LoRaWAN. They show that the conventional method for assigning the SF parameter to the devices in a LoRaWAN network, which effectively minimizes the consumption of individual devices, has some drawbacks when it comes to the scalability of the network as a whole. Therefore, they propose another method of assigning the SFs to the nodes, which improves the probability of data delivery at the cost of a minor increase of the devices consumption. They have formulated their optimization problem, and demonstrated the operation of the proposed SF assignment strategy for a scenario where all end-devices experience identical radio conditions. The results of the conducted simulations confirm and characterize the utility of the proposed method which increases the probability of uplink data delivery by 20% to 40%.

In [82], the authors investigate the use of directional antennas and the use of multiple base stations as methods of dealing with inter-network interference. They compare the effectiveness of these two approaches via simulation. They show that both methods are able to improve LoRa network performance in the presence of interference as well as the reception rate. However, the results show that the use of multiple base stations clearly outperforms the use of directional antennas. For example, in a setting where data is collected from 600 nodes which are interfered by four networks with 600 nodes each, using three base stations improves the Data Extraction Rate (DER) from 0.24 to 0.56 while the use of directional antennas provides an increase from 0.24 to only 0.32. Furthermore, the

authors show that when the distance to the interfering base stations increases, the DER of a deployed LoRa network increases. Also, with more interfering networks, DER decreases significantly in particular when the number of nodes is high.

In [3], the authors conduct a series of experiments to verify the promises Semtech has made in terms of transmission distance, end-device lifetime, and node capacity. Their results show that LoRa is capable of communicating over 10km under line-of-sight environments. However, under non-line-of-sight environments, LoRa performance is severely affected by obstructions such as buildings and vegetations. Results show that LoRa is capable of communicating up to 4km and 5km for Packet Reception Rate (PRR) of 90% and 70% respectively by using. Regression were performed on available data to extrapolate the PRR for distances beyond 9km. With the extrapolated data, LoRa is expected to be able to support up to 10km using SF12 with PRR 70%. Although the capability of LoRa receivers to demodulate colliding packets on different SFs is in contrast with other LPWAN technologies, a LoRa gateway could still experience collisions from packets with the same SF. However, collisions could be averted by leveraging a Carrier Sense Multiple Access (CSMA) mechanism prior to transmission. Carrier Activity Detection (CAD) mode as shown in Fig. 3.3 is capable on sensing a potentially colliding packet during the transmission of the preamble. The authors exploit this mode by implementing a simple CSMA mechanism and discover that using CSMA-CAD provides up to 20% PRR improvement.

In [84, 85], a study on LoRaWAN scalability has been presented and analyzed, where the authors developed a mathematical model of the transmission process. They concluded that the network capacity is of only one message with a payload of 51 bytes every ten seconds. This capacity corresponds to 5000 end-devices each transmitting two messages per day [84]. This capacity is for confirmed uplink traffic.

In [86], the authors show their results regarding the data transfer for a single end-device in LoRaWAN networks. They show that nodes near the gateway can send only 2 kbit/s in the uplink. The maximum upload rate available for the more distant end-devices decreases with the distance between the end-device and the gateway and for the most distant end-devices drops to 100 bits/s in average. Moreover, authors show that the absence of clear channel assessment mechanism increases the probability of packet collisions. This absence also threatens the reliability, and may cause long channel access delays due to channel access closure after previous data transfers.

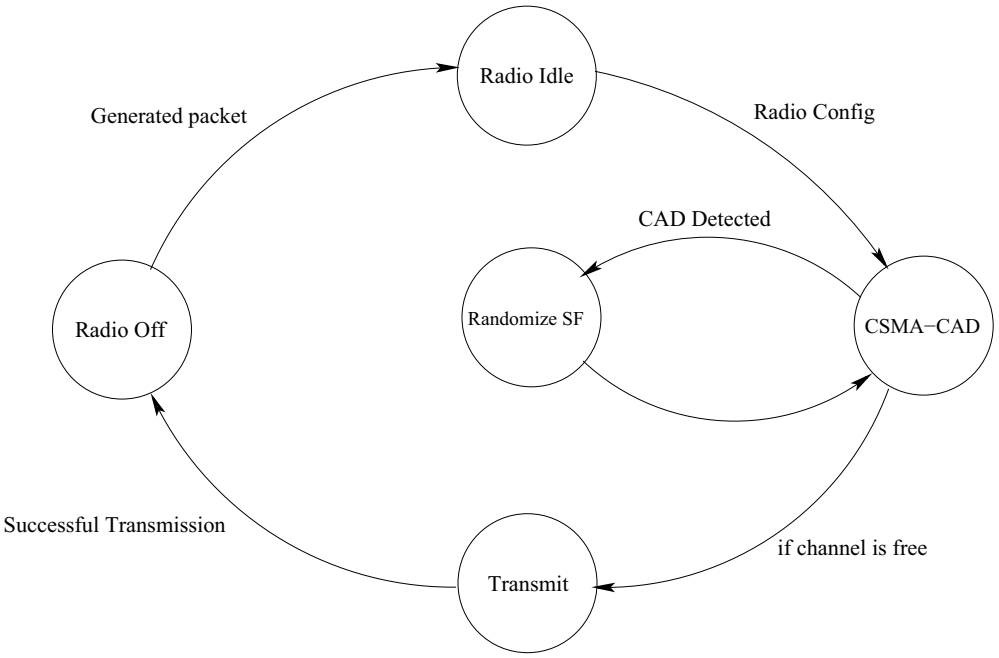


Figure 3.3: The CSMA-CAD mechanism for an end-device [3].

In [87], the authors investigate the scalability in term of the number of end-devices for a single gateway LoRaWAN deployments. First, they determine the intra-technology interference behavior with two physical end-devices, by checking the impact of an interfering node on a transmitting node. Measurements show that even under concurrent transmissions, one of the packets can be received under certain conditions. Based on these measurements, they create a simulation model for assessing the scalability of a single-gateway LoRaWAN network. They show that when the number of nodes increases up to 1000 per gateway, the losses will be up to 32% in LoRa. While in pure Aloha the losses are around 90%.

Fig. 3.4 shows all of the possible interfered positions. Table. 3.1 shows for each case whether the packet is received correctly, lost or received with the wrong payload CRC according to the measurements. The last column shows how they classified the packet in their model. Based on these results from real measurements, they make the following conclusions:

1. If the interferer starts after the preamble of the interfered, and the RSSI from the

interferer is at the same level or lower than the interfered transmission, then the interfered transmission will be received correctly.

2. If the interferer starts after the end of the preamble and the header, and has a higher RSSI at the receiver, then the first transmission will be received with the wrong payload CRC.

3. If the last six symbols of the transmitter preamble are received correctly, the receiver can synchronize with the transmitter. This means that the frame is received correctly.

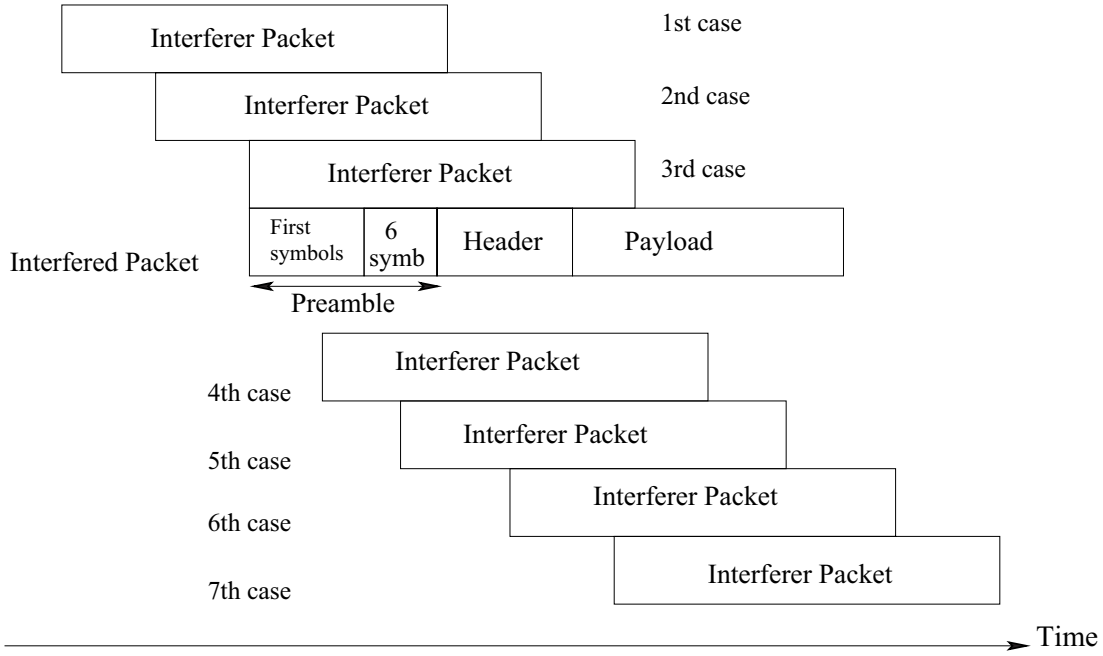


Figure 3.4: All the possible cases where an interferer can collide with an interfered.

In [88], the authors presented an in-depth investigation of LoRaWAN frame collisions and the capture effect in particular through various experiments. They focused on correct reception of data at the application, instead of at the gateway, and they considered multi-gateways, dense scenarios. For example, their experiments showed that using multiple gateways instead of a single gateway increases the probability of receiving correct frames. Their results show that for a single gateway, the Data Extraction Rate (DER) mostly

Table 3.1: Status of the packet from the interfered transmission, for the cases from Fig. 3.4.

Cases from Fig. 3.4	Interferer RSSI >	Interferer RSSI ∼	In the model of Paper [87]
1st case	Lost	Lost	Lost
2nd case	Lost	Lost	Lost
3rd case	Lost	Lost	Lost
4th case	Lost	Received mostly correctly ∼90%	Lost
5th case	Lost	Received mostly correctly ∼90%	Lost
6th case	Lost	Received mostly correctly ∼90%	Lost
7th case	Received with the wrong CRC	Received correctly	Received with the wrong CRC or correclty based on RSSI

depends on the distance between the end-devices and the gateway, and that the DER declines gradually as the number of end-devices increases. Adding more gateways improves the DER. Furthermore, they found that most frames hardly reached the more distant gateways, which is possibly due to the low SF used most of the time. While the weaker frames could not reach the more distant gateways, the stronger frames could be decoded properly. Collisions can also aggravate the situation, especially for the frames that use large SFs and required longer time on air.

In [89], the authors aim at assessing the performance level of LoRaWAN by analyzing the number of packet collisions that can occur. In addition, they proposed a series of solutions for reducing the number of collisions and increasing the capacity of the communication channel. They studied the percentage of frame collisions in three different cases represented in Table 3.2. Configuration A is most often used in practice, since it ensures

Table 3.2: Simulation parameters in three different cases.

Parameters	Configuration A	Configuration B	Configuration C
SF	12	6	12
BW(kHz)	125	500	125
CR	4/5	4/5	4/8

the largest communication range. Configuration B corresponds to the fastest transfer rate; this is the reason of the lowest error rate observed. Because in configuration B the airtime of the frame is the lowest, the probability of a collision occurring is low. In configuration C, the airtime is greater than that in B. Therefore, the percentage of collisions is much higher compared to configuration B. Thus, the number of collisions for 100 nodes is 26.6% for configuration A, 9.73% for configuration B, and 31% for configuration C. On the other

hand, their conducted results show that the maximum number of nodes which can communicate on a LoRa channel is approximately 875 nodes for configuration A, and 1000 nodes for configuration C.

In [90], the author analyzed the collision and packet loss in LoRaWAN. Based on the LoRaWAN features, he developped closed-form expressions of collision and packet loss probabilities. Simulation results confirm his theoretical developments. He also showed that his theoretical expressions are more accurate than the Poisson distributed process to describe the collisions. For instance, he compares the probability of at least one collision for a payload size between 1 and 59 bytes, when 1000 nodes are considered for each SF. The results can be grouped in three parts providing the same performance. The first part is composed of SF 7 and 8, the second of SF = 9, and the third of SF 10, 11, 12. Indeed, SF 10 and 11 give the same probability of collisions as SF 12 but they offer a lower sensitivity.

In [91], the authors presented an approach to increase the network throughput through a Slotted-ALOHA (S-ALOHA) overlay on LoRaWAN networks. Their method is based on an innovative synchronization service that is suitable for low-cost wireless sensor nodes. They modelled the LoRaWAN channel with extensive measurement on hardware platforms, and they quantified the impact of tuning parameters on physical and MAC layers, as well as the packet collision rate. In Slotted-ALOHA, the channel time is divided into slots, which have fixed length T and are composed of two parts: a transmission time (T_r) and a tolerance interval (T_b), as shown in Fig. 3.5.(a). Every end-device must transmit a frame only at the beginning of a slot. If two or more end-devices transmit their packet during the same slot, a collision occurs; otherwise, no collision is generated, and the data are properly sent (Fig. 3.5.(b)). Results show that Slotted-ALOHA significantly improves the performance of traditional LoRaWAN networks regarding packet loss rate and network throughput. Afterwards, the authors perform an S-ALOHA implementation over LoRaWAN, where a slotted LoRaWAN (S-LoRaWAN) was developped using the Slotted-ALOHA. Then they compared it with the LoRaWAN standard protocol. Results show that in the case of the LoRaWAN standard protocol, the overlapping of the frames is much more probable than with S-LoRaWAN. The throughput improvement is 5.8 times larger in S-LoRaWAN compared to the LoRaWAN standard protocol, with a reduction of packet collisions of 26%.

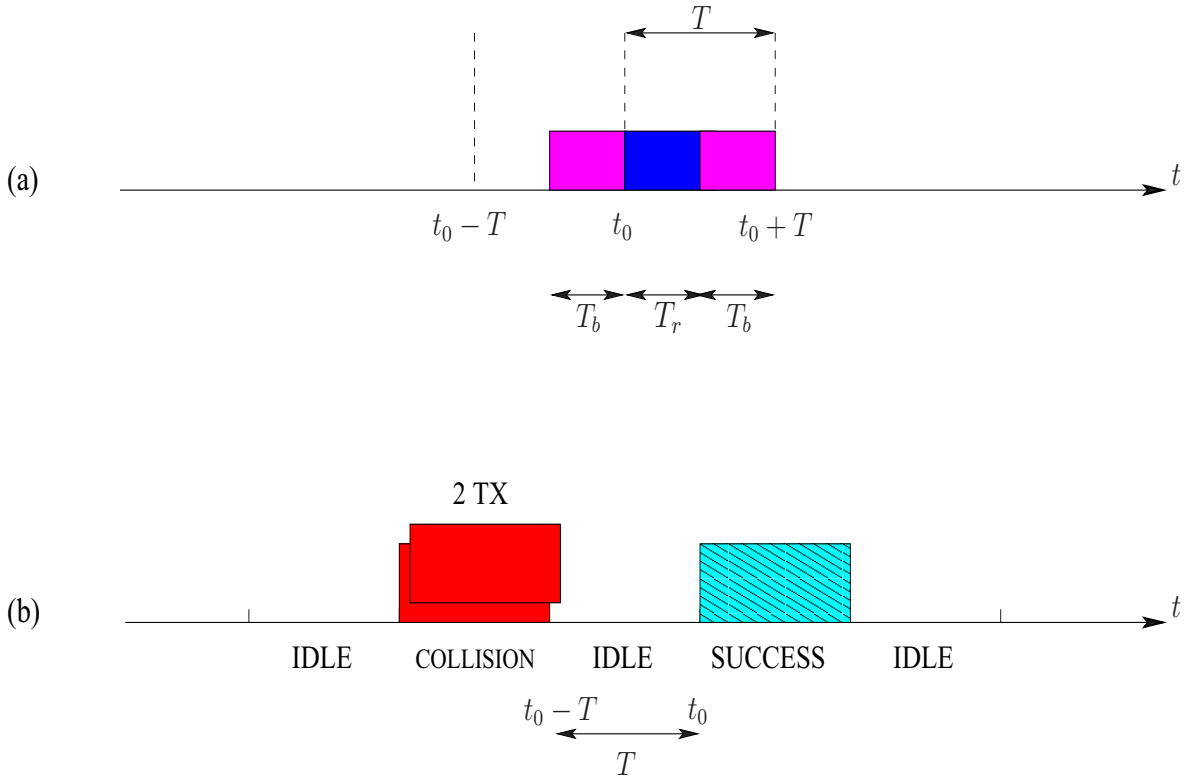


Figure 3.5: (a) Slot width definition in S-ALOHA, (b) example of idle, collided and successful slot.

In [92], the authors first presented a NS-3 module that simulates the behavior of LoRa in an accurate way. They show that the module correctly represents the capture effect that lowers the packet drop rate resulting from collisions. Second, the authors wanted to improve the performance of LoRa devices while not impacting energy consumption. They have used the NS-3 simulator to evaluate CSMA and CSMA-x, the proposed enhanced access methods that lower the collision ratio.

Principle of CSMA: Let us assume N contending devices. When an end-device $i \in N$ has a frame to send, it randomly chooses a communication channel c_i. It performs CCA (Clear Channel Assessment) to test if there is an ongoing transmission on the channel. Only when the channel is clear, the device starts its transmission, otherwise, it goes to sleep for a random duration and attempts a transmission later on. The random interval is equal to k slots, where $k \in [0, 2^n - 1]$ for the n^{th} transmission attempt (the maximum

value of n is set to 3). Fig. 3.6 illustrates the principle.

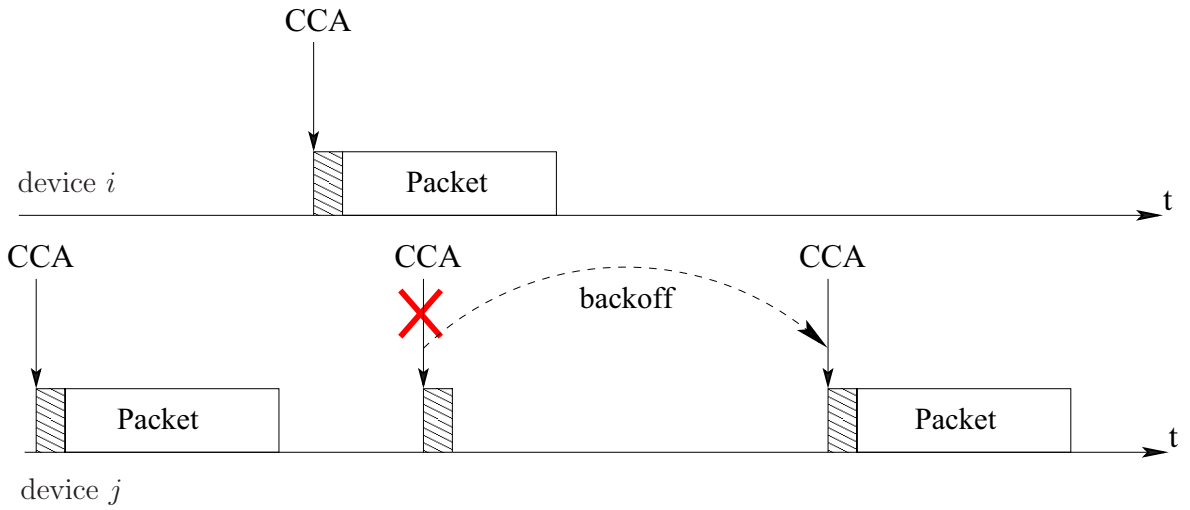


Figure 3.6: Principle of CSMA: device j sends a packet after a CCA and backs off when it detects that the channel is busy.

Principle of CSMA-x: Another variant of CSMA called CSMA-x is to listen to the channel for a small interval of time called CCG (Clear Channel Gap) before attempting a transmission. For instance, CSMA-10 corresponds to CSMA with an interval of 10 ms before a transmission. When the device detects a transmission during this interval, it backs off as in the basic CSMA. Fig. 3.7 illustrates the principle.

The simulation results of [92] show that CSMA considerably lowers the collision ratio while only slightly increasing energy consumption. The authors also observe that CSMA-10 presents lower energy consumption than LoRaWAN for a large number of devices. Another advantage of CSMA-x consists of increased throughput and larger network capacity because the ETSI restrictions on the duty cycle do not apply, as listen before talk is used.

In [41], the authors state that the ADR provides a chance to optimize the total throughput in LoRaWAN. Nevertheless, if most devices use the same data rate without considering the contention problem, the throughput may be reduced. Thus, the authors propose contention-aware ADR to get an optimal throughput, and they find the optimal set via the gradient projection method. In particular, when a large number of devices have similar link quality, namely in the case of biased usage of the SFs, the proposed method

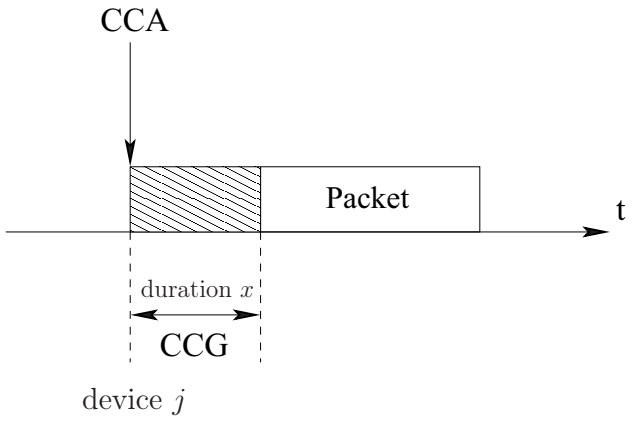


Figure 3.7: Principle of CSMA-x: device j sends a packet after a CCA and a CCG interval.

can achieve considerably higher throughput than the current system owing to the load balancing effect. The authors said that practical issues should be considered also, such as duty cycle of downlink, and consideration of all SFs in the evaluation. Their main goal was throughput optimization. Therefore, the data rate is adjusted in the direction of increasing the number of devices using small SFs. Although their policy increases overall throughput, the transmission success ratio of the devices decreases.

3.3 Related work on synchronized collided LoRa signals

In [93], the authors worked on constructing an efficient multi-hop network based on the sub-GHz LPWAN technology. They investigated the combination of LoRa and concurrent transmissions (CT). CT is a flooding protocol that considers synchronized packet collisions that happen when multiple relays perform immediate retransmissions at the same time. They found that, due to the time domain and frequency domain energy spreading effects, LoRa is robust to the packet collisions resulting from CT. They found that the receiver performance under CT can be further improved by introducing timing offsets between the relaying packets. Therefore, they proposed a timing delay insertion method, the offset-CT method, that adds random timing delay before the packets while preventing the timing offset from diverging over the multi-hop network. The authors refer to the CT-based multi-hop LoRa network based as CT-LoRa. Their experiments demonstrate

the feasibility of CT-LoRa multi-hop network, and the performance improvement brought by the CT method.

The CT flooding multi-hop protocol: The CT flooding is a link-layer protocol for wireless multi-hop networks. While flooding describes the broadcast-based network-layer behavior, CT describes the link-layer behavior of the relay nodes. Instead of trying to avoid packet collisions, CT allows multiple nodes to transmit packets that carry the same content simultaneously. By allowing such synchronized packet collisions, CT enables fast back-to-back packet relaying which greatly improves the efficiency of the network. As illustrated in Fig. 3.8, in each packet flooding, there would be one and only one node serving as the initiator. The initiator broadcasts the first packet and triggers the flooding. Every node that successfully receives the packet for the first time shall then perform immediate retransmission as another broadcast. The same procedure carries on until the packet floods over the whole network. The CT protocol helps to realize a simple but efficient one-to-all fast packet broadcast by allowing the synchronized packet collisions.

The offset-CT Method: In order to enhance the time-domain energy spreading effect and further improve the receiver reliability, the authors further propose the offset-CT method. It is a simple but effective method that increases the timing offset between the packets while maintaining a virtual timing alignment of each hop. The novelty of this proposal is twofold. First, for the practical CT-LoRa usage where the transmitter number cannot be determined, they propose to introduce a random timing delay uniformly distributed between 0 and one-symbol time before every retransmission of the relay packet. Second, in order to prevent the timing offset from diverging, they propose to carry the delay information in each packet, so that the relay that successfully demodulates the packet could insert the complementary delay to align the timing. The offset-CT is a timing delay insertion method. Specifically, the authors propose to insert a two-part delay before each retransmission of the relay packet, called as Part-A and Part-B delay. Fig. 3.9 illustrates an example of the timing diagram of the offset-CT method. Before each retransmission, part A and part B delays are inserted. The Part-A delay τ^A is a random variable uniformly distributed between 0 to T_S where T_S is the symbol time. The information of τ^A is carried in each packet. The relay which successfully decodes the packet would first insert a Part-B delay with a duration of $T_S - \tau^A$, and then inserts another newly generated random Part-A delay before the retransmission. The duration of each relay is fixed as

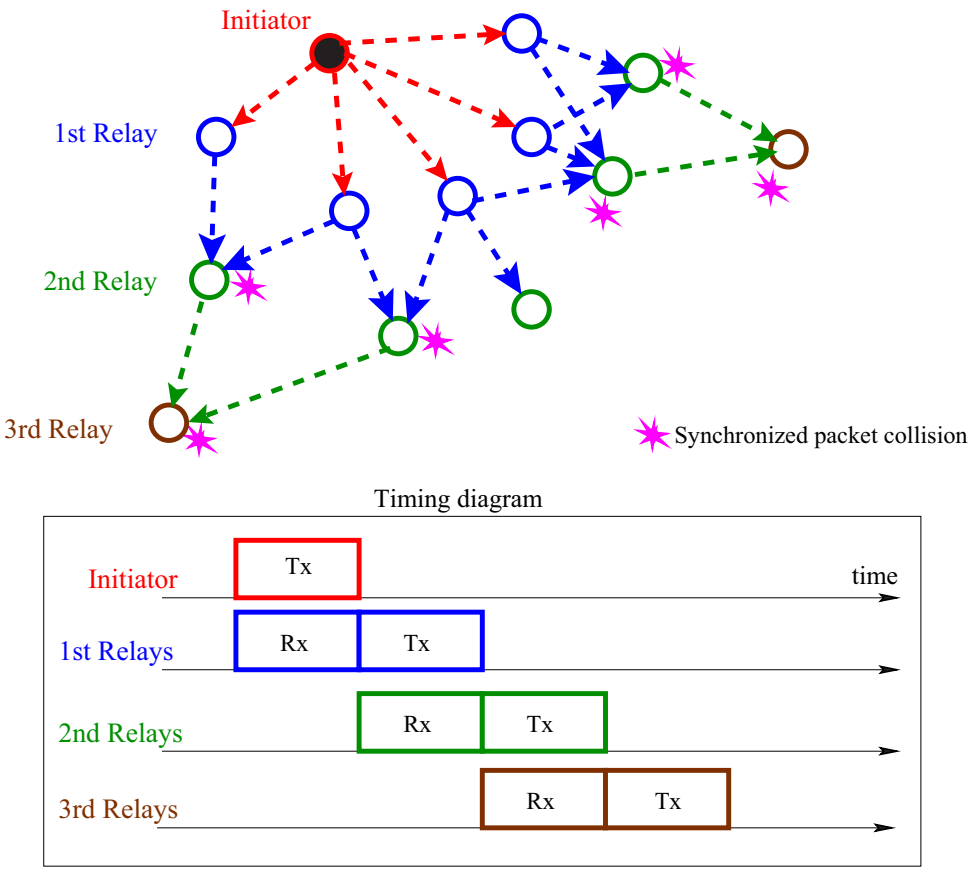


Figure 3.8: The operation of the CT flooding protocol. An initiator starts the first packet transmission, and the other relay nodes simply do immediate retransmission after the reception. The total replay can be shrinked to only a few packet lengths. On the other hand, synchronized packet collisions may happen frequently in this protocol.

$T_P + T_S$ where T_P is the packet length. The packets are allowed to be randomly shifted in a range of T_S in each hop. Note that in the first-hop relay, the Part-B delay is always zero.

Their results showed that CT-LoRa experiences a high packet reception rate performance under the typical multiple-building area network scenario. Moreover, they showed that LoRa survives the CT purely by capture effect which is considered in order to increase the probability of decoding colliding LoRa signals. If the colliding signals are not decoded with the capture effect, they are considered lost. Moreover, their results show that offset-CT significantly improves the PRR as well as reduces the average hop count

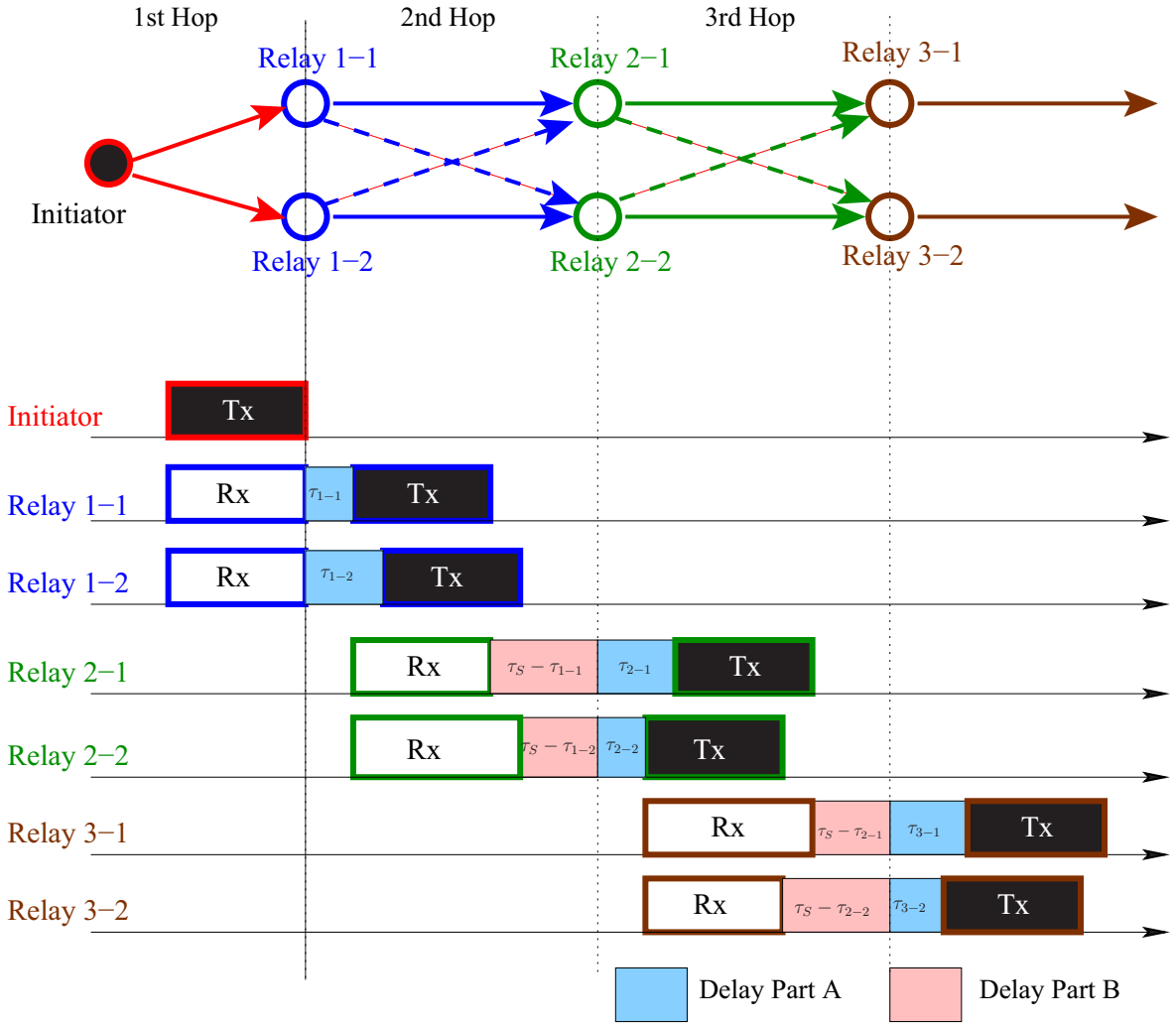


Figure 3.9: Illustration of the offset-CT method.

in the critical scenario (where multiple nodes are put closely to each other to make the power offset between the packet very small).

In [94], the authors presented Choir which is a system that improves throughput and range of LPWANs in urban environments. Choir proposed a novel approach that exploits the natural hardware offsets between low-power nodes to separate collisions from several transmitters using a single-antenna base station. Choir directly improves the throughput of dense urban LPWANs by decoding transmissions from multiple nodes simultaneously

with minimal coordination overhead. Specifically, signals from two transmitters are likely to experience a small frequency offset, due to a difference in the frequency of their oscillators. This results in the two chirps being slightly offset in frequency and thus being separable. This is a very promising technique, and is closely related to our proposed algorithm. The characteristic of Choir is that it relies on the frequency offset to separate and decode synchronized interfering transmissions. Further, Choir allows groups of LoRaWAN sensor nodes with correlated data to reach the base station, despite being individually beyond communication range.

Decoding Data from Collisions: Authors note that once the wireless channels and frequency offsets are estimated, decoding data is simple. Specifically, let us consider collisions of two transmitters synchronized in time whose data as well as preamble symbols collide. They first estimate the peak locations, i.e. frequency offsets, f_1 and f_2 averaged across each symbol of the colliding preamble. f_1 and f_2 correspond to the small hardware offsets. Then they repeat this process for the data symbols, where peak locations are given by $d_1 + f_1$ and $d_2 + f_2$, a sum of both the frequency offsets and the data (d_1, d_2). One can then subtract the known frequency offset from these values to obtain the data.

Mapping Symbols to Users within a Packet: the authors use both time and frequency offsets to map which symbols (i.e., chirps) correspond to which user within a frame along with the channels. Like hardware offsets, wireless channels are expected to remain consistent for a given client over a frame and vary between clients. For instance, in Fig. 3.10, they observe that peaks of the same user over two symbols are not only identical in frequency offset, but also in relative height. This means that channel magnitude and phase, after correcting for any phase offsets between symbols introduced by frequency offsets, can serve as a feature to identify users. This allows the authors to build a semi-supervised clustering model using the fractional part of peak location, channel magnitude, and phase.

Here are the limitations of Choir:

1. While Choir allows collisions from multiple transmitters to be decoded, its gains are bounded and limited when increasing the number of nodes, as the possibility of overlapping frequency offsets increases with collisions from a larger number of transmitters.

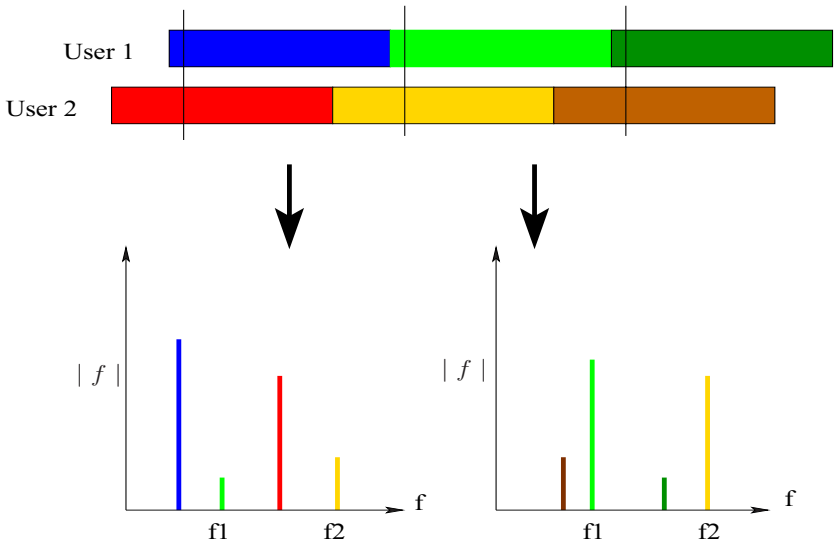


Figure 3.10: Inter-Symbol Interference: Spectrogram of two collided chirps, and the corresponding Fourier transform peaks.

2. If Choir fails to decode the synchronized colliding frames, the entire frames must be retransmitted as in the traditional LoRaWAN protocol.

3. Moreover, if a collision happens again between the same transmitters, frames remain undecodable because the transmitters do not change their frequency offsets.

In [95], the authors propose two algorithms to decode colliding signals: one algorithm requires the transmitters to be slightly desynchronized, and the other requires the transmitters to be perfectly synchronized. For the algorithm which is slightly desynchronized, the authors use the timing information to match the correct symbols to the correct end-devices. They show that their algorithms are able to significantly improve the overall throughput of LoRa. In the case of two completely synchronized signals, the authors propose a simple algorithm for this case. When two such frames collide, the algorithm stores the possible values for each symbol, and requests any of the transmitters to retransmit its frame. When one frame is retransmitted, the algorithm is able to decode it, and is able to deduce the values of the colliding frame of the other node too, by elimination. Thus, instead of having to retransmit two colliding frames, only one retransmission is required.

In [42], the authors extend their preliminary proposal of [95]. They proposed a col-

lision resolution technique that enables to decode two or more superposed LoRa signals. The proposed decoding algorithm exploits a slight desynchronization among superposed signals as well as the specificities of LoRa physical layer. They show that the decoding performance of their collision resolution technique can be further improved by making use of the CRC which is already available in each frame. Simulation results show that, compared to the conventional LoRaWAN protocol, the proposed CR-MAC protocol provides remarkable performance improvements, both in terms of system throughput and energy efficiency. In addition, the proposed protocol enables significant delay reductions.

Figure. 3.11 shows an example of the superposition of two slightly desynchronized signals. The preamble length is three symbols (2 up-chirps instead of 6, no sync word, and 1 down-chirp instead of 2.25), and SF7. The figure shows the signal of the first transmitter ED1 starting at t_0, the signal of the second transmitter ED2 starting at $t_0 + \delta$, and the superposed signal at the receiver. The data transmitted by ED1 and ED2 is (32, 32) and (96, 0) respectively.

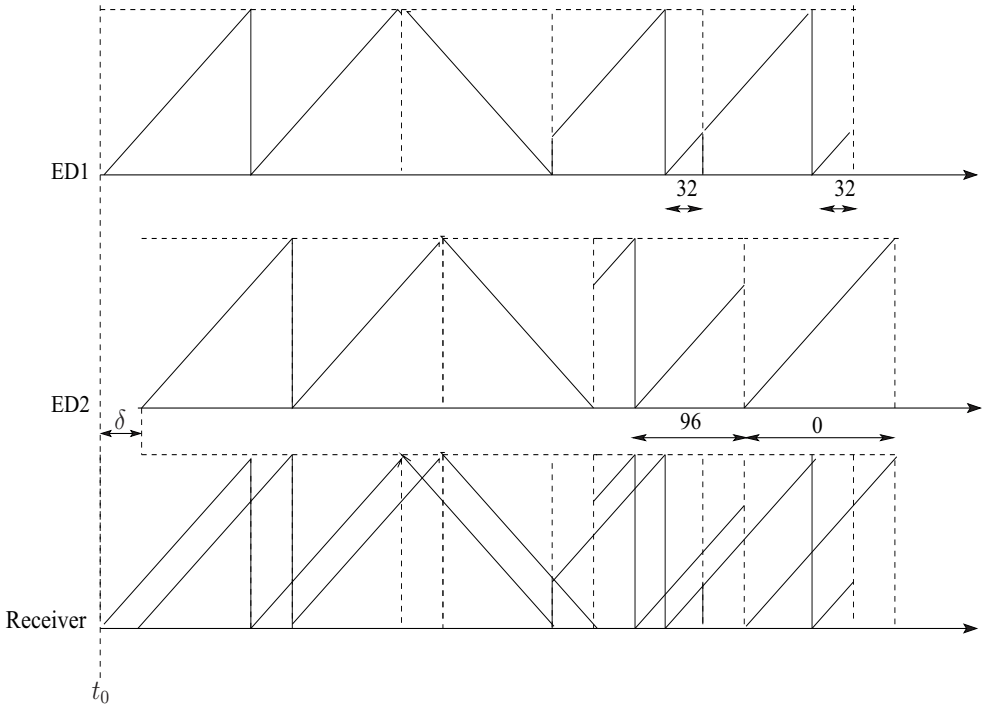


Figure 3.11: Superposition of two slightly desynchronized signals.

Proposed collision resolving MAC protocol: The proposed decoding algorithm requires transmissions to be slightly desynchronized, by less than one symbol, which is a rare event in LoRaWAN. Thus, authors designed a new MAC protocol called Collision Resolving-MAC (CR-MAC). The CR-MAC protocol works as follows: Each gateway sends periodic beacons on each SF. These beacons are sent simultaneously by all gateways, as in Class B of LoRaWAN. Upon receiving a beacon, each end-device starts S consecutive slots, whose duration is equal to the maximum frame transmission plus one symbol. To transmit a frame, an end-device has to wait for the beginning of a slot. It then draws a random number between 0 and $s = (SD/\delta) - 1$, and delays its transmission by $s \times \delta$ where SD is the duration of the sweep and called symbol duration.

Figure 3.12 depicts an example of the CR-MAC protocol with three beacons, and $S = 3$ slots after each beacon. At the beginning of each slot, there are $s = 4$ sub-slots, which correspond to possible starting times for the transmission of frames within each slot.

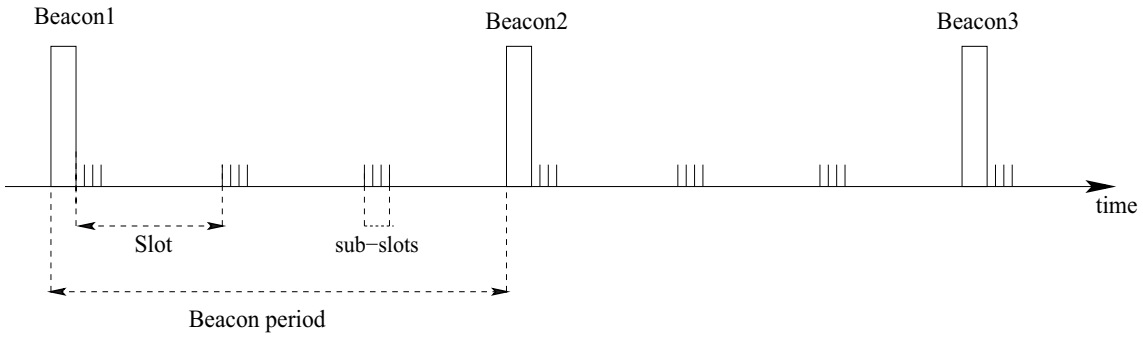


Figure 3.12: Proposed CR-MAC protocol.

Summary of the different protocols proposed for decoding synchronized collided LoRa signals

In Table 3.3, we present a summary recapitulating the different protocols proposed for decoding synchronized collided LoRa signals.

Table 3.3: Summary of the different protocols proposed for synchronized collided LoRa signals.

Papers	Advantages	Limitations
Paper [93]	• The concurrent transmission (CT) is a flooding protocol that considers synchronized packet collisions that happen when multiple relays perform immediate retransmissions at the same time. It is a multi-hop network. • CT is further improved by introducing timing offsets between the relaying packets.	• If the colliding signals are not decoded with the capture effect, they are considered lost and not retransmitted.
Paper [94]	• Choir proposed a novel approach that exploits the natural hardware offsets between low-power nodes to separate collisions from several transmitters. • Choir directly improves the throughput of dense urban LPWANs by decoding transmissions from multiple nodes simultaneously with minimal coordination overhead.	• While Choir allows collisions from multiple transmitters to be decoded, its gains are bounded and limited when increasing the number of nodes, as the possibility of overlapping frequency offsets increases with collisions from a larger number of transmitters. • If Choir fails to decode the synchronized colliding frames, the entire frames must be retransmitted as in the traditional LoRaWAN protocol. Furthermore, if a collision happens again between the same transmitters, frames remain indecodable because the transmitters do not change their frequency offsets.
Paper [95]	• Instead of having to retransmit two colliding frames, only one retransmission is required.	• The proposed algorithm considered the case of only two synchronized collided signals.

Chapter 4

State of the Art for Gateway Selection in Downlink Communication

Contents

The LoRaWAN architecture consists of end-devices connected to the network server through one or multiple gateways. These gateways relay messages from end-devices to the network server (i.e. uplink communication), and from the network server back to end-devices (i.e. downlink communication). Each gateway in LoRaWAN makes a single-hop star network of end-devices around it. Similarly, all gateways are connected to the network server. This makes each LoRaWAN network a star-of-stars topology. In contrast to traditional cellular networks, the end-devices are not related to a specific gateway in order to have access to the network. The same uplink data frame may be received and forwarded by more than one gateway: LoRa gateways simply forward frames from the end-devices to the network server after adding information about the quality of the reception. The network server is in charge of eliminating duplicate frames. The network server selects a single gateway for sending a reply (if any) in downlink communications [66].

In this chapter, we present some of the related work for LoRaWAN downlink communications. We also present related work in the context of the gateway selection by the network server.

4.1 Related work for downlink traffic in LoRaWAN

While the main use case for LoRaWAN networks is sensor data collection, downlink transmissions can be required for the acknowledgment of important traffic or to configure the sensors.

Recall that all end-devices start in Class A. Everytime an uplink frame is sent, a first receive window, Rx1, is opened by the end-device RECEIVE_DELAY1 seconds after the end of the transmission (with default value is 1 second) with the same SF used for the uplink frame and using the same channel. If a preamble is detected during Rx1, the radio receiver stays active for the downlink frame reception. If no downlink frame is received in Rx1, the end-device opens a second receive window, Rx2, after RECEIVE_DELAY2 seconds (default value is RECEIVE_DELAY1 + 1 seconds). Rx2 is opened on the 869.525 MHz, and the downlink transmitted should use the lowest available data rate (i.e. [SF=12, DR0]) with a duty cylce of 10%. Again, if a preamble is detected, the gateway must send the downlink frame exactly at the beginning of one of the two Rx windows, in order to allow the radio receiver of the end-device to detect the downlink preamble.

If the duty cycle is saturated for the sub-band corresponding to the downlink transmission channel, the gateway will not be able to forward the downlink frame. In this case, the end-device will perform the retransmission of its confirmed uplink frame, and possibly end up switching to a higher SF.

LoRaWAN uplink and downlink flow is illustrated in Fig. 4.1. In this figure, the end-device sends a confirmed uplink frame to the network server through three gateways. Then, the network server selects one gateway to forward the downlink frame back to the end-device.

Downlink traffic in LoRaWAN is recently arousing interest. Few studies have been done on this topic. Most of these studies focus on showing the negative impact of downlink traffic on the overall network capacity and performance, usually identifying the duty cycle as the main problem.

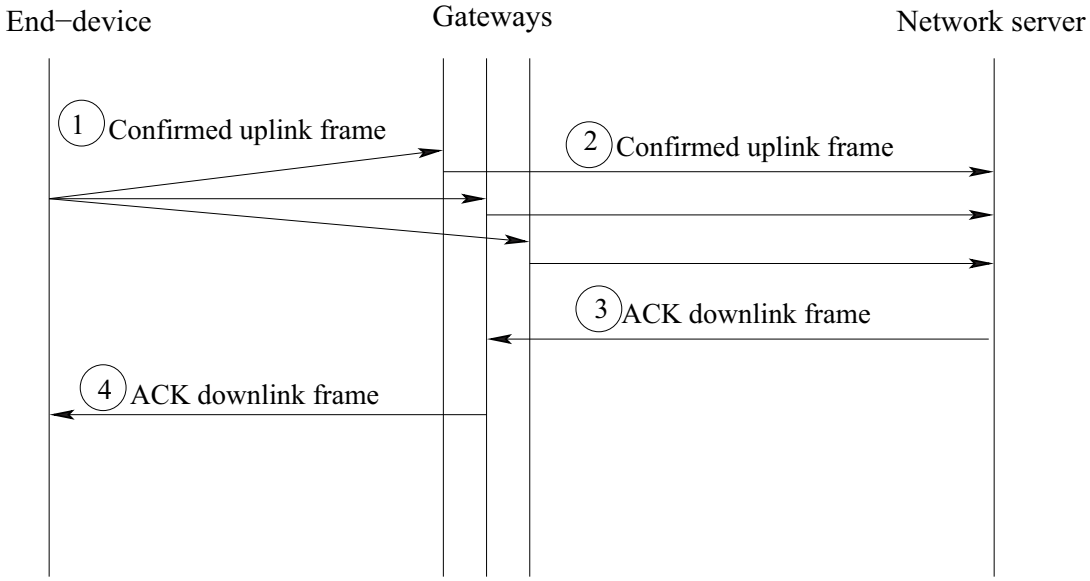


Figure 4.1: LoRaWAN uplink and downlink flow.

The authors in [96] showed that an incautious use of the downlink feature can bring a significant decrease in the performance of the network, especially for large scale deployments. Additionally, they presented some insights on how certain design choices for downlink communication in LoRaWAN weaken and hinder confirmed traffic usage. The incorrect reception of acknowledgment frames may trigger the retransmission of frames

that had actually been successfully delivered to the gateway. They observed that the interferences are the dominant cause of frame losses, but after a certain traffic load, the main limiting factor becomes the saturation of the receive paths (i.e., the communication channels) at the gateway. In order to mitigate the problem of the receiver saturation, they propose to increase the number of parallel receive paths at the gateway. However, they found that such receive paths need to be coordinated in order to avoid that an incoming signal is locked into multiple paths. Moreover, a uniform deployment of multiple gateways in the coverage area may reduce the distance to the end-devices, thus enabling the use of higher bitrates and, hence, a contraction of the frame transmission times that, in turn, yields a reduction of the collision probability and of the busy time of the receive paths. More generally, they suggest that a possible way to reduce the frame loss probability is to re-balance the SF distribution in the network, in order to maximize the probability of parallel reception. Their simulation results showed that the frame delivery ratio increases significantly when the gateway duty cycle restriction is disabled. In other words, they showed that the duty cycle restrictions at the gateway affect the system performance in presence of confirmed traffic. They also showed that the performance of a single LoRaWAN cell can significantly degrade when the fraction of end-devices that require confirmed traffic grows excessively.

The authors in [97] address the bidirectional traffic problem by first introducing LoRaWANSim simulator, an extension of LoRaSim that includes downlink frames and retransmissions. On one hand, some of the findings of the authors are relative to the aggressive data-rate back-off approach during retransmission recommended in the initial version (i.e., V1.0) of LoRaWAN specification, and hence they are currently less relevant. On the other hand, the authors identify the scalability problem that arises when a large number of end-devices request ACKs. From their simulation results, they found that as the percentage of uplink messages requiring an ACK increases, the network performance severely degrades. With 100% of frames requesting an ACK, the network can barely operate at 15% of its capacity in comparison to the scenario where no frames request an ACK. Clearly, an increase in the percentage of ACKs for larger network sizes will make the gateway running out of transmit opportunities often due to duty cycle limitations thereby failing to return a significant number of ACKs. This in turn increases the number of retransmissions, leading to a considerable increase in the energy consumption.

The authors in [98] characterized the performance of LoRaWAN, paying special attention to the effect of downlink communications on the network performance. Among the most important results of their measurements is the observation of the strong negative effect the downlink communications have on the performance of the uplink communications. Their experiments have shown that even the uplink transmissions encoded with different data rates and done in different frequency channels may get affected. The other notable results of their work include: 1) the observed effects of the end-device's SF on the distribution of the frames among the channels, and 2) the effect of the uplink data rate (i.e., SF) on the selection of the receive window in downlink.

The authors in [99] aim to assess the performance of LoRaWAN in terms of dependability of a LoRa network under a massive number of frame arrivals, part of which require to be acknowledged by the network server. To do so, they implemented an event-driven simulator in Matlab to emulate the MAC protocol defined by LoRaWAN, while abstracting the PHY implementation. The simulation results showed that the performance of the LoRaWAN network is severely impacted by the downlink traffic generated by feedback frames and by the fraction of confirmed traffic. On the other hand, if there is a relatively low fraction of high-priority users, increasing the number of transmission attempts yields a higher throughput. Finally, as expected, the performance analysis of the different SFs reveals that the lower the SF, the lower the failure probability of the frame delivery.

The authors in [100] explain the reasons of gateway congestion by highlighting the duty cycle limitation. They performed a downlink traffic analysis using the ns-3 network simulator. A multi-gateway architecture is proposed and improvements are evaluated in terms of lower duty cycle saturation and better distribution of the workload through the gateways. Taking into consideration the gateway selection algorithm, in their implementation, the server simply tries to schedule an ACK in the first available gateway among the pertinent gateways. Using the LoRaWAN ns-3 module, a scalability analysis of LoRaWAN shows the detrimental impact of downstream traffic on the delivery ratio of confirmed upstream traffic. The analysis shows that increasing gateway density can reduce but not eliminate the effect of downstream traffic, as stringent duty cycle requirements for gateways continue to limit downstream opportunities.

The authors in [101] provided an overview of LoRaWAN capabilities and limitations. This overview explains why a LoRaWAN deployment must be carefully dimensioned to

meet the requirements of each use case. Thus, the combination of the number of end-devices, the selected SFs and the number of channels determine if the LoRaWAN ALOHA-based access and the maximum duty cycle regulation fit each use case. For instance, it was observed that deterministic monitoring, such as industrial automation, critical infrastructure monitoring and actuation, require real time operation, and therefore cannot be guaranteed with current LoRaWAN state of the art. On the other hand, the authors stated that the reliability in LoRaWAN is achieved through the acknowledgment of frames in the downlink. However, the capacity of the network is reduced not only due to transmissions in the downlink, but also due to the off-period time following those transmissions (gateways must be compliant with the duty cycle regulation). The duty cycle has a significant impact on the capacity of the network. Therefore, the design of the network and the applications that run on it must minimize the number of acknowledged frames to avoid a capacity drain.

The authors in [102] studied the usage of LoRa in indoor environments. Measurements were conducted in the main campus of the University of Oulu, Finland. Results indicate that with the largest SF of 12 and a transmit power of 14 dBm, the whole campus area can be covered by a single base station. The average measured packet success delivery ratio for this case was 96.7%, even with neither acknowledgements nor retransmissions. The campus was covered also with lower SFs and a transmit power of 2 dBm, but considerably more packets were lost.

The authors in [3] conducted a series of experiments to verify some features of LoRa technology. Their results show that LoRa is capable of communicating over a range of 10km under line-of-sight environments. However, under non-line-of-sight environments, LoRa performance is severely affected by obstructions such as buildings and vegetations. Moreover, they showed that a LoRa gateway supports up to 6,000 nodes with a PRR requirement of >70%. They also explored the impact of each LoRa transmission parameter and proposed an algorithm to determine optimal settings in terms of coverage and power consumption under non-line-of-sight environments. Finally, they showed that downlink traffic and especially ACK packets, could block a significant amount of uplink traffic since in order to transmit an ACK packet, a gateway has to switch one of its two radio chains from RX mode to TX mode [103]. As stated in [103], the SX1301 digital baseband chip has two TX/RX interfaces. So when the device has to transmit an ACK packet, it has to

switch one of its interfaces to the TX mode. Any LoRa packet may be transmitted on any of the two radios. Only a single packet may be transmitted at any given time. Transmit operation interrupts all current reception operations. This change in modes would disable all uplink traffics transmitting in the channels served by that particular radio chain during the ACK duration.

The authors in [84] analysed the performance and network capacity of LoRaWAN through simulations with 100-5000 devices, using the three default channels and data rates 0 to 5. They estimated both the Packet Error Rate (PER) and the Packet Loss Rate (PLR). They explained the limitations such as the duty cycle restrictions and the recommended behavior for re-transmissions. They showed that the duration of the transmission of an acknowledgement frame can be more than 1 second, and thus the authors claim that the probability of a repeated collision is high. The authors also drew the conclusion that one solution to these problems is to increase the density of gateways within the network to help offload the otherwise very busy gateways.

As a conclusion, we can see that all the results of the aforementioned papers have shown that the performance of LoRa is severely impacted if the number of end-devices that require confirmed data (i.e., ACK frames) grows.

4.2 Related work for the selection of gateway

In LoRaWAN uplink communications, the end-device sends its frame in broadcast, and all the gateways that receive the frame forward it to the network server. Furthermore, in LoRaWAN downlink communications, it is up to the network server to choose only one gateway from all the available ones (which are associated to the end-device) in order to reply to this end-device with a downlink ACK frame as shown in Fig. 4.2. In this figure, the end-device sends an uplink frame which is received by two gateways: Gateway 1 and Gateway 2. Then both gateways forward this frame to the network server. The latter has to choose which gateway to select to send an ACK back to the end-device. However, LoRaWAN specification does not propose an algorithm for this selection.

A variety of approaches exists in the literature regarding the load balancing in heterogeneous networks ([104], [105], [106], [107], [108]), where the network infrastructure is supported by heterogeneous elements consisting of Macro Base Stations (MBSs) which

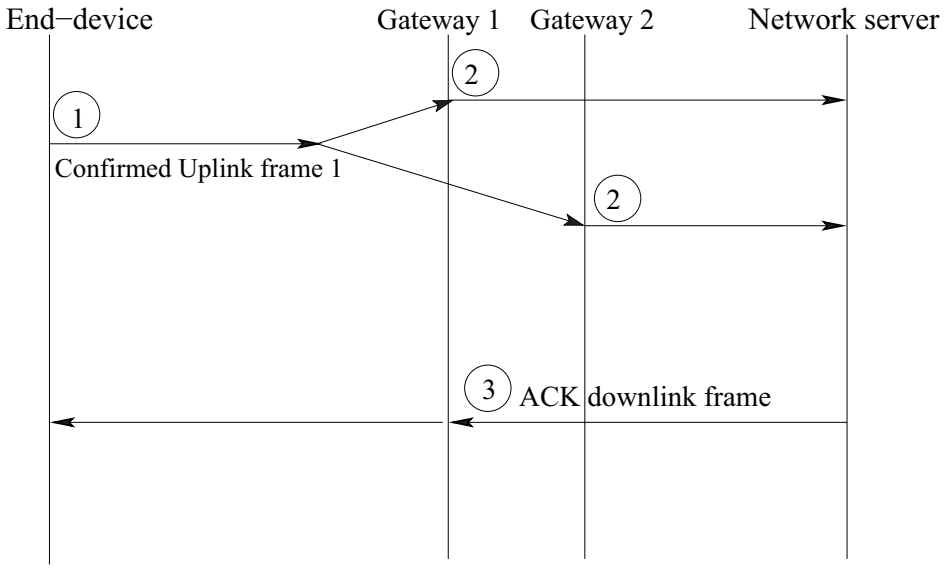


Figure 4.2: LoRaWAN gateway selection in downlink communications.

provide a wide area coverage, and Small Base Stations (SBSs) which aim to cover high traffic hotspots.

In [104] and [105], the authors proposed clustering techniques for optimizing the load balancing in heterogeneous networks. More precisely, in [104], the authors developed a new approach to the modeling and analysis of heterogeneous cellular networks (HetNets). Their proposed approach accurately incorporates coupling across the locations of users and base stations, which exists due to the deployment of SBSs at the places of high user density (referred to as user hotspots). Modeling the locations of the geographical centers of user hotspots as a homogeneous Poisson Point Process (PPP), they assumed that the users and SBSs are clustered around each user hotspot center with two different distributions. The MBS locations are modeled by an independent PPP. This model is consistent with the user and SBS configurations considered by 3GPP [109]. Using this model, the authors studied the performance of a typical user in terms of coverage probability and throughput for two association policies: i) Policy 1, under which a typical user is served by the open-access base station that provides maximum averaged received power, and ii) Policy 2, under which the typical user is served by the small cell tier if the maximum averaged received power from the open-access SBSs is above a certain power threshold; and macro tier otherwise.

A key intermediate step in their analysis is the derivation of distance distributions from a typical user to the open-access and closed-access interfering SBSs. Their analysis shows that as the number of SBSs reusing the same resource block increases, the probability of coverage decreases and the throughput increases. Therefore, in contrast to the usual assumption of orthogonal channels, it is reasonable to allocate the same resource block to multiple SBSs in a given cluster as long as the probability of coverage remains acceptable. This approach to HetNet modeling and analysis significantly generalizes the state-of-the-art approaches that are based on modeling the locations of base stations and users by independent PPPs.

In [105], a joint user association (UA) scheme with a joint coordinated multi-cell processing (JP-CoMP) using a hybrid self-organizing network (SON) is proposed for a practical clustered heterogeneous cellular network (cHCN) to maximize the network-wide proportional fairness among users. The cell range expansion and the enhanced intercell interference coordination have been considered as key items in the long-term evolution-advanced to offload macrocell users to small-cell base stations (SBSs). However, in a cHCN where SBSs are not distributed at random but are clustered instead, the coverage of inner SBSs in a small-cell cluster would be hardly expanded and an increased bias may result in poor link quality as well as high load in outer SBSs. Thus, the load-balancing capability becomes lower than expected in a cHCN. In order to cope with such a problem, a network architecture and protocol for the cHCN is suggested by the authors, and a feasible suboptimal iterative algorithm for determining the joint UA solution of the proposed hybrid SON is provided. It is shown that the proposed hybrid SON scheme with the proposed joint UA solution is very effective in handling the load balancing in a practical cHCN, not only improving the performance of the inner sBS users by reducing the inter-cell interference, especially for intratier offloaded users, but also enabling more aggressive intertier offloading by effectively improving the link quality of cluster edge users without causing an unnecessary resource waste.

In [106], the authors investigated downlink multi-antenna HetNets with flexible cell selection, and shown that simple selection bias-based cell selection criterion closely approximates more complex selection rules to maximize the mean Signal-to-Interference-plus-Noise Ratio (SINR). Under this simple cell selection rule, they derived exact expressions for coverage probability and rate achievable by a typical user. An approximation of the

coverage optimal cell selection bias for each tier is also derived in closed form. Due to this connection, there is a natural expansion of coverage regions of small cells whenever small cells can use multi-antenna transmission for range expansion, e.g., by using beamforming. This leads to a natural balancing of load across tiers, which reduces the additional artificial cell selection bias needed to offload sufficient traffic to small cells.

In [107], the authors proposed a load balancing solution for a two-tier heterogeneous networks based on stochastic geometry. Their algorithm performs a Cell Range Expansion technique (CRE) biasing to achieve an optimal SBS density regarding network energy efficiency. The authors have analyzed the energy efficiency for a two-tier HetNets consisting of MBSs and small cell base stations SBSs called Pico cell Base Stations (PBSs) by means of stochastic geometry theory, where CRE implemented on PBS and Almost Blank Subframe (ABS) based on an Enhanced Inter-cell Interference Coordination (eICIC) scheme is adopted by MBS for downlink interference mitigation to PBS CRE User Equipments. They first derived the closed-form expression of the network energy efficiency. Then, a linear search algorithm is adopted to optimize the small pico CRE bias and PBS density, respectively. Finally, a heuristic based algorithm is proposed to optimize the small pico CRE bias and PBS density jointly to achieve the network energy efficiency maximization. Simulation results showed the accuracy and the effectiveness of their proposed optimization algorithms for the network energy efficiency optimization with reduced complexity.

In [108], the authors proposed a load balancing scheme for downlink communications based on machine learning techniques to enhance the capabilities of an urban IoT network operating under the LoRaWAN standard. In homogeneous wireless networks, a device is associated with the base station (BS) providing the strongest signal. This association method is not efficient for heterogeneous networks in terms of network capacity where the device association methods based on signal metrics may lead to a major load imbalance. Hence, a load balancing method for heterogeneous networks has been proposed. Their model predicted a device-BS association by avoiding the signal-based measurements. They propose a decision-making model in order to achieve a load balance and, consequently, improve the network capabilities in terms of packet delivery ratio (PDR) and energy cost of data (ECD) delivery. Additionally, a supervised classifier is applied in order to accomplish a biasing scheme by observing metrics that are not directly related to signal strength. In this way, their model learns from data to predict a device-BS association

(i.e., the selection of a BS that should forward the downlink frames to the end-device) without considering signal-based measurements. They extracted several variables (such as frequency, data rate, etc.) from data and waived the RSSI metric. Additionally, their method employed a Markov Decision Process (MDP) to determine whether a BS needs to be balanced or not. Their simulation results showed that their MDP-based decision-making model has better results (in terms of PDR and ECD delivery) when the classifier predictions are considered and compared to an unbalanced network. The limitation of the proposed model is the time delay caused by the decision process that may be unacceptable for several WAN applications. The time complexity analysis for the implementation of the authors's model must be considered especially where there is a large number of end devices. In this thesis, we consider the RSSI with the load balance together in the gateway selection for downlink communication. Subsequently, we study the impact of the combination of these two metrics on LoRaWAN performance in different scenarios of gateway deployment. Ultimately, we show the effect of these scenarios on the overall network performance.

Paper [110] was made in the context of the gateway selection for downlink communications in LoRaWAN. As stated in [110], one possible selection algorithm for the gateway is based on the Signal to Noise Ratio (SNR) which is a good way to estimate the wireless link quality assuming the same bidirectional transmission conditions. In this SNR-based algorithm, the selection of the gateway only depends on the information carried by the duplicates received from the targeted end-device. In the de-duplication phase, the server records the gateway that forwarded the frame with the best SNR value. The duty cycle of a gateway which serves a large number of end-devices may quickly reach saturation and the gateway would then miss the RX windows. Moreover, if the selected gateway is consistently receiving a large amount of uplink traffic, the number of missed uplink frames during the downlink transmissions would most probably also be significant. Thus, with such a naive selection algorithm, the gateway load is not taken into consideration and it mechanically introduces notable frame loss. In order to better balance and spread the downlink traffic all along the network, the gateway selection algorithm should also take into account the effective ability of the gateway to forward the downlink, by checking its duty cycle saturation and its already scheduled downlink frames, in order to avoid overlaps. This multi criteria gateway choice algorithm attempts to strike a balance between trying to use the best radio link and avoiding frame losses. Simulation results in [110]

show that, despite the use of more gateways, frame loss for higher proportion of confirmed messages is still important. This is a direct consequence of the behavior of the gateway selection algorithm which does not attempt to spread downlink traffic in any way by always and only selecting the gateway with best reception conditions. Hence, in order to better balance the load of the downlink traffic, the balanced gateway selection algorithm has been tested by the authors. They showed the improvements introduced by the balanced algorithm compared to the SNR-based algorithm with their respective frames loss. Their results show that, in a multi-gateway implementation, the balanced algorithm decreases the frame loss by 25% compared to the SNR-based algorithm and by 66% compared to the single-gateway architecture, producing losses that never exceed 20% of the total traffic. The main reason of this improvement gives a better distribution of the downlink traffic. With this algorithm, the ACK loss, which was the main cause of frame loss for high percentage of confirmed frames, is strongly decreased. Unlike before, a downlink frame is not scheduled on the gateway with the best SNR if its duty cycle is saturated for the given sub-bands or if it overlaps with other already scheduled downlink frames. The algorithm tries the second best gateway in terms of SNR, and so on. Obviously, ACKs can still be lost due to other factors, especially if transmitted by a gateway that presents a lower SNR value. For downlink traffic loads, frame loss are decreased thanks to the balanced gateway selection algorithm, enabling the deployment of applications that require such confirmed data communication. It is worth noting that here the authors did not take into consideration the location of the gateways with respect to the end-devices. In this thesis, we consider both the gateway load and the RSSI metrics in the gateway selection for downlink communication. Moreover, we consider different scenarios of gateway deployment, and we study their impact on the throughput, the delay, and the energy consumption in LoRaWAN.

In [111], the authors showed the implementation result of the LoRa network server on the OpenStack platform as shown in Fig. 4.3. They classified the operations of the LoRa network server into four blocks, gateway agent, application server agent, data processing, and control, with considering system scalability and maintainability.

As mentioned previously, when the network server has a frame to send to an end-device, it has to select a single gateway to relay this frame. However, LoRaWAN neither specifies how to select the gateway, nor provides any information or recommendation on

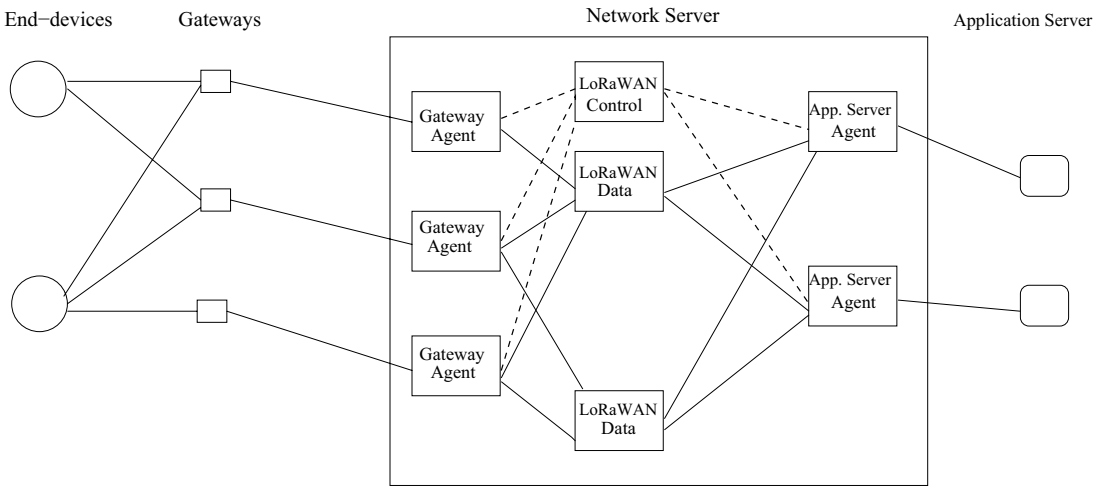


Figure 4.3: Structure of network server implemented on the Openstack platform.

this selection. Researchers have identified that the gateway selection is an important issue in LoRaWAN. Both [66] and [111] state that the network server needs to select the best gateway among all candidates when replying in donwlink communications to each end-device. As we have shown, few works [108, 110] have provided information on this selection, and studied the impact of this selection on LoRaWAN throughput.

Part III

CONTRIBUTION I

Chapter 5

Efficient Decoding of Synchronized Colliding LoRa Signals

Contents

LoRa gateways are able to decode superposed LoRa signals as long as they are sent on different channels or on different SFs. When several signals are received on the same channel and with the same SF, a difference of received power might cause the strongest signal to be captured by the receiver. This is called the capture effect [55], [87]. When several signals have a similar receive power, a collision occurs and all signals are considered lost [71]. In this chapter, we focus on decoding superposed LoRa signals received on the same channel, with the same SF, with similar receive power, in the case where the signals are fully synchronized.

In this chapter, we present our contributions for the physical and MAC layers. First, we design a slotted version of the conventional LoRaWAN and we call it slotted-backoff LoRaWAN. It is used as a comparison basis for our MAC protocol. Second, we discuss about the superposition between fully synchronized LoRa signals in the physical layer. Third, we propose a new MAC protocol that makes use of this feature of the physical layer.

5.1 Slotted-backoff LoRaWAN

Here we present our slotted-backoff LoRaWAN. This protocol is slotted, that is, each transmission starts at the beginning of a slot after a random backoff. Synchronization is achieved thanks to periodic beacons sent by the gateway. If the end-device (referred to as ED) senses the medium and finds it idle, it transmits its frame. Otherwise, it waits for a new slot and tries again its transmission with a new backoff. Thus, the probability of collision is reduced because end-devices sense the medium before trying to use it. More precisely, once an end-device has a frame to transmit, it randomly chooses a backoff delay BD from the range $[0; 2^{BE} - 1] \times min_{SlotDuration}$, where BE is the backoff exponent, and $min_{SlotDuration}$ is the minimum duration needed to detect a preamble. Once an end-device ED^x has waited for the chosen number of backoff periods, it performs carrier sensing for a duration equal to $carrier_{sensing}$. Note that $carrier_{sensing}$ is much smaller than the duration of a slot $slot_{duration}$. If the medium is idle (i.e. if no other end-device ED^y, with $y \neq x$, is transmitting on the same channel and SF during the transmission of ED^x), ED^x begins the transmission of its frame until it is entirely transmitted. If ED^x attempts to transmit during the middle of the transmission of ED^y, ED^x backs off to avoid collisions.

If ED^x attempts to transmit its frame at exactly the same time as ED^y, a collision occurs and the colliding frames are retransmitted after the sleep period and after a random delay from the second receive window as in [5]. Note that the time is divided into slots of one frame duration, plus the start time of the second receive window, plus the sleep period plus the random delay chosen by the ED. The interval between the start of two successive beacons is called the beacon period.

Figure 5.1 shows our slotted-backoff LoRaWAN. ED_i^x corresponds to frame number i of end-device x. Here, ED^1 sends a frame during Slot1. Then, ED^2 attempts to send a frame, but finds that ED^1 is already sending, hence ED^2 backs off, and the first frame of ED^1 (i.e ED_1^1) is successfully sent. Afterwards, ED^1 waits for the minimum sleep period (to respect the duty cycle limitation) plus a backoff before sending a new frame ED_2^1 as shown in Slot4. In Slot4, ED^3 attempts to send a frame (i.e ED_1^3) at the same time as ED_2^1. In this case, a collision occurs between ED_2^1 and ED_1^3. Therefore, each colliding frame has to be retransmitted by its end-device while respecting the duty cycle limitation. As stated previously, the retransmission has to be done after a random delay between 1 and 3 seconds from the second receive window (i.e. Rx2).

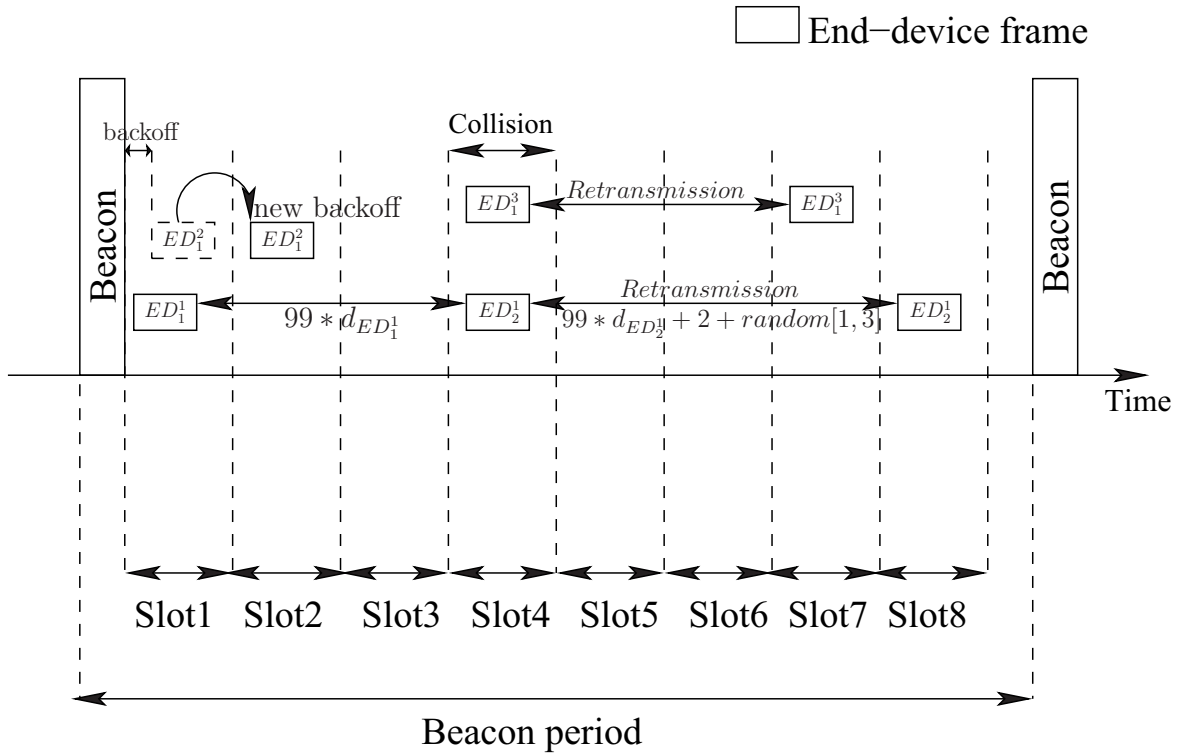


Figure 5.1: Slotted-backoff LoRaWAN.

5.2 Dealing with superposed frames at the physical layer

In LoRaWAN, when a collision occurs, the colliding end-devices have to retransmit their entire colliding frames, which leads to increase the delay of these frames, as well as the energy consumption of the end-devices. In parallel, the overall network throughput decreases. For these reasons, we propose the new mechanism described in this section.

We present our method to deal with the superposition between synchronized signals in LoRa physical layer. Then, we design an algorithm to extract information from these superposed signals.

We consider the superposition of signals from transmitters that are fully synchronized. We consider the same size for all the transmitted frames. Recall that papers [93] and [94] have ensured the feasibility of the synchronization among LoRa signals, by implementing a real LoRa system where transmissions were synchronized in time.

Figure 5.2 shows an example of the reception of three fully synchronized signals under SF7 without preambles for clarity reasons. The frame transmitted by ED1 is $f_1 = (64, 32, 32)$, the frame transmitted by ED2 is $f_2 = (96, 0, 32)$, and the frame transmitted by ED3 is $f_3 = (96, 64, 32)$. The signals collide at the receiver. However, the receiver is able to store all the superposed symbols at each symbol duration. Thus, the receiver extracts symbols {64, 96} during the first symbol duration, symbols {32, 0, 64} during the second symbol duration, and symbol {32} during the third symbol duration. However, the receiver is not able to determine to which frame each symbol belongs.

5.3 Proposed MAC protocol

In this section, we present our MAC protocol used to decode synchronized colliding signals, based on the physical protocol described in subsection 5.2.

5.3.1 Description of our MAC protocol

Our proposed protocol relies on the gateway guessing a frame and sending it to end-devices, and end-devices sending bitmaps in order to determine the correct symbols of the frames, instead of retransmitting the whole frames. This is only when synchronized signals collide. Also, we use an algorithm to recover the colliding signals, named bitmap processing algorithm (as shown in Section 5.3.3).

When a collision occurs between frames[1], the gateway stores all superposed symbols at each symbol duration using the technique described in subsection 5.2. Then, the gateway sends an arbitrary frame built from these symbols. The gateway frame contains, in addition to the symbols, the order of the colliding end-devices using an identifier on one symbol per end-device. The gateway waits for bitmaps from the end-devices in order to decode the colliding frames, where each bit corresponds to the symbol chosen in the gateway frame. For instance, if the j^{th} bit of the bitmap of ED i is 1, it means that the j^{th} symbol in the gateway frame was also the j^{th} symbol of the frame sent by the ED. And if the j^{th} bit of the bitmap of ED i is 0, it means that the j^{th} symbol of the gateway frame

[1]Recall that a collision occurs when several signals are received on the same channel, with the same SF, and with similar receive power.

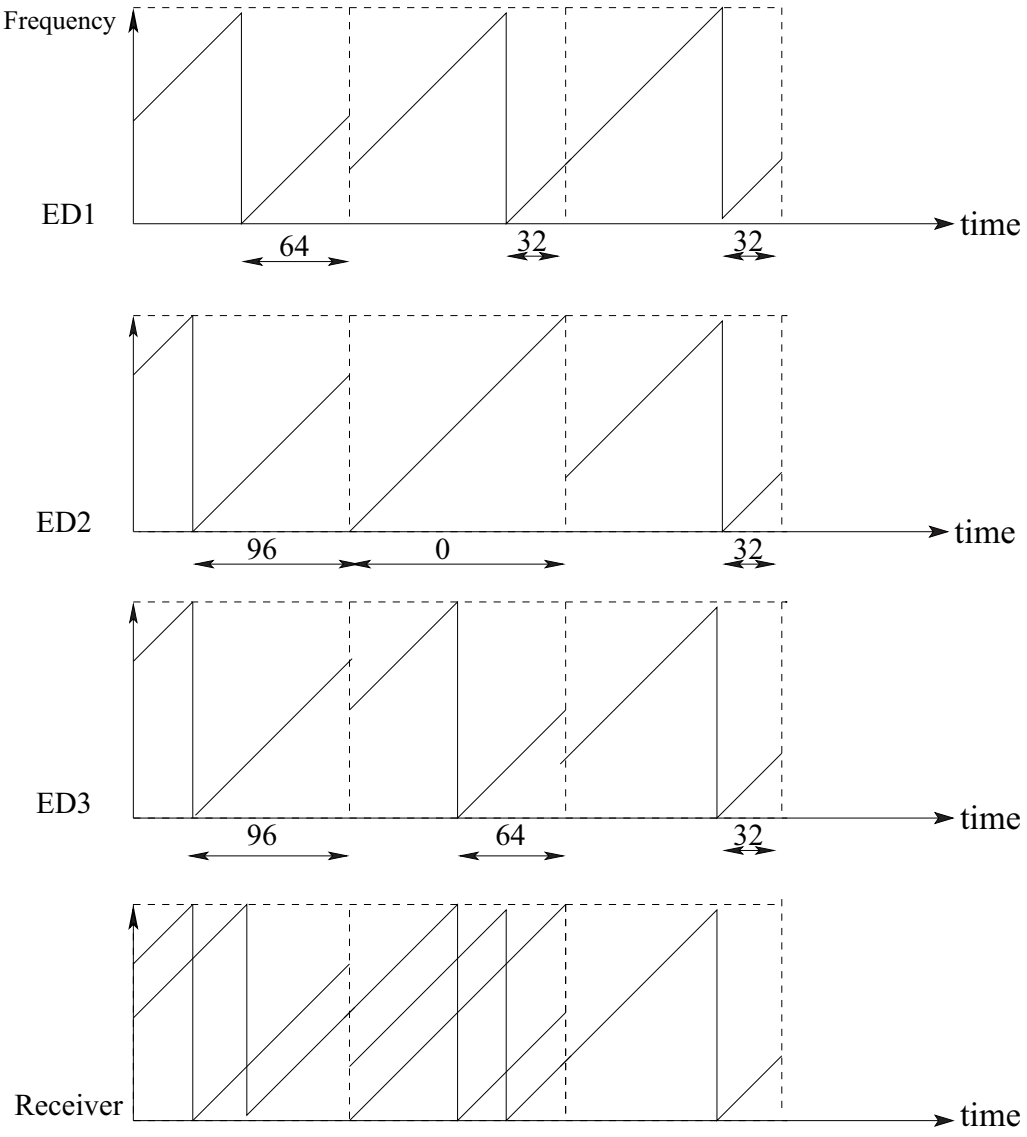


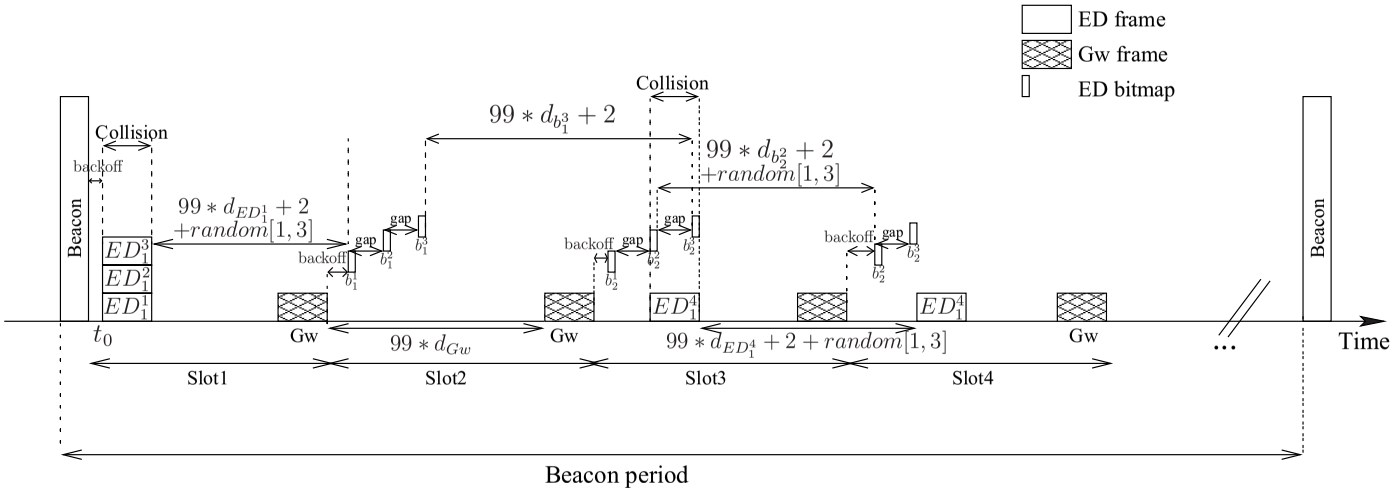
Figure 5.2: Superposition of three fully synchronized signals.

was not the j^{th} symbol of the frame sent by the ED. The gateway sends an ACK to each ED if its bitmap has been received. In case of collision between bitmaps of different EDs, no ACK is sent to the ED, which retransmits a bitmap after a sleep period from the start

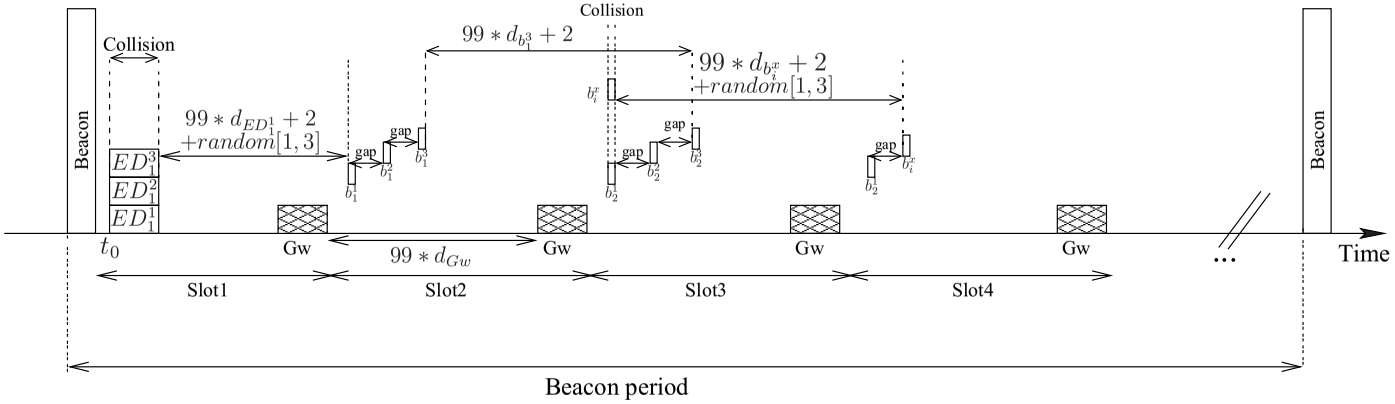
time of Rx2 plus a random delay between 1 and 3 seconds (same ACK_TIMEOUT period as in the conventional LoRaWAN protocol). Otherwise, if the bitmaps are successfully received by the gateway, but the ACK is not received by the corresponding ED because the gateway was busy sending another ACK, the ED has also to retransmit its bitmap again to the gateway until it receives an ACK. Finally, as long as the frames sent by the end-devices are not yet decoded, the gateway keeps guessing and sending new frames, and the end-devices keep replying by sending new bitmaps.

Figure 5.3 illustrates the new MAC protocol implementing our proposed algorithm (Fig. 5.3(a) is depicted for four end-devices, and Fig. 5.3(b) is depicted for three end-devices). In our MAC protocol, beacons are sent by the gateway to synchronize the communications. Note that in each slot, the end-devices transmit at the begining of this slot after a random backoff, while the gateway transmits after the end of all EDs transmissions and at the end of this slot. As an example for Slot1 of Fig. 5.3(a),(b), we assume that initially (i.e. at t_0) the frame transmissions on the same slot are fully synchronized. By referring to Slot1 in the example of Fig. 5.3(a), there is a random backoff before each end-device's transmission in order to decrease the probability of collisions between frames and bitmaps. Here, the three end-devices ED1, ED2 and ED3 have chosen the same backoff, thus their frames collide with each other. Then, the gateway (referred to as Gw) sends a frame built from the superposed signals. Then, in Slot2 in the example of Fig. 5.3(a), each of the three EDs replies to the gateway frame by sending a bitmap. It is also necessary to add a random backoff on the sending of bitmaps (as shown in Slot2 and Slot3 in the example of Fig. 5.3(a)), because collisions between frames and bitmaps, or between bitmaps themselves are possible. Afterwards, the gateway sends another frame as long as there are frames that are not yet decoded.

We distinguish three types of collisions in our proposed MAC algorithm: a *frame collision* which is a collision that occurs between frames only and which leads to generate bitmaps (as shown in Slot1 of Fig. 5.3(a)), a *bitmap collision* which is a collision that occurs between bitmaps only (as shown in Slot3 of Fig. 5.3(b)), and a *mixed collision* which is a collision that occurs between bitmaps and frames at the same time (as shown in Slot3 of Fig. 5.3(a)).



(a) An illustration of *frame collision at Slot1* and *mixed collision at Slot3*.



(b) An illustration of *frame collision at Slot1* and *bitmap collision at Slot3*.

Figure 5.3: New MAC protocol for the superposition of multiple synchronized signals.

5.3.2 Timing computation

We develop a timing computation model for the transmission process to further describe our MAC protocol. Specifically, we consider separately the first and the subsequent transmission attempts. Note that if there is no collision, the end-devices receive ACKs from the gateway as in the conventional LoRaWAN.

The first frame transmission: The first transmission attempt of the end-devices frames is made on Slot1 in the example of Fig. 5.3(a), where three end-devices ED1, ED2 and ED3 are initially fully synchronized, and sent their uplink frames at the same initial start time t_0. This causes a collision at the gateway. Hence, the latter stores the superposed symbols, and sends a frame composed from these superposed symbols (See Section 5.3.3 for an example of the frame sent by the gateway).
The start time of the first gateway frame ($t_{0_{Gw}}$) is equal to the initial start time of the EDs frames (t_0) plus the duration d_{ED} (i.e time on air) of the EDs frames plus an unused time δ (between the last frame of an ED and the frame of the Gw) as follows:

$$t_{0_{Gw}} = t_0 + d_{ED} + \delta \tag{5.1}$$

The first bitmap transmission: The first transmitted bitmap sent by an ED is done after a sleep period plus 2 seconds (which is the start time of Rx2) plus a random delay between 1 and 3 seconds. Afterwards, each bitmap of a given end-device is separated from the bitmap of the previous and the next end-devices by a small amount of time called guard interval (i.e. *gap*). This guard interval ensures that the bitmap of an end-device does not collide with the bitmap of another end-device, even when considering clock drifts.
By referring to Fig. 5.3(a), each end-device in Slot2 sends a bitmap in reply to the Gw frame. The bitmaps b_1^1, b_1^2, and b_1^3 are sent by ED1, ED2 and ED3 respectively, after 99 times the duration of the frame transmission of ED1, ED2 and ED3 (i.e. the off period). Moreover, each colliding ED should wait, in addition to the sleep period, for 2 seconds (which is the start time of Rx2) plus a random delay ACK_TIMEOUT between 1 and 3 seconds. Furthermore, for each end-device x, its bitmap b_i^x is delayed to avoid a collision with bitmaps $b_i^{(x-1)}$ and $b_i^{(x+1)}$. In other words, for an end-device x, the start time $t_{b_i^x}$ of

its bitmap number i should respect the following rule:

$$(t_{b_i^{(x-1)}} + d_{b_i^{(x-1)}}) \times (1 + \Delta_{max}) \leq t_{b_i^x} \times (1 - \Delta_{max}) \tag{5.2}$$

where $d_{b_i^{(x-1)}}$ is the time on air of the the previous bitmap, and Δ_{max} is the maximum drift.

Hence, the start time $t_{b_i^x}$ of a bitmap b_i^x is given by the following equation:

$$t_{b_i^x} = t_{b_i^{(x-1)}} + d_{b_i^{(x-1)}} + gap \tag{5.3}$$

with

$$t_{b_i^{(x-1)}} = t_{Slot} + (x-2) \times d_{b_i^{(x-1)}} + (x-2) \times gap \tag{5.4}$$

and t_{Slot} is the start time of the current slot.

Moreover, the duration of a slot is given by the following:

$$d_{Slot} = \max(d_{ED} + d_{Gw}, d_{b_i^x} \times x + gap \times (x-1) + d_{Gw}) \tag{5.5}$$

where d_{ED} is the duration of the end-device frame, d_{Gw} is the duration of the gateway frame, and x is the maximum number of EDs in collision.

In relation to the slot duration, we set the start time of the last slot as follows:

$$t_{Slot_{nmaxslots}} = d_{Slot} \times (nmaxslots - 1) \tag{5.6}$$

where $nmaxslots$ is the maximum number of slots.

The subsequent bitmap transmissions: The start time of a bitmap for the subsequent transmissions is given by equation (5.7). It states that the end-device should respect the duty cycle limitation, and should wait for the gateway frame before sending its bitmap b_i^x with $i > 1$.

$$t_{b_i^x} = \max(t_{b_{(i-1)}^x} + 100 \times d_{b_{(i-1)}^x}, t_{Gw} + 100 \times d_{Gw}) \tag{5.7}$$

Bitmap collision: A bitmap collision may occur as shown in Slot3 of Fig. 5.3 (b), where

the bitmap b_2^1 of ED1 collides with a bitmap b_i^x of an ED x. In this case, each colliding ED (i.e. ED1 and ED x) retransmits its colliding bitmap (i.e. b_2^1 and b_i^x respectively) as shown in Slot4 of Fig. 5.3(b). The maximum number of bitmap retransmissions attempts is set to 8. Thus, ED x retransmits its colliding bitmap number i while respecting the duty cycle of 1% as follows:

$$t_{b_i^x} = t_{b_{(i-1)}^x} + 100 \times d_{b_i^x} + 2 + random[1,3] \tag{5.8}$$

Mixed collision: In the mixed collision represented in Slot3 of Fig. 5.3(a), the frame of ED4 collides with the bitmaps b_2^2 and b_2^3 from ED2 and ED3 respectively. The gateway receives the bitmap b_2^1 of ED1 and decodes it, while bitmaps b_2^2 and b_2^3 are not received. This leads ED2 and ED3 to retransmit their colliding bitmaps in Slot4. We set the maximum number of retransmissions to 8 (as in the conventional LoRaWAN). Hence, after 8 attempts, the colliding bitmap is lost. Also, ED4 retransmits its colliding frame number f as shown in Slot5 while respecting the duty cycle of 1% as follows:

$$t_{ED_f^x} = t_{ED_{(f-1)}^x} + 100 \times d_{ED} + 2 + random[1,3] \tag{5.9}$$

Note that the frame of a given end-device may collide with all the bitmaps of the other y end-devices that are sent on the same slot if:

$$d_{ED} \geq y \times d_{b_i^y} + (y-1) \times gap \tag{5.10}$$

5.3.3 Bitmap processing algorithm

Algorithm 1 presents the bitmap processing algorithm used to decode fully synchronized colliding signals for x transmitters (i.e. end-devices), with $x \geq 2$. This algorithm determines the parts of the frames of EDs from the bitmaps received from these EDs.
When a collision occurs for x EDs, the gateway can extract the colliding symbols, but is not able to determine to which frame each symbol belongs. Hence, the gateway considers that all the end-devices frames contain missing symbols, represented by * in Algorithm 1. Then, the gateway guesses and sends a frame built from the superposed symbols. At this step, each ED x replies to the gateway frame by sending a bitmap b_i^x (which corresponds

to the i^{th} bitmap transmitted). For each *bit* at position j in b_i^x, the algorithm checks the value of *bit*. If *bit* is equal to 1, then the algorithm replaces * in the frame of ED x with the current symbol of position j of the frame guessed by the gateway. On the other hand, if *bit* is equal 0, the algorithm checks if the number of superposed symbols at the current position j is equal to 2. If it is the case, then the algorithm replaces the * with the other current symbol at position j of the superposed symbols. In addition, the algorithm verifies if at position j, all the symbols of the EDs have been decoded (i.e not equal to *), and if there is still a missing symbol (i.e *) in a frame of another ED y. If it is the case, then the algorithm replaces the * in the symbol j of y by the remaining current symbol in the same position j of the superposed symbols. As long as there are missing symbols that can not be decoded by the gateway, the process is repeated until the decoding of all colliding signals.

Algorithm 1 Bitmap processing algorithm.

```
1: while a frame contains * do
2:     the gateway guesses a possible frame and sends it
3:     for each colliding end-device x do
4:         x sends a bitmap b_i^x
5:         for each bit at position j in b_i^x do
6:             if bit = 1 then
7:                 symbol j of the frame f_x ← symbol j of the gateway frame
8:             else if bit = 0 and the number of superposed symbols at the current
9:                 position j is 2 then
10:                symbol j of f_x ← the other symbol at position j of the superposed
11:                symbols
12:            end if
13:        end for
14:        for each colliding end-device y do
15:            for each symbol at position j do
16:                if y ≠ x and all symbols j of f_x are different from * and the symbol j
17:                    of f_y is * then
18:                    symbol j of f_y ← the remaining symbol j of the superposed symbols
19:                end if
20:            end for
21:        end for
22:    end for
23: end while
```

Guessing the frame: In this paragraph, we show that the choice of symbols by the gateway has an impact on the number of bitmap transmissions needed for each end-device. We recall that when a collision occurs between frames, the gateway stores all superposed symbols at each symbol duration. Then, it sends a frame built from these symbols. The gateway waits for bitmap generation from the end-devices in order to decode the colliding frames. As long as the frames sent by the end-devices are not completely decoded yet, the gateway sends a new frame, and the end-devices reply by sending new bitmaps. Hence, the number of bitmap transmissions by the end-device is influenced by the frame sent by the gateway.

We refer to Fig. 5.2 to give an example of our proposed algorithm which is described by Algorithm 1. We give two examples to decode the colliding signals of Fig. 5.2. The first example is given in Table 5.1, and the second example is given in Table 5.2.

By referring to Table 5.1, the steps needed to decode the colliding frames of Fig. 5.2 are the following:

Step 1: The gateway sends a frame with the following arbitrary set of symbols $f_{G1} = (64, 0, 32)$. ED1 replies with the bitmap $b_1^1 = (1, 0, 1)$, ED2 replies with $b_1^2 = (0, 1, 1)$, and ED3 with $b_1^3 = (0, 0, 1)$. From the gateway perspective, the current data frame of ED1 corresponds to $f_1 = (64, *, 32)$, the current data frame of ED2 corresponds to $f_2 = (96, 0, 32)$, and the current data frame of ED3 corresponds to $f_3 = (96, *, 32)$. The symbol 96 was obtained because there were only two possible symbols in the first symbol duration, and the symbol was not 64 for f_2 and f_3.

Step 2: Since some of the frames of the end-devices still contain missing symbols that cannot be deduced by elimination, the gateway sends another frame $f_{G2} = (96, 0, 32)$. ED1 replies with $b_2^1 = (0, 0, 1)$, and ED3 with $b_2^3 = (1, 0, 1)$. The updated frames of ED1 and ED3 remain the same as in *Step 1*, (i.e $f_1 = (64, *, 32)$, $f_2 = (96, 0, 32)$ and $f_3 = (96, *, 32)$). ED2 did not reply since its frame was decoded in *Step 1*.

Step 3: Since some of the frames of the end-devices still contain missing symbols that cannot be deduced by elimination, the gateway sends another frame $f_{G3} = (96, 32, 32)$. ED1 replies with $b_3^1 = (0, 1, 1)$, and ED3 with $b_3^3 = (1, 0, 1)$. Now the updated frame of ED1 is $f_1 = (64, 32, 32)$, and the updated frame of ED3 is $f_3 = (96, 64, 32)$. Note that the second symbol of f_3 is decoded by deduction.

In this example, the average number of bitmap transmissions for each end-device is

	Gateway frame	End-device bitmaps	Current end-device frame
Step 1	$f_{Gw_1} = (64, 0, 32)$	$b_1^1 = (1, 0, 1)$ $b_1^2 = (0, 1, 1)$ $b_1^3 = (0, 0, 1)$	$f_1 = (64, *, 32)$ $f_2 = (96, 0, 32)$ $f_3 = (96, *, 32)$
Step 2	$f_{Gw_2} = (96, 0, 32)$	$b_2^1 = (0, 0, 1)$ $b_2^3 = (1, 0, 1)$	$f_1 = (64, *, 32)$ $f_2 = (96, 0, 32)$ $f_3 = (96, *, 32)$
Step 3	$f_{Gw_3} = (96, 32, 32)$	$b_3^1 = (0, 1, 1)$ $b_3^3 = (1, 0, 1)$	$f_1 = (64, 32, 32)$ $f_3 = (96, 64, 32)$

Table 5.1: First example for decoding the superposed signals of Figure 5.2: three steps are required.

2.33 transmissions.

We now consider new arbitrary frames. By referring to Table 5.2, the steps needed to decode the colliding frames of Fig. 5.2 are the following:

Step 1′ : same as *Step 1*

Step 2′ : The gateway sends $f'_{G2} = (96, 32, 32)$. ED1 replies with $b_2^1 = (0, 1, 1)$, and ED3 with $b_2^3 = (1, 0, 1)$. So the updated frames of ED1 and ED3 become $f_1 = (64, 32, 32)$, and $f_3 = (96, 64, 32)$ respectively.

Here, the average number of bitmap transmissions for each end-device is 1.66 transmissions.

Therefore, the choice of the symbols by the gateway impacts the number of needed bitmap transmissions for each end-device.

In our algorithm, we propose a random selection of symbols that are not already sent by the gateway.

	Gateway frame	End-device bitmaps	Current end-device frame
Step $1'$	$f_{Gw_1} = (64, 0, 32)$	$b_1^1 = (1, 0, 1)$ $b_1^2 = (0, 1, 1)$ $b_1^3 = (0, 0, 1)$	$f_1 = (64, *, 32)$ $f_2 = (96, 0, 32)$ $f_3 = (96, *, 32)$
Step $2'$	$f_{Gw_2} = (96, 32, 32)$	$b_2^1 = (0, 1, 1)$ $b_2^3 = (1, 0, 1)$	$f_1 = (64, 32, 32)$ $f_3 = (96, 64, 32)$

Table 5.2: Second example for decoding the superposed signals of Figure 5.2: two steps are required.

5.4 Parameter settings

Simulations are carried out using our own simulator developed in Java. We considered only one gateway in a saturated network. We made several simulations and varied several parameters to study their impact on the system performance. For some simulations, we varied the number of end-devices but we set the payload size of the sent frames to 10 bytes. For other simulations, we increased the size of the sent frames. We also varied the duty cycle (referred to as dc) to two values: $dc = 1\%$ and $dc = 10\%$, and the SF of the end-devices to 7 and 12. The end-devices were operating as Class A end-devices and used the fixed transmit power of 14 dBm (25 mW), which is the default radiated transmit power for EU868MHz ISM band [5]. All end-devices are sending on the same channel with the same SF, and we assume no capture conditions. We set the bandwidth to 125 kHz in order to have a fair comparison of the delay, as the delay depends on both the bandwidth and the SF [72]. We varied the number of end-devices in the network to the following values {5, 10, 20, 40, 80, 160, 320, 640}. Then we classified the network into small (where the number of end-devices is in the range of 5 - 80), or large network (where the number of end-devices is in the range of 160 - 640). Simulation results are obtained by averaging over one thousand samples.

In the conventional LoRaWAN protocol, we assume that when an end-device transmits an uplink frame, the gateway sends an ACK only during Rx1, not during Rx2 for the sake of simplicity. If the ACK is not received, the end-device retransmits the same frame until an ACK is received in Rx1, or until a maximum number of transmissions attempts (equal to 8) is reached. Hence, we consider that each retransmission is started after an

ACK_TIMEOUT initiated at the start time of the last second receive window (i.e. Rx2), and equal to a random value between 1 and 3 seconds as in LoRaWAN specifications.

In the slotted-backoff LoRaWAN protocol, we assume that the time is divided into slots and that frame transmissions on the same slot are fully synchronized. As in the conventional LoRaWAN, the gateway sends an ACK during Rx1 only. We chose $BE = 3$. The number of preamble symbols is set to 8. The minimum duration needed to detect a preamble (i.e $min_{Slot\,Duration}$) is set to 12.54 ms (the preamble duration under SF7). We set the number of slots $nmaxslots$ in a beacon period to 3000 slots. Furthermore, in case of collision, we consider 8 retransmissions for successful reception, which is the default number of retransmissions attempts in LoRaWAN [5]. We used $Slot_{Duration} = 5\ s$.

In our proposed MAC protocol, we assume that the time is divided into slots and that frame transmissions on the same slot are fully synchronized. We assume that when an ED transmits a frame, the gateway sends an ACK during Rx1 only. If an ACK is not received, the ED retransmits the corresponding bitmap until an ACK is received in Rx1, or until a maximum number of 8 transmissions attempts is reached. Hence, we consider that each bitmap retransmission is started after an ACK_TIMEOUT initiated at the start time of Rx2, and equal to a random value between 1 and 3. The network is saturated, hence all the three aforementioned types of collisions (i.e. frame collision, bitmap collision, and mixed collision) may occur. We also set the number of slots $nmaxslots$ in a beacon period to 3000 slots. Furthermore and as stated previously, in case of collision, we consider 8 retransmissions of bitmaps until a successful reception, as done in LoRaWAN. We used $gap = 1\ s$, and $Slot_{Duration} = 5\ s$.

5.5 Simulation results

In this section, we evaluate and compare the network performance in terms of system delay, energy consumption, and system throughput for the conventional LoRaWAN protocol, the slotted-backoff LoRaWAN protocol, and our proposed MAC protocol.

5.5.1 Number of colliding frames

Figure 5.4 shows the average number of colliding frames in both the conventional LoRaWAN and slotted-backoff LoRaWAN protocols[2] for a small network. We use SF7 and a duty cycle of 1%. We notice a decrease in the number of collisions with the slotted-backoff LoRaWAN. Hence, the slotted-backoff LoRaWAN protocol brought an improvement in comparison with the conventional LoRaWAN protocol due to channel sensing. Additionally, it is obvious that the number of collisions increases with the number of end-devices in the network.

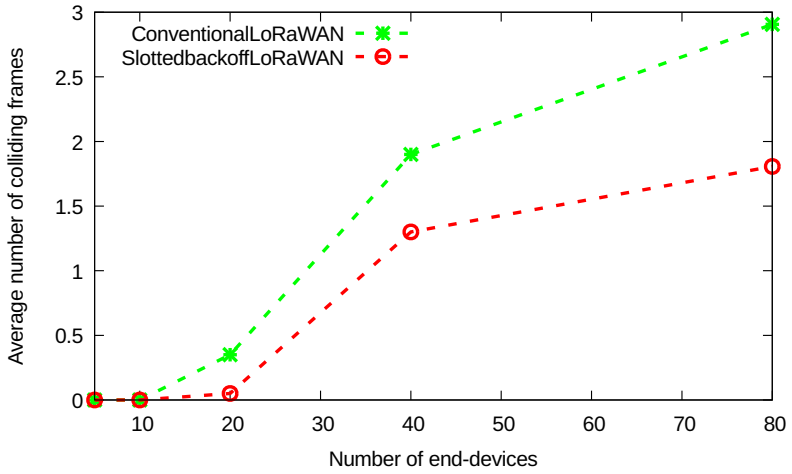


Figure 5.4: Average number of colliding frames in both the conventional LoRaWAN and slotted-backoff LoRaWAN protocols for a small network.

Figure 5.5 shows the average number of colliding frames according to the total number of end-devices in the network for two different duty cycles ($dc = 1\%$ and $dc = 10\%$), with SF7, in the conventional LoRaWAN protocol[3]. Here, we use a small network of 5, 10, 20, 40 and 80 end-devices. It is obvious that when the number of end-devices in the network increases, the number of collisions increases too. Moreover, we can notice that when the duty cycle increases, the number of colliding end-devices increases considerably

[2]Here we did not include our proposed algorithm. Our goal is to show that the slotted-backoff LoRaWAN has decreased the number of collisions compared to the conventional LoRaWAN. However, the colliding frames remain undecodable by the receiver.

[3]Here we did not include the slotted-backoff LoRaWAN or our proposed algorithm. Our goal is only to show the impact of increasing the duty cycle.

with the total number of end-devices in the network. Otherwise, with a low duty cycle, the number of colliding end-devices increases slightly with the total number of end-devices in the network. This is due to the fact that with a large *dc*, each end-device transmits more frequently, which leads to more collisions in the network.

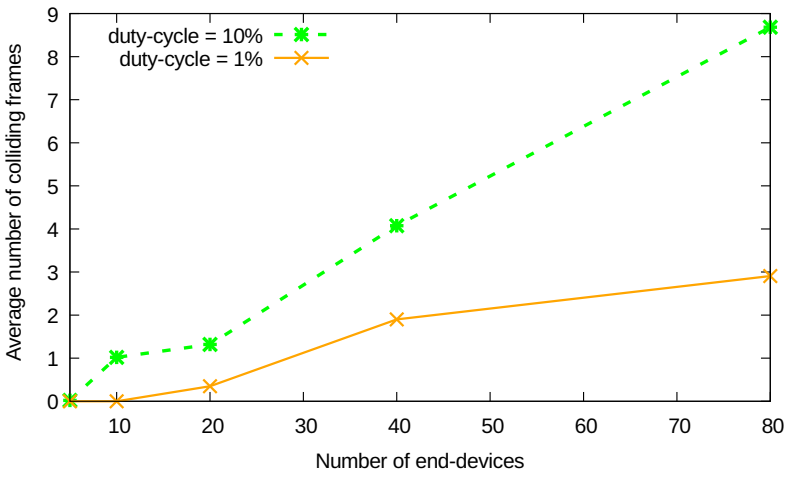


Figure 5.5: The average number of colliding frames increases with the total number of end-devices in the network and with the duty cycle (small network).

5.5.2 Number of retransmissions and frame losses

Figures 5.6 and 5.7 show the average number of retransmissions needed to successfully decode the colliding frames and the percentage of losses respectively, as a function of the number of end-devices in the network for conventional LoRaWAN, slotted-backoff LoRaWAN, and our MAC protocol. Here, we use a duty cycle of 1% with SF7, and frames with payload size of 10 bytes each. We notice that the number of retransmissions and the percentage of losses increase by increasing the number of end-devices for the three protocols. This is due to the fact that when increasing the load (i.e. the number of EDs in the network), collisions become more important. This causes more retransmissions and hence more losses. We notice also that the number of retransmissions in both the conventional and slotted-backoff LoRaWAN is greater than that in our proposed protocol. This is related to the short size of bitmaps which leads to decrease the time on air and therefore the probability of collisions in our proposed protocol. This reduces further the

number of retransmissions and the percentage of frame losses as shown accordingly in Fig. 5.7. Similarly, we observe that in a small network (as shown in Fig. 5.6 and 5.7 where the number of end-devices is in the range of 5 - 80), the difference between the number of retransmissions and the percentage of losses in the slotted-backoff LoRaWAN and the conventional LoRaWAN is small.

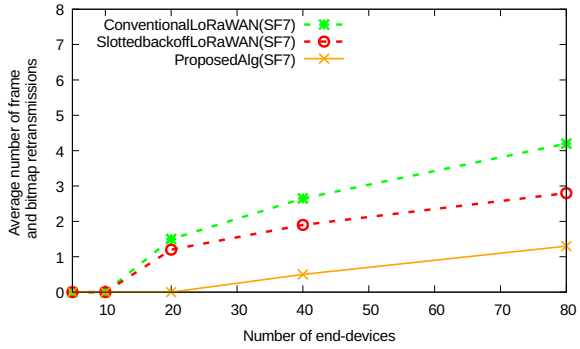


Figure 5.6: Average number of bitmap and frame retransmissions in a small network with dc = 1%.

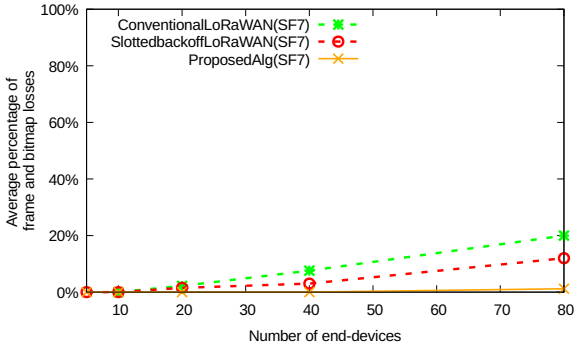


Figure 5.7: Average percentage of bitmap and frame losses in a small network with dc = 1%.

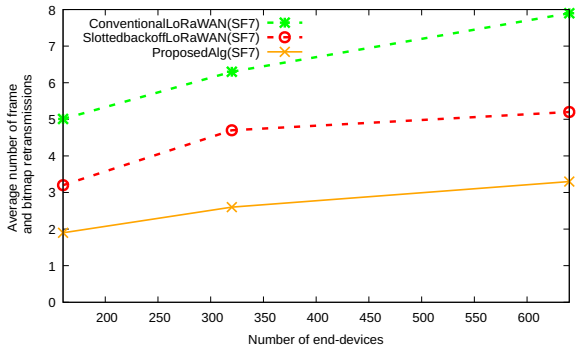


Figure 5.8: Average number of bitmap and frame retransmissions in a large network with dc = 1%.

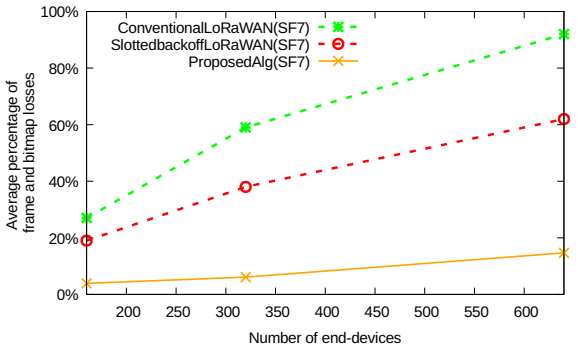


Figure 5.9: Average percentage of bitmap and frame losses in a large network with dc = 1%.

Then, we increased the number of end-devices in the network. Figures 5.8 and 5.9 present the number of retransmissions and the percentage of losses respectively, according to the number of end-devices in the range of 160 - 640. We notice that the difference between the slotted-backoff LoRaWAN and the conventional LoRaWAN increases considerably compared to the range of 5 - 80 of Fig. 5.6 and 5.7, where the difference between

both versions of LoRaWAN protocols was small. We found that this difference depends on the network load. If the network load is high, there are more collisions and retransmissions in the conventional LoRaWAN than in the slotted-backoff LoRaWAN. Therefore the difference increases between them in favor of the slotted-backoff LoRaWAN. It is obvious that in slotted-backoff LoRaWAN, the probability to send frames in collisions is lower than in conventional LoRaWAN as it is based on carrier sensing, while in the conventional LoRaWAN, the end-devices use pure ALOHA mechanism for transmission which increases the percentage of collisions. For example, with 640 end-devices in the network, the average number of frame retransmissions is almost 8 with 95% losses in conventional LoRaWAN, 5.2 with 62% losses in slotted-backoff LoRaWAN, and 3.3 with 14.7% losses in our algorithm. Our proposed algorithm always outperforms both LoRaWAN protocols regarding the number of retransmissions and frame losses. As a result, we observe that the percentage of losses is reduced in our algorithm by 75% compared to slotted-backoff LoRaWAN.

In the next simulations, we increased the spreading factor to SF12. Figures 5.10 and 5.11 show the average number of retransmissions and the percentage of losses respectively as a function of the number of end-devices in the network for the three aforementioned protocols. Here, we use a duty cycle of 1%. We notice that the number of retransmissions in both the conventional and slotted-backoff LoRaWAN is always greater than that in our proposed protocol. Moreover, we observe that the number of retransmissions and losses with SF12 is greater than the number of retransmissions and losses with SF7 (as already shown in Fig. 5.6 and 5.7). This is because the time on air with SF12 is higher than the time on air with SF7, which increases the number of collisions.

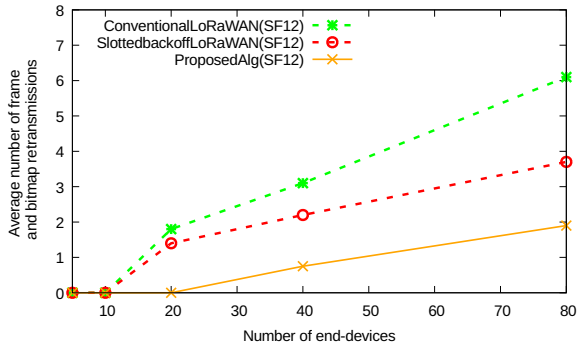


Figure 5.10: Average number of bitmap and frame retransmissions in a small network with $dc = 1\%$.

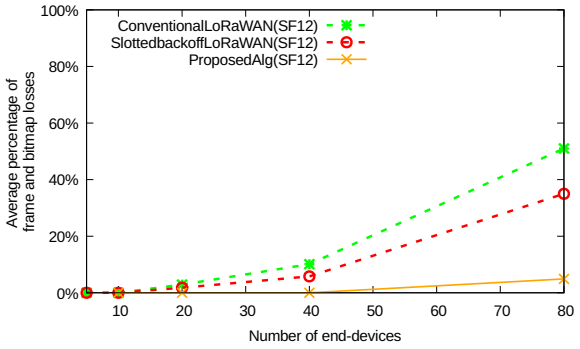


Figure 5.11: Average percentage of bitmap and frame losses in a small network with $dc = 1\%$.

In Figures 5.12 and 5.13, we increased the duty cycle to 10%, and used a small network (i.e. the number of end-devices is in the range of 5 - 80) with SF7. We find that the number of retransmission attempts and losses in the three protocols increases in comparison with Fig. 5.6 and 5.7, where the duty cycle used was 1%, but it remains greater in both versions of LoRaWAN than that in our protocol. This is due to the fact that when using a high duty cycle, the number of colliding frames increases as shown previously in Fig. 5.5. Consequently, the number of retransmissions and losses increases. Therefore, when the duty cycle increases, collisions and losses become larger [87].

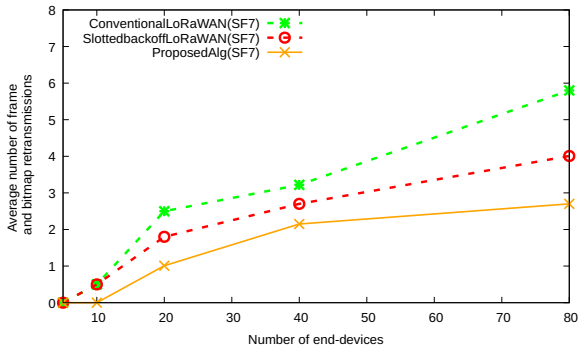


Figure 5.12: Average number of bitmap and frame retransmissions in a small network with $dc = 10\%$.

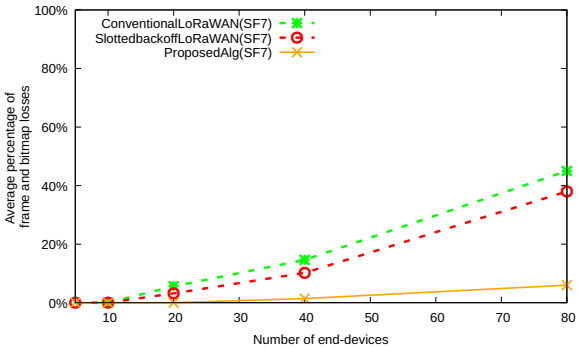


Figure 5.13: Average percentage of bitmap and frame losses in a small network with $dc = 10\%$.

5.5.3 Delay

Figure 5.14 shows the average delay as a function of the number of end-devices for the three aforementioned protocols, with a duty cycle of 1%, spreading factor SF7, and payload size of 10 bytes. We notice that the delay increases with the number of end-devices for all protocols, and that our MAC protocol performs better than the other two protocols, as it shows a delay reduction of 35% compared to slotted-backoff LoRaWAN for small networks. This is due to the fact that our MAC protocol is based on sending bitmaps which have a short size compared to a complete frame, as in the case of both versions of LoRaWAN. Thus, our MAC protocol reduces the duration between retransmissions and hence the delay. Moreover, as our protocol is able to reduce collisions, retransmissions are not always needed. It is worth mentioning that when there is no collision (i.e. the number of end-devices is in the range of 5 - 20), the delay in the conventional LoRaWAN is slightly smaller than the delay in the slotted-backoff LoRaWAN. This is because in the latter protocol, each end-device senses the channel before transmitting and waits for a slot, which increases the delay.

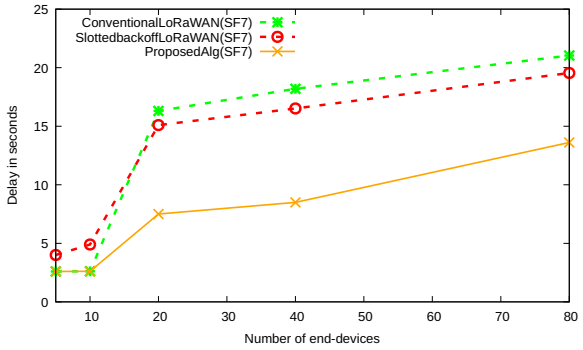


Figure 5.14: The delay computed in a small network.

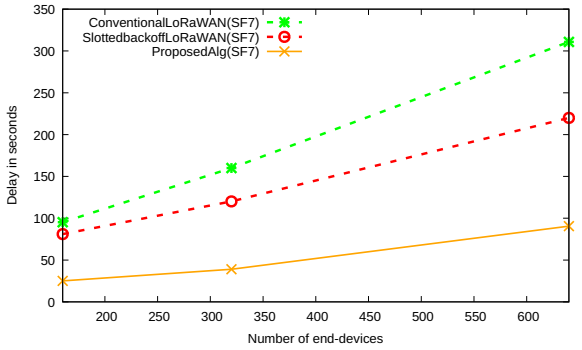


Figure 5.15: The delay computed in a large network.

Figure 5.15 shows that the difference in the delay between the conventional and slotted-backoff LoRaWAN becomes considerable with a large load. When increasing the number of end-devices, the slotted-backoff LoRaWAN is more suitable than the conventional LoRaWAN where the end-devices encounter more collisions, and hence more retransmissions, which increases the delay. Similarly, we observe that our MAC protocol performs better than the other two protocols, and shows a delay reduction of up to 65% compared to

slotted-backoff LoRaWAN for large networks.

Figure 5.16 shows the average delay (using log scale) in terms of the number of end-devices for the three aforementioned protocols with a duty cycle of 1% and SF12. We notice a considerable difference in the delay when increasing the spreading factor. This is because the frame transmission duration greatly depends on SF. With SF12, the transmission duration of a frame is about 32 times larger than with SF7, thus inducing a larger delay for correct frames reception compared to SF7 (see Fig. 5.14). Similarly, we observe that our MAC protocol performs better than the other two protocols, and shows a delay reduction of up to 80% with SF12 compared to slotted-backoff LoRaWAN for small networks.

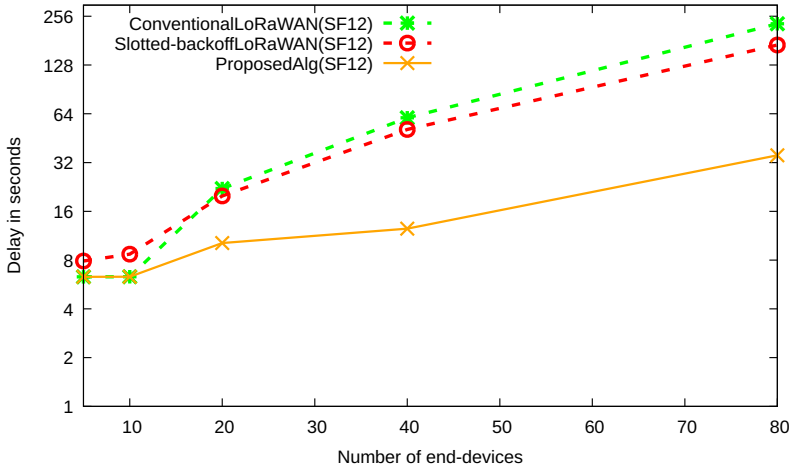


Figure 5.16: The delay according to the number of end-devices for our MAC protocol outperforms the delay for the conventional and slotted-backoff LoRaWAN protocols in small network.

Figure 5.17 shows the average delay in terms of the size of the frames for the three protocols. We varied the payload size in order to show its impact on the delay needed to successfully decode the colliding signals. Here, we fixed the number of end-devices in the network to 80, and increased the payload size to 30 and 90 bytes. We used a duty cycle of 1% with SF7 and SF12. We notice that the delay increases with the size of the frames in all protocols. Indeed, dealing with large frames yields to long transmissions time, and thus long duration for channel unavailability for each end-device. For example, for SF7, a frame of 10 bytes needs 3.8 less time than the duration of a frame of 90 bytes to be transmitted. Moreover, a frame of 30 bytes with SF7 needs about 20 times less than the

duration of the same frame size with SF12. We found that our MAC protocol outperforms both versions of LoRaWAN under all retransmission conditions. In reality, the gain may be even higher because LoRaWAN might use the 8 retransmissions defined in [5] which is not the case for our MAC protocol, as the latter is able to decode superposed signals with lower number of retransmissions attempts using the collision reduction technique (i.e. the transmissions of bitmaps instead of complete frames). For instance, we observe in our MAC protocol a reduction of the delay of up to 60% for large frames compared to slotted-backoff LoRaWAN.

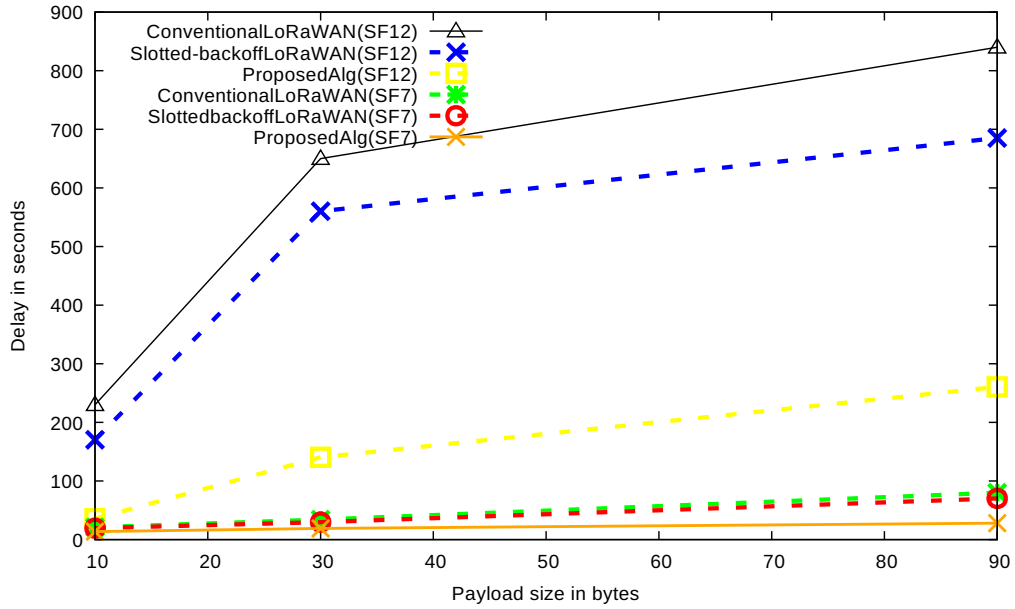


Figure 5.17: The delay according to the payload size for our MAC protocol outperforms the delay for both versions of LoRaWAN.

5.5.4 Energy consumption

In this subsection, we present the evaluation of the energy consumption of the end-devices, as the energy consumption is a crucial aspect in LoRaWAN networks.

Figures 5.18 and 5.19 show the energy consumption (using log scale for Fig. 5.19) computed as a function of the number of end-devices in the network with SF7 and SF12 respectively, for both versions of LoRaWAN and our MAC protocol. Here we used a duty

cycle of 1%. We notice that the energy computed in the three protocols increases with the number of end-devices in the network. This is because when the number of end-devices increases, the number of collisions increases which leads to more retransmissions and hence to increase the consumed energy. Moreover, the consumed energy in both versions of LoRaWAN is greater than that in our algorithm for both SFs. This is due to the long transmission of frames in both versions of LoRaWAN, which is much longer than the short transmission of bitmaps. For instance, we observe a reduction in the energy consumption in our protocol of up to 55% with SF7, and up to 90% with SF12 compared to conventional LoRaWAN for a small network. Note that when the probability of collisions is low (i.e. when the number of end-devices is in the range of 5 - 20), the energy consumption in the conventional LoRaWAN is slightly smaller than the energy consumption in the slotted-backoff LoRaWAN. It is worth mentioning that the energy computed for the three protocols increases with the SF as shown in Fig. 5.19. Indeed, the greater the value of SF, the more time is taken to send a frame (i.e., long time on air), so the more consumed energy is needed to transmit data. In other words, frames transmitted with SF12 have a large time on air which results into more energy consumption per time unit, compared to SF7.

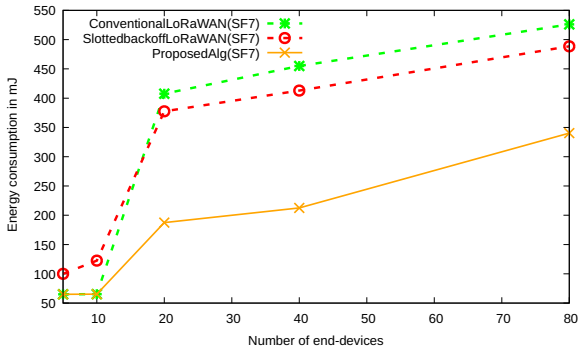


Figure 5.18: The energy consumption with SF7.

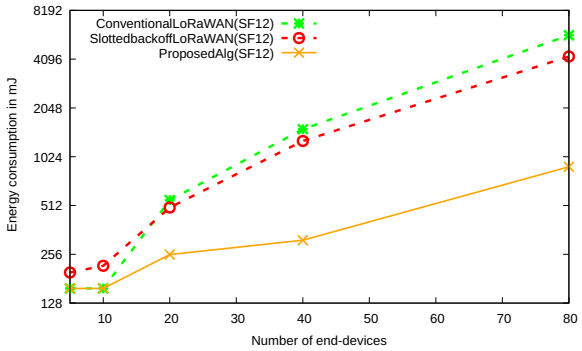


Figure 5.19: The energy consumption with SF12.

Figure 5.20 shows the energy consumption (using log scale) computed for both versions of LoRaWAN and our MAC protocols according to the payload size, for SF7 and SF12 and with $dc = 1\%$. Here, we fixed the number of end-devices in the network to 80 and we varied the size of the frames. It is obvious that the energy computed by both versions

of LoRaWAN and our MAC protocol increases with the size of the frames and with SF. Indeed, when the size of the frame increases, more time is taken to send it (i.e., long time on air), which results into more energy needed to transmit the frame. Results show that the consumed energy in both versions of LoRaWAN is greater than in our protocol due to the long delay resulting from the transmission of large frames versus the transmission of short bitmaps. Compared to conventional LoRaWAN, we observe in our MAC protocol a gain between 40% and 55% using SF7, and a gain between 70% and 90% using SF12.

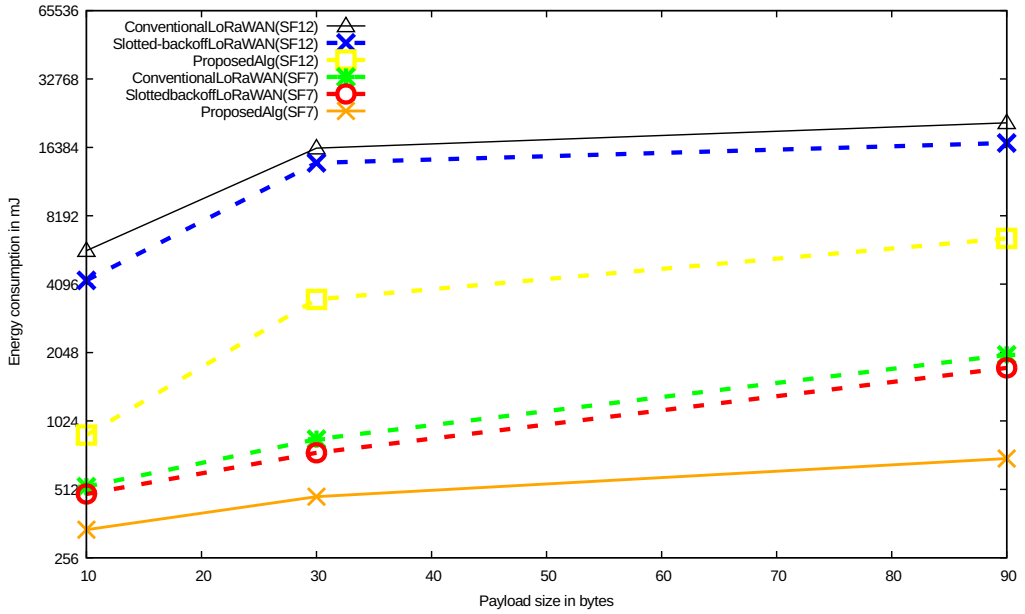


Figure 5.20: The energy consumption according to the payload size for our MAC protocol outperforms the energy consumption for both versions of LoRaWAN.

5.5.5 Throughput

Collision is a factor that negatively impacts LoRaWAN throughput, which is already very limited. In this subsection, we present the evolution of the throughput computed for an end-device in both versions of LoRaWAN and in our proposed algorithm.

Figure 5.21 shows the average throughput (using log scale) as a function of the number of end-devices in the network for both versions of LoRaWAN and our MAC protocol. Here we use SF7 with $dc = 1\%$. We notice that the average throughput in the three

aforementioned protocols decreases with the number of end-devices. This is due to the fact that when the number of end-devices in the network increases, collisions occur more frequently. Moreover, we notice that the throughput for an end-device in both versions of LoRaWAN is smaller than that for an end-device in our algorithm. This is due to the delay in both LoRaWAN which is greater than the delay in our algorithm, and which leads to decrease LoRaWAN throughput compared to our algorithm. In addition, we have more frame losses in both versions of LoRaWAN compared to our proposed protocol, which decreases further the throughput. We notice that our protocol enables an increase in the throuphput of up to 55% with SF7 compared to conventional LoRaWAN. This shows that our algorithm provides a remarkable throughput gain.

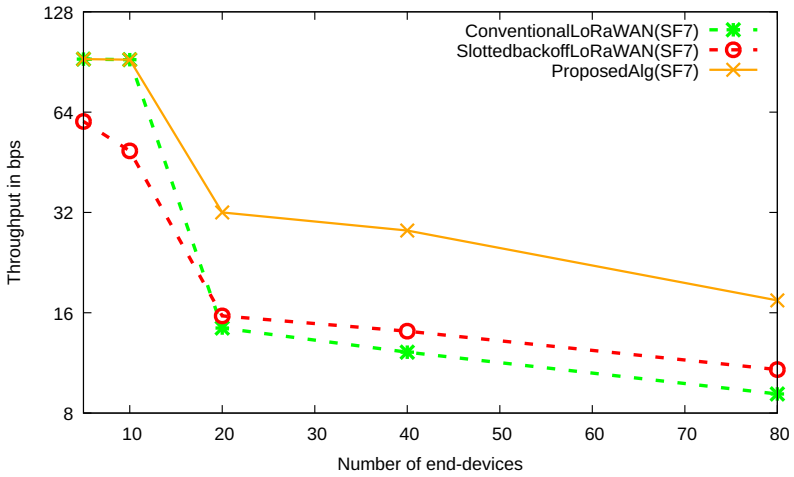


Figure 5.21: The throughput according to the number of end-devices for our MAC protocol outperforms the throughput for both versions of LoRaWAN.

Figure 5.22 shows the average throughput (using log scale) as a function of the number of end-devices in the network for both versions of LoRaWAN and our MAC protocol. Here we use SF12 with $dc = 1\%$. We notice that the throughput computed with SF7 is larger than that with SF12. Moreover, we notice that our proposed protocol shows a gain in the throughput of up to 85% with SF12 compared to conventional LoRaWAN. We observe that the performance of LoRaWAN degrades consistently compared to our MAC protocol for both spreading factors SF7 and SF12, since the percentage of frame losses is greater in LoRaWAN than in our algorithm. For instance, the percentage of frame losses with

SF12 is about 10 times lower in our proposed algorithm than in conventional LoRaWAN as shown previously in Fig. 5.11. This percentage of frame losses is less drastic using our MAC protocol. This is due to the fact that with our proposed decoding technique (i.e. the transmissions of bitmaps instead of complete frames), our MAC protocol is able to reduce the number of collisions, and hence the number of retransmissions with losses. In addition, the delay in both versions of LoRaWAN is greater than the delay in our protocol, which leads to further decrease the LoRaWAN throughput. Consequently, our MAC algorithm enables better collision decoding since it corresponds to sending short bitmaps.

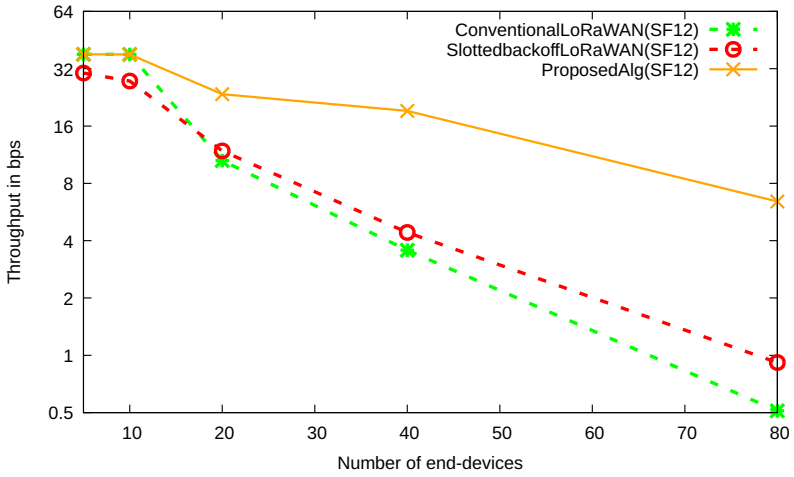


Figure 5.22: The throughput according to the number of end-devices for our MAC protocol outperforms the throughput for both versions of LoRaWAN.

5.6 Conclusion

Collisions in LoRaWAN are damaging to the overall network performance. When a gateway receives several superposed LoRa signals with similar receive power levels, on the same channel and with the same SF, LoRaWAN is unable to decode these signals which are hence lost. In this chapter, we proposed a collision resolution algorithm at the physical layer that enables to decode synchronized colliding frames in LoRa while mitigating the harmful effects of collisions (i.e frame retransmissions and losses). We also propose a MAC algorithm in order to synchronize end-devices and to retransmit bitmaps in reply to

guesses from the gateway instead of retransmitting whole frames. Based on our simulation results, we show that our proposed MAC algorithm is able to significantly improve the network performance, in terms of system throughput and energy consumption. In addition, our proposed protocol enables significant delay reductions needed to successfully decode the colliding frames.

Part IV

CONTRIBUTION II

Chapter 6

Gateway Selection for Downlink Communication

Contents

6.1 Introduction

In LoRaWAN, end-devices send data to the network server through gateways. End-devices and gateways communicate using LoRa, while gateways and the network server communicate over an IP network [79]. We recall LoRaWAN architecture in Fig. 6.1.

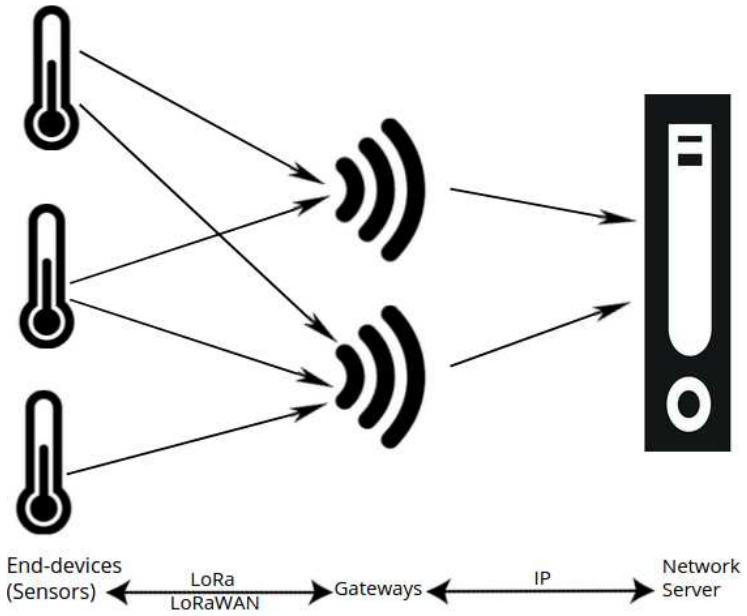


Figure 6.1: LoRaWAN architecture.

When an end-device transmits an uplink message, all gateways that receive this message transmit it to the network server, which removes duplicates. When the network server has a frame to send to an end-device, it selects a single gateway to relay this frame. However, LoRaWAN does not specify how to select this gateway.

In this chapter, we focus on the gateway selection for downlink communications in LoRaWAN in order to improve the throughput of the network. Moreover, we work on decreasing the energy consumption of the end-devices as well as decreasing the delay of the frames. In this regard, we present and evaluate several algorithms for selecting the best gateway for downlink communications, while increasing LoRaWAN throughput, and decreasing the delay and the energy consumption for different types of gateway deployment. In addition, we study the impact of SF, which is a key characteristic in LoRaWAN,

on the overall network energy consumption. We show that the choice of SF is crucial for realizing long-term performance of a LoRaWAN device. We also explain how to find the right balance between battery performance of end-devices and network throughput.

Regarding the overall system performance, our contributions are four-fold:

1. We study three types of gateway deployments and we show that the system throughput, delay and energy consumption of the end-devices depend on this deployment.
2. We show that balancing the number of end-devices per gateway (also known as load) improves the throughput compared to choosing the gateway with the highest signal quality.
3. We show that combining load and signal quality does not further improve the throughput.
4. We show that choosing the gateway with the highest signal quality reduces the delay and the energy consumption compared to choosing the gateway with the lowest load.

6.2 Propositions

In this section, we present our contributions. First, we classify the gateway deployments into three scenarios. Then, we study the existing algorithms for gateway selection.

6.2.1 Scenarios of gateway deployment

Many research works [112–119] on LoRaWAN consider very different scenarios without classifying them. Therefore, we propose in this chapter a classification of deployment scenarios, as they have a large impact on the throughput. We also show this impact.

In this subsection, we classify the gateway deployment into three scenarios: urban scenario, environmental scenario, and hybrid scenario. To do this, we consider a 2-dimensional space where end-devices and gateways are deployed.

Urban scenario is mostly used for monitoring applications such as smart cities with smart parkings and smart buildings [120], [70], [121]. In this deployment, all end-devices are in communication range of all gateways. Figure 6.2(a) shows an example of urban

deployment covering a city. In this example, all devices are deployed in the bottom-left quarter.

Environmental scenario is moslty used for monitoring applications such as volcanoes, forests, agriculture and lakes [101], [122], [123]. In this deployment, end-devices are deployed in the critical zone and send data to gateways that are localized far away from the monitoring area. Figure 6.2(b) shows an example of environmental deployment on a volcano. In this example, we consider that end-devices are covering a part of the volcano which is located in the bottom-left quarter of the figure. However, gateways are deployed in a distant city which is located in the top-right quarter of the same figure. We refer to this scenario as Env.

Hybrid scenario can be used for the same monitoring applications as in urban and environmental scenarios. For example, this deployment can be used for smart industrial control where sensors in manufacturing plants or mobile industries can relay critical data to a LoRaWAN network where it can be analyzed [124]. In this deployment, the gateways and the end-devices are not all scattered in the same area (i.e., the gateways area is larger than the end-devices area). Indeed, Fig. 6.2(c) shows that one gateway is in the vicinity of all end-devices, and that the remaining gateway is further away from these end-devices.

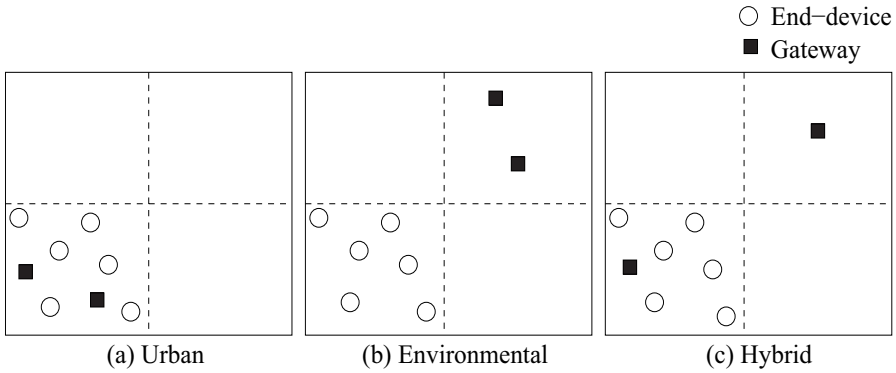


Figure 6.2: Example of the three scenarios of gateway deployment.

6.2.2 Gateway selection algorithms

We now identify three classes of algorithms that the network server might use in order to select the suitable gateway for each end-device in downlink communications.

The first algorithm is based on the received signal strength indicator (RSSI). The RSSI value is the signal power received by the receiver and can be used as an indication of how well a receiver can hear a signal from a sender. The goal of this algorithm is to increase the link quality and the throughput. In the following, we refer to this algorithm as Alg-HR (where Alg stands for algorithm, and HR stands for highest RSSI). Using Alg-HR, the network server selects, among all gateways that are in communication range with an end-device, the gateway receiving frames from the end-device with the highest RSSI. Figure 6.3(a) shows an example of Alg-HR selection. In this example, we consider four end-devices and two gateways. We assume that all end-devices are in communication range with both gateways. As end-devices A, B, and C are closer to gateway Gw1 than to gateway Gw2, the network server selects Gw1 to communicate with end-devices A, B, and C. End-device D is closer to Gw2 than to Gw1. Thus, the network server selects Gw2 to communicate with D. This is probably the algorithm used by most network servers and which serves as a reference.

The second algorithm balances the number of end-devices per gateway. The goal of this algorithm is to reduce the load of a gateway, and to balance the number of ACKs that can be sent by this gateway. In the following, we refer to this algorithm as Alg-LB (where LB stands for load balance). Using Alg-LB, the network server selects, among all gateways that are in communication range with an end-device, the gateway with the lowest load (i.e the lowest number of end-devices), in order to communicate with this end-device. Figure 6.3(b) shows an example of Alg-LB selection. In this example, in order for Gw1 and Gw2 to have the same load, end-devices A and D are associated to Gw2 while end-devices B and C are associated to Gw1.

The third algorithm is a combination of Alg-HR and Alg-LB. In the following, we refer to this algorithm as Alg-LBHR. Using Alg-LBHR, the network server selects a gateway among all gateways that are in communication range with an end-device. The selected gateway is a gateway that has not reached the maximum load yet (since there is a maximum target load). If there are several such gateways, the one with the highest RSSI is selected. Figure 6.3(c) shows an example of Alg-LBHR. In this example, Gw1 and Gw2 have the same load as in Fig. 6.3(b). However, the assignment of gateways for each end-device is different from Alg-LB as the selection is also based on the RSSI. Thus, each gateway tends to have the same number of end-devices and the end-devices are closer to

the gateway from which they receive their downlink communications (which is not the case in Alg-LB). Hence, in Alg-LBHR the first priority is the load balancing, then the RSSI comes in a second step.

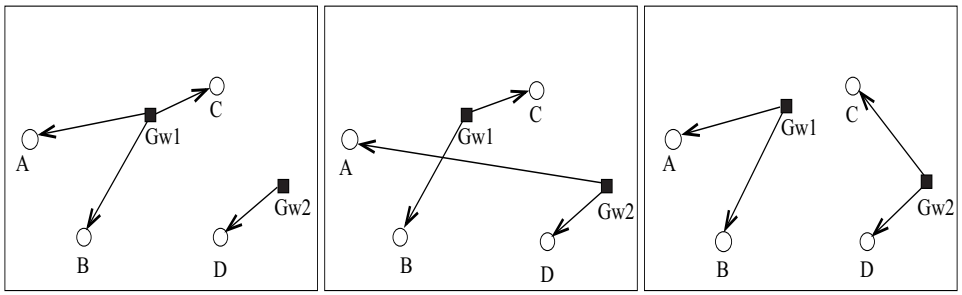


Figure 6.3: Example of the gateway selection algorithms.

6.3 Parameter settings

In this section, we study the performance of gateway selection algorithms on the different scenarios of gateway deployment.

6.3.1 Implementation details on our simulator

Simulations are carried out using our own simulator developed in Java [125] and following the LoRaWAN specification [5]. Our simulator is created solely for simulating LoRaWAN, considering sensor specific characteristics. The network consists of two types of devices (end-devices and gateways) sending their measured data frames to each other for uplink and downlink communications. This simulator is used to model the collision behaviour in LoRaWAN system. Simulation procedure includes building the hardware architecture of the transmitting devices, modeling the communication channel, and the receiving side architecture. It offers basic functionality to simulate LoRa networks. The developed simulator also provides additional features for modeling sensor networks such as sensor channel models, power models (battery and radio), protocol stacks for wireless sensors, frame generation and re-generation. It provides basic layers such as flooding behaviour

for uplink communications as well as MAC layers (i.e., ALOHA protocol). Transmitted frames are monitored, and new frames are generated into the network in a specific duration. Furthermore, the configuration of the debug options is fine grained providing the desired debug output at runtime. The running simulations can be visualized and controlled by the Java-based graphical user interface (GUI). Our developed simulator is portable so that it can be run on the most common operating systems such as Windows, Linux and Mac OS. The collision model we built in our simulator is described in subsection 6.3.3.

We consider that transmissions for all SFs and channels are orthogonal, and that uplink and downlink transmissions do not interfere with each other. In addition, a partial overlapping of two frames triggers a collision. We consider the capture effect conditions where a frame is decoded if the received signal is 6 dB greater than the sum of the signal strengths of all interferers.

We consider the following: for the urban scenario, end-devices and gateways are deployed in an area size of $2 * 2km^2$. For the Env scenario, end-devices are deployed in an area size of $2 * 2km^2$, and gateways are deployed in another area of size $2 * 2km^2$ which is distant from the area of the end-devices. The two areas of $2 * 2km^2$ are in a square of $4 * 4km^2$ as shown in Fig. 6.2(b). For the hybrid scenario, end-devices and one gateway are deployed in an area size of $2 * 2km^2$, and the remaining gateways are deployed in a distant area of $2 * 2km^2$. Likewise, the two areas of $2 * 2km^2$ are in a square of $4 * 4km^2$ as shown in Fig. 6.2(c).

We computed several metrics such as (1) the number of collided frames for uplink communications, (2) the load per gateway and the number of received acknowledgments (ACKs) for confirmed frames for downlink communications, (3) the throughput of LoRaWAN is analyzed and discussed based on the obtained results, (4) the energy consumption of the end-devices and (5) the delay of the frames.

We set some parameters as listed below.

1. We consider a network composed of g gateways with N end-devices. g varies from 2 to 8, and N from 50 to 150. The communication range between end-devices and gateways is about $R = 4\ km$. We also consider a random location for end-devices and gateways. We ensure that each end-device is in communication range with at least one gateway. Consequently, if for example there are two end-devices where

each one is connected only to a single gateway, in this case we try to balance the load for the other gateways for the two algorithms Alg-LB and Alg-LBHR.

2. The RSSI for an end-device is computed using the Hata-Okumura [126, 127] propagation model for medium-sized cities, with a default transmission power of 14 dBm (as explained in the upcoming subsection 6.3.2).

3. We only use the three mandatory 125-kHz channels from the 868-MHz band for data communications in LoRaWAN. The channel used by each end-device is randomly chosen from the following set $\{868.1, 868.3, 868.5\}$ and the duty cycle of 1% is respected. For the second receive window Rx2 we use the fixed frequency 869.525 MHz.

   Simulation results are obtained by averaging over ten thousand samples.

6.3.2 Simulations on the received powers of all interferers

The RSSI is the signal power received in dBm. We recall that the RSSI value can be used as an indication of how well a receiver can hear a signal from a sender. The total received RSSI also includes the interference from other sources than the one sending. In other words, the total received RSSI is the total signal power received at the receiver side including external noise interference. The RSSI is represented in negative dBm, which means that a value closer to 0 indicates a better signal. For example, with RSSI = -30dBm, the signal is considered strong, while with RSSI = -120dBm, the signal is considered weak. Generally, an increase of bandwidth lowers the receiver sensitivity (which is the minimum power level at which the receiving node is able to receive the frames being transmitted), whereas an increase of the spreading factor decreases the sensitivity threshold (and therefore, the sensitivity increases because the receiver is more sensitive). Table 6.1 shows LoRa receiver sensitivity in dBm with bandwidth 125kHz and at different SFs. These RSSI values are the one we used in our simulations after we calculated the sensitivity thresholds using the Hata-Okumura propagation model.

Table 6.1: LoRa receiver sensitivity in dBm with BW=125 kHz and at different spreading factors.

SF	Sensitivity Range (dBm) [128]
SF7	(- 124, 0)
SF8	(- 129, - 124)
SF9	(- 130, - 129)
SF10	(- 133, - 130)
SF11	(- 135, - 133)
SF12	(- 137, - 135)

6.3.3 Collision behavior and interferences

We recall that when two LoRa transmissions overlap at the receiver, several conditions determine whether the receiver can decode the frames. These conditions depend on channel, SF, power and timing. As LoRa is a form of frequency modulation, it presents the capture effect that occurs when two signals are present at the receiver and the weaker signal is suppressed by the stronger one. Therefore, frame x collides with frame y when $P_x - P_y < P_{Threshold}$ ([82], [83]), where P_x is the received signal strength of transmission x, P_y is the received signal strength of transmission y, and $P_{Threshold}$ is a power threshold equal to 6 dB.

Interference Modeling: Using the collision behavior and parameters shown previously with end-devices and gateways as input, we create in our simulator a simulation model for determining the number of end-devices that can be served with a single LoRaWAN gateway. This simulation model also describes and determines the conditions triggering collisions between frames of different end-devices. We generate a vector of spreading factors (SF) used by each end-device such that end-device i uses $SF[i]$, with $i = 1, ..., N$ and N the total number of end-devices served by the gateway. The vector is populated randomly with values for SF ranging from 7 to 12.

Next to this, we generate a second vector of RSSI values at the receiver $RSSI[i]$, with $i = 1, ..., N$ and N the number of end-devices in the network. In other words, this is the RSSI that the gateway observes when the end-device i transmits. Hence, we have a table of RSSI values per gateway. Indeed, the SF that will be used by the end-device is related to the RSSI at the gateway for that end-device. When an end-device is far away from the gateway or its signal is highly attenuated, the RSSI will be low, consequently forcing the

end-device to use a higher SF. Thus, modifying the values of array $SF[i]$.

We further generate a channel vector $CHAN[i]$, with $i = 1, ..., N$. Since we only use three 125-kHz channels from the 868-MHz band for data communication in the LoRaWAN network, the values of the channel vector will be randomly chosen from the interval $[1, 3]$. These three channels at 868 MHz are mandatory for each Class A device.

As a last step, we generate the matrix of the start time of frame transmission $Time[i][j]$, with $i = 1, ..., N$ and $0 \leq j < n$, N the number of end-devices in the network and n the number of frames that each end-device has to send. Hence, i designates the end-device and j the number of the generated frame. It is worth noting that n depends on the SF for each end-device during the tests. In order to respect the 1% duty cycle of the physical layer, two consecutive frame transmission start times are separated at least by a time difference of $(\tau \times 100 - \tau)$ seconds, with τ the on-air time of the previous transmission. For each end-device $i = 1, ..., N$, we iterate through all other possible interferers k with $k \neq i$. If a frame of end-device i overlaps with the time of another frame of end-device k, then the capture effect condition is checked to see if the frame survives the collision or is lost.

In order to identify the collisions, we proceed as shown in Alg. 2.

Algorithm 2 Algorithm to determine the frame collisions.

Data: N, number of end-devices; n_i, number of frames that end-device i sends; SF, spreading factor vector; $CHAN$, channel vector; $Time$, Starting time matrix; $RSSI$, vector of RSSI values for each end-device; τ_i, the time on air of the frame of end-device i.
for i from 1 to N **do**
  for j from 1 to n_i **do**
    for k from 1 to N **do**
      for l from 1 to n_k **do**
        if $(k \neq i)$ and $(SF[i] = SF[k])$ and $(CHAN[i] = CHAN[k])$ **then**
          if $(Time[i][j] \leq Time[k][l]$ **and** $Time[k][l] \leq Time[i][j] + \tau_i)$
            or $(startTime[k][l] < startTime[i][j]$ **and**
            $startTime[i][j] < startTime[k][l] + \tau_k)$ **then**
            test the capture effect condition on frame j of end-device i.
          end if
        end if
      end for
    end for
  end for
end for

6.4 Simulation results

6.4.1 Collisions

We analyze the results of the simulations based on the model described previously with the parameter settings presented in Section 6.3. Figure 6.4 shows the number of transmitted and collided frames per second for uplink communications with $g = 4$ gateways in terms of the number of end-devices in the network. Obviously, the number of transmitted and collided frames increases with the number of end-devices (i.e with the size of the network). In this figure, the percentage of collided frames increases from 5% with 50 end-devices to 15% with 150 end-devices. Therefore, adding more end-devices greatly increases the collision probability. Thus, with a high traffic load, collisions become more important.

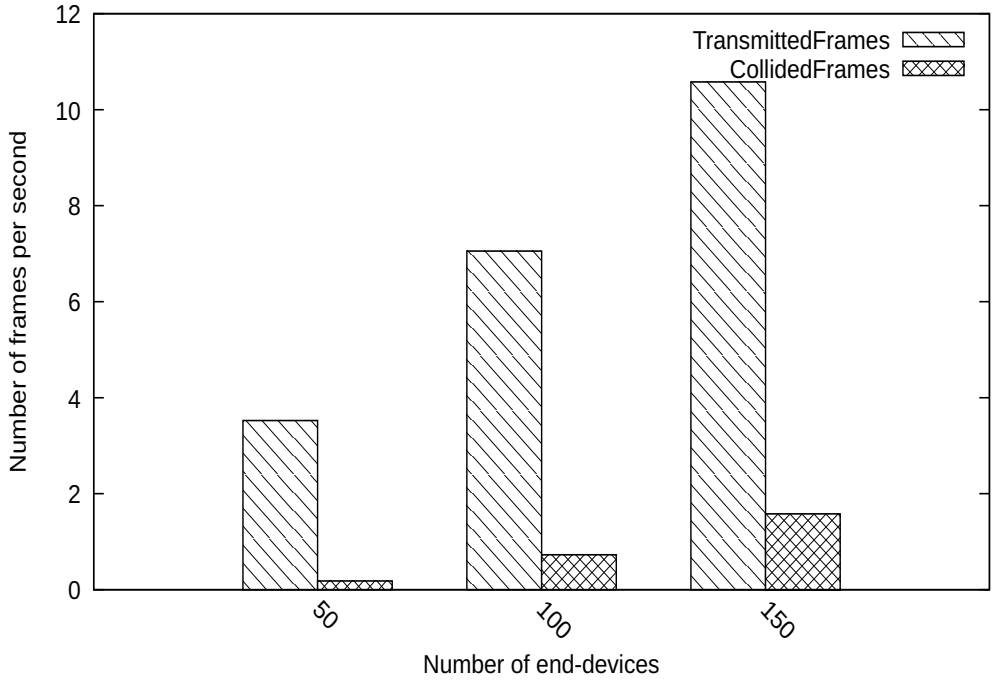


Figure 6.4: Number of transmitted and collided frames per second for uplink communications.

6.4.2 Gateway load

Figure 6.5 shows the average of the maximum load per gateway for each algorithm in each scenario. We used $g = 4$ gateways and $N = 100$ end-devices. We observe that in Alg-HR, the average load per gateway is greater than that in Alg-LB and Alg-LBHR. This is because in Alg-LB and Alg-LBHR, the load is distributed almost equally between the gateways (i.e we are balancing the load between gateways), which is not the case in Alg-HR. Furthermore, we observe that in ENV and hybrid scenarios, the gateway load for Alg-HR is greater than that in urban scenario. This is due to the gateway deployment, since in ENV and hybrid scenarios, one gateway is closer to the end-devices than the other gateways, which yields a high load for this gateway compared to other gateways.

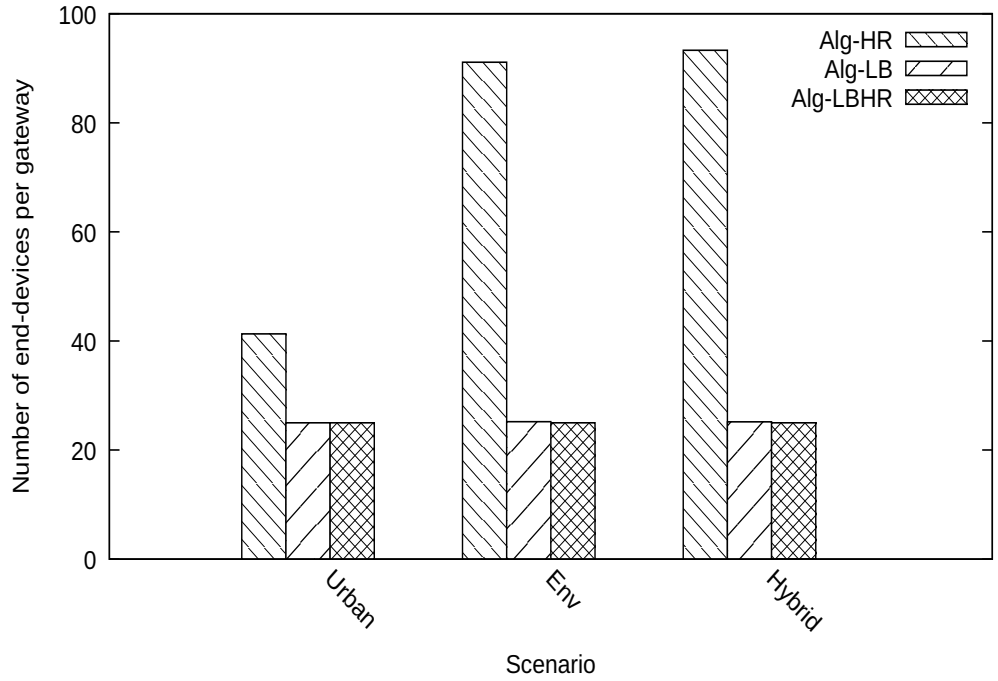


Figure 6.5: Average of the maximum load per gateway with $g = 4$ gateways and $N = 100$ end-devices in downlink communications.

6.4.3 Confirmed throughput

If no collision occurs for an uplink, the frame is acknowledged by the gateway. Indeed, the gateway tries to acknowledge the frame, but sometimes it can not (because of the duty cycle on Rx1 and Rx2, or because it is already busy transmitting another ACK). After each uplink frame transmission, the end-device waits for an ACK from a gateway. For class A end-devices, LoRaWAN acknowledgments can be received either during Rx1 or Rx2.

More simulations were run to study the throughput by extending the previously described simulations to incorporate frame confirmations. We assume the confirmation is a message without a payload that has the ACK bit set to 1 and a length of 1 byte. We consider the following model of class A: after the reception of a frame, the gateway tries to send an ACK in Rx1. If Rx1 is busy (because the gateway is busy transmitting, or because Rx1 is not available due to the duty cycle) the gateway tries to send an ACK in

Rx2. If Rx2 is also busy (because the gateway is busy transmitting, or because Rx2 is not available due to the duty cycle) the ACK is not sent. Then we compare the number of received acknowledged frames in each of the three aforementioned algorithms for the three scenarios[1].

Figure 6.6 depicts the number of received ACKs for confirmed uplink with $g = 4$ gateways and $N = 100$ end-devices. We observe that Alg-HR is the worst compared to Alg-LB and Alg-LBHR as the load is less equally distributed among gateways in Alg-HR (Fig. 6.5). Indeed, when the load per gateway increases, the number of received ACKs decreases as collisions increase. Therefore, the load balancing improves the overall throughput, and avoids bottlenecks caused by an excessive load on a single gateway. Additionally, we observe that the throughput is almost the same in Alg-LB and Alg-LBHR. Hence, we found that combining both the load and the signal quality (as for Alg-LBHR), does not improve further the throughput. For example, results show that for urban scenario, 84.1% of the frames have been acknowledged by the gateway in Alg-LB, compared to 78.9% in Alg-HR; for ENV scenario, 84.1% of the frames have been acknowledged by the gateway in Alg-LB, compared to 48.9% in Alg-HR; and finally for hybrid scenario, 84.1% of the frames have been acknowledged by the gateway in Alg-LB, compared to 48% in Alg-HR.

[1]Retransmissions are not considered in our simulations.

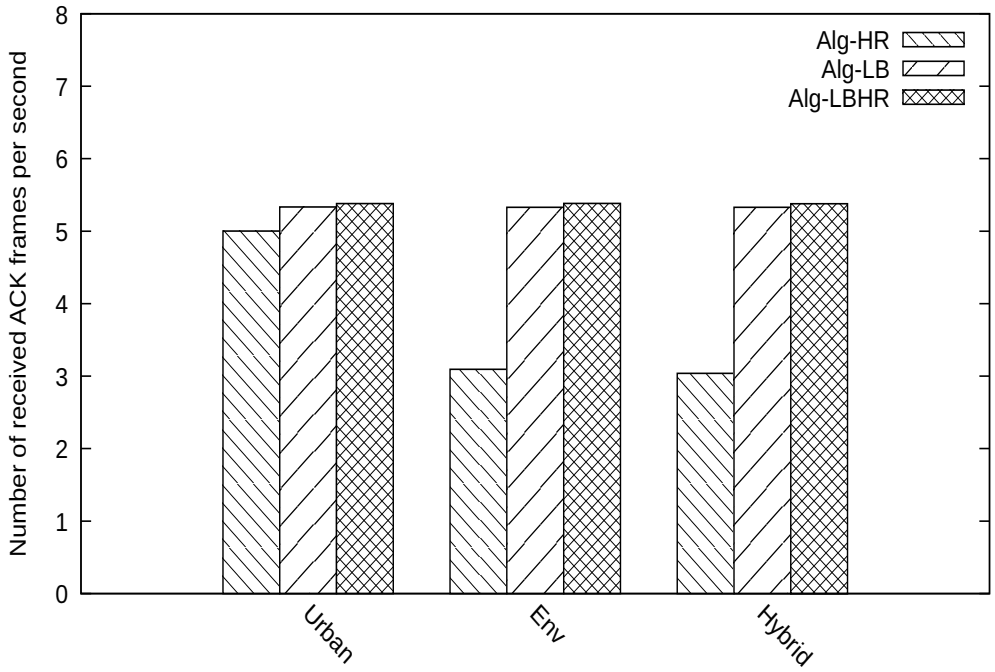


Figure 6.6: Number of received ACK frames with $N = 100$ end-devices and $g = 4$ gateways for the three algorithms in each scenario.

Figure 6.7 shows the number of received ACK frames per second in term of the number of end-devices for the ENV scenario. We notice that the number of received ACKs decreases with the increase in the number of end-devices. This is due to the fact that the number of collisions in uplink communications increases, leading to a decrease in the number of successful frames, and hence to a decrease in the network throughput. Therefore, we notice that LoRaWAN does not scale with the number of end-devices. For example, results show that in Alg-LB, 90% of the ACKs were received with 50 end-devices, while 64% of the ACKs were received with 150 end-devices.

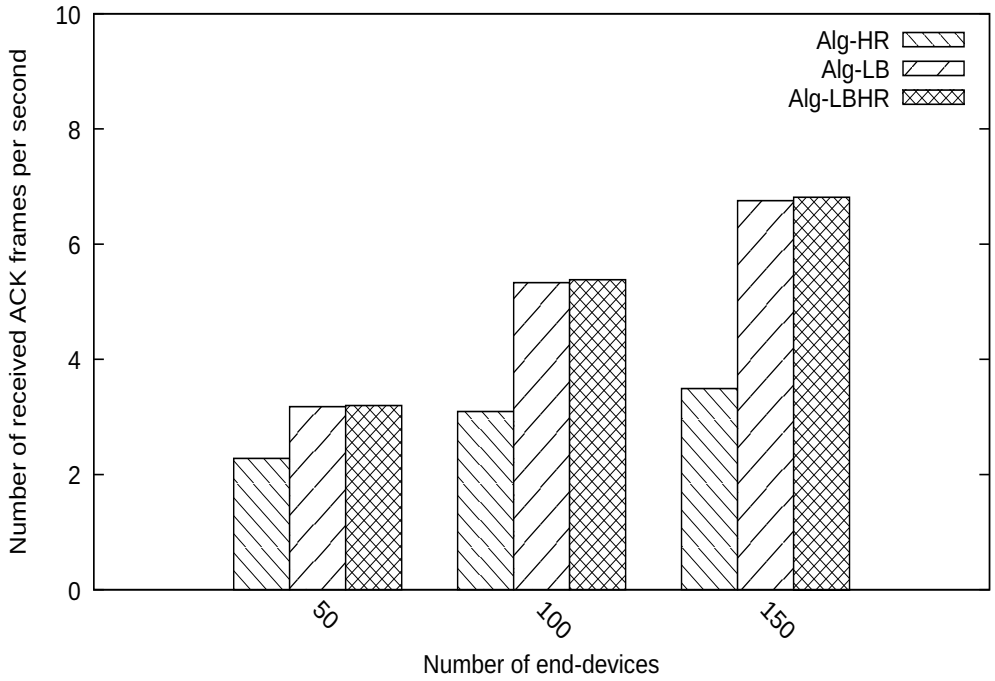


Figure 6.7: Number of received ACK frames according to the number of end-devices with $g = 4$ gateways for ENV scenario.

Figure 6.8 shows the number of received ACK frames per second in terms of the number of gateways for the ENV scenario. We observe that the number of received ACKs increases significantly for both Alg-LB and Alg-LBHR, while it increases slightly for Alg-HR. This is due to the fact that in Alg-LB and Alg-LBHR, the load per gateway decreases when increasing the number of gateways in the network, leading to an increase in the number of received ACKs. Results show that the increase in the number of gateways clearly improves the throughput which reaches, when using 8 gateways, a gain of 45% for Alg-LB and Alg-LBHR compared to Alg-HR. It is worth to mention that in urban scenario, the number of received ACK frames per second in terms of the number of gateways for Alg-HR increases considerably to attain a gain of almost 20%. Hence, adding more gateways is a possible solution for decreasing collisions and improving LoRaWAN throughput.

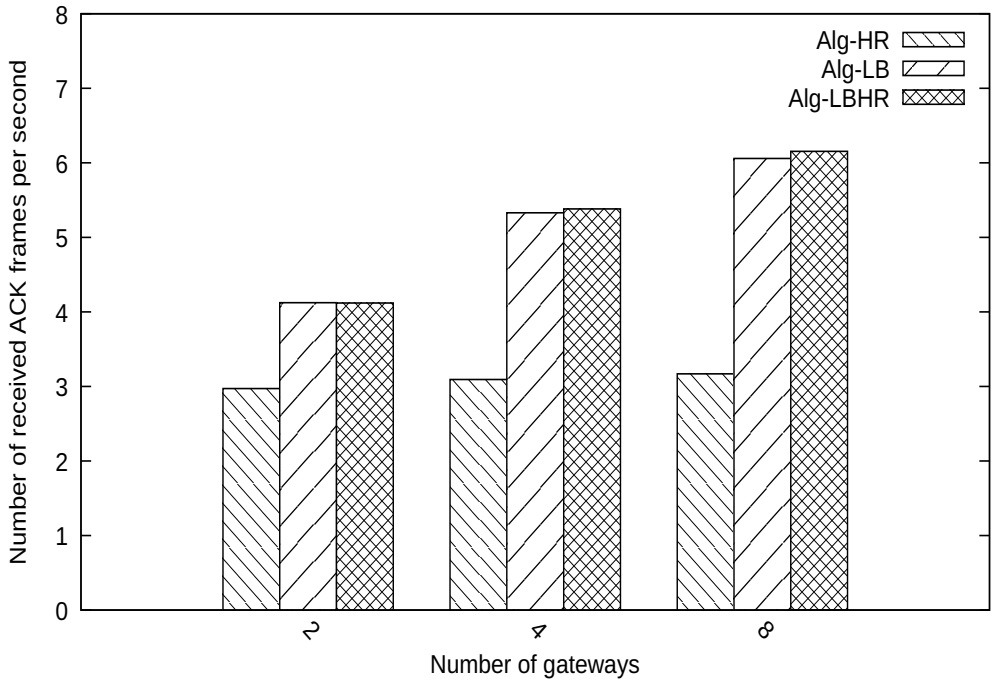


Figure 6.8: Number of received ACK frames according to the number of gateways with $N = 100$ end-devices for ENV scenario.

6.4.4 Delay

In LoRaWAN, one of the factors that can affect the delay is the duty cycle limitation. In other words, due to the durations of sleep state of every device in LoRaWAN, minimizing delay is one of the important issues in such networks. This is because when a frame is generated for an end-device in the sleep state, it is necessary to wait for the end-device to wake up before the frame can be received. Therefore, it is worth working on decreasing the impact of the duty cycle in order to reduce the overall LoRaWAN delay for forwarding messages (i.e. data latency). For this reason, the downlink should always be chosen with the nearest gateway. In this regard, we ran more simulations to check the impact of the three algorithms as well as the three scenarios on the evolution of the delay. Hence, we aim to find the nearest gateway in downlink communications in order to reduce the delay as much as possible. We recall that we focus on the downlink.

Figure 6.9 shows the relationship between the delay for forwarding data and the dis-

tance between end-devices and a single gateway in ENV scenario. We define the average delay of a given frame as the time interval elapsed from the instant it is generated until it is delivered to the end-device. We aim here at presenting the evolution of the delay with the distance between the end-device and the gateway. We evaluate the average delay for distances equal to 500, 1000, 1500, and 2000 meters. We can notice that with the increase of the aforementioned distance, the average delay for forwarding data also increases, which is obvious. In other words, the further we move the gateway away from the end-devices, the more the delay increases. This is because the SF increases with distance, and the time on air increases with SF.

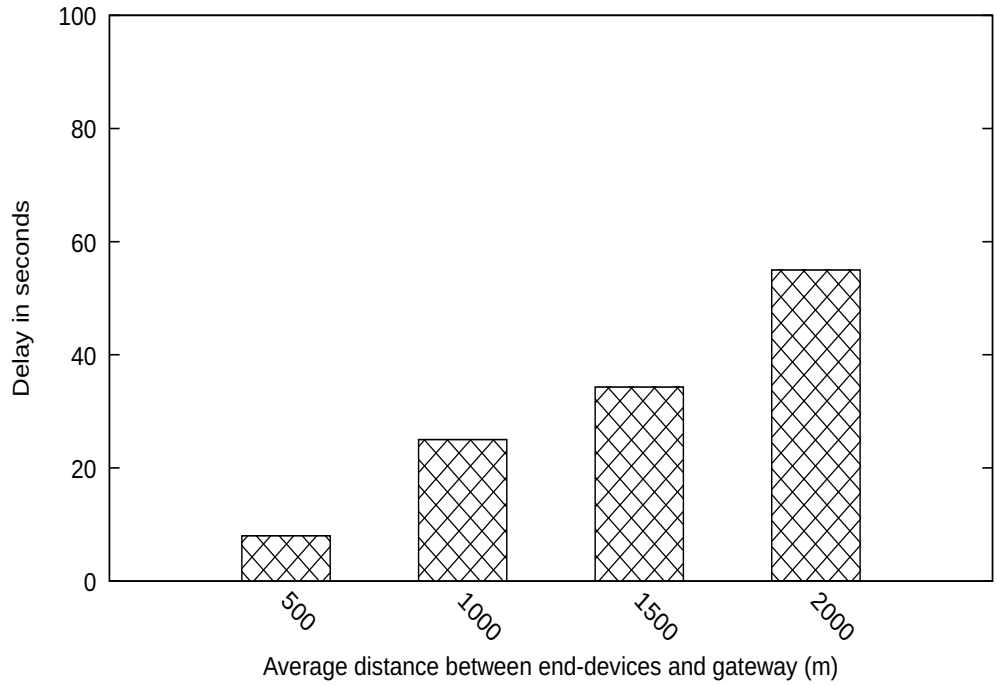


Figure 6.9: Average delay according to the distance between $N = 100$ end-devices and $g = 1$ gateway for ENV scenario.

Figure 6.10 shows the average delay for ACKs delivery in each algorithm and in each scenario. It is worth noting that the delay computed for the three algorithms with the three scenarios decreases the most for Alg-HR. Indeed, the shorter the distance between the end-device and the gateway, the less time is taken to send a frame (i.e., short time on

air) as shown previously in Fig. 6.9. Moreover, we find that the urban scenario gives the lowest delay for delivering the ACKs. This is due to the gateway deployment. Indeed, in urban scenario all gateways are closer to the end-devices than in the other two scenarios. For instance, we found that in urban scenario, the Alg-HR has led to a reduction in the delay of about 50% compared to Alg-LB algorithm.

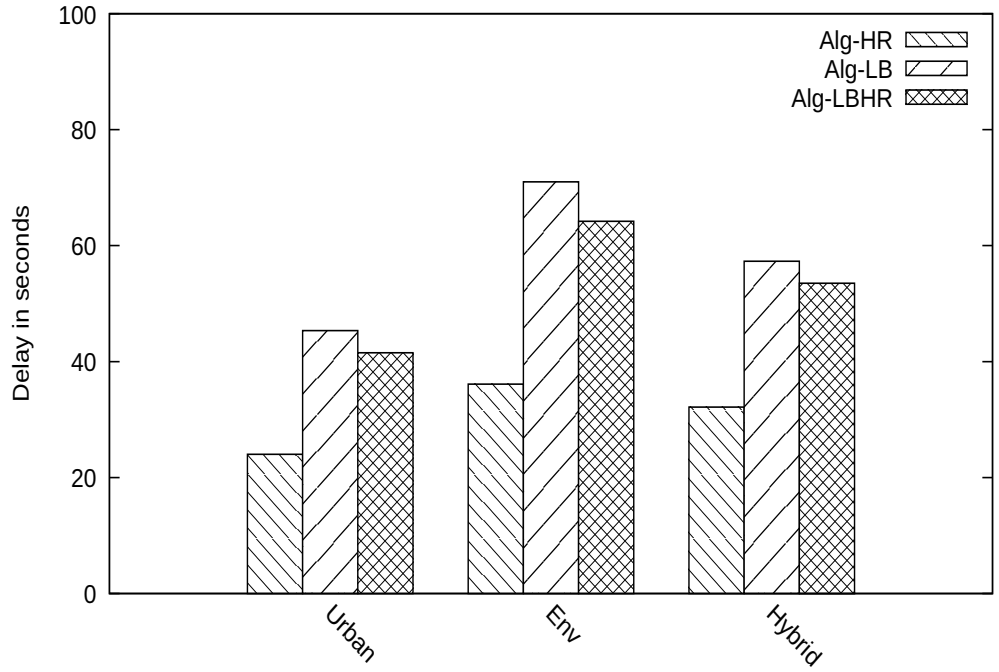


Figure 6.10: Average delay with $N = 100$ end-devices and $g = 4$ gateways for the three algorithms in each scenario.

Figure 6.11 shows the average delay for ACKs delivery according to the number of gateways with $N = 100$ end-devices for ENV scenario. We can see that the delay computed decreases when the number of gateways increases in the network. Hence, adding more gateways reduces the average distance, therefore the SF is reduced which reduces the delay. Consequently, the more gateways LoRaWAN has, the less delay the data frames will experience. For instance, we found that 8 gateways bring a gain of 50% compared to 2 gateways when using Alg-HR algorithm. Indeed, when adding more gateways in the network, the distance between the end-device and the gateway is shortened, which leads

to shorten the time on air of the frames.

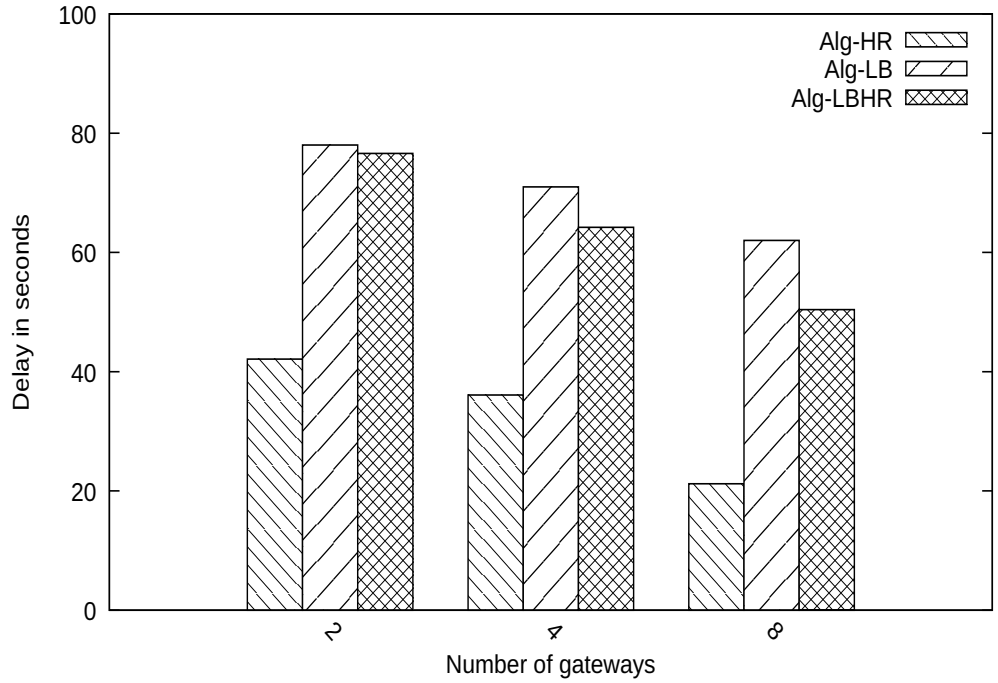


Figure 6.11: Average delay according to the number of gateways with $N = 100$ end-devices for ENV scenario.

6.4.5 Energy consumption

LoRaWAN end-devices have a limited power supply because each device is equipped with an attached battery. In most situations, they are deployed in a hostile environment where it is hard to change or charge their battery. Hence, the energy consumption is an important metric in the performance evaluation of a LoRaWAN network. Indeed, the network lifetime is regarded as a fundamental parameter in the context of availability in LoRaWAN, and hence it should be considered when deploying a LoRaWAN network.

Several factors can be a source of energy over-consumption: state of the radio operator module (awake state: sending or listening, and sleep state), retransmissions, collisions, the device position (end-device or gateway), the time on air for the frame transmission, mobility, etc. The network topology represents also a major factor on energy consumption

as it can impact the devices batteries. Indeed, a transmission requires a higher power if the distance between the end-device and the gateway is high. This leads to increase the energy consumption of the end-device. In other words, a transmission on a short distance consumes less energy (for the end-device). In this subsection, we focus on the impact of the topology (i.e. the scenario of gateway deployment) on energy consumption of the end-devices, and we determine the most energy efficient topology for the network lifetime of LoRaWAN.

In this regard, more simulations were run to study the evolution of the energy consumption of the end-devices in the three aforementioned scenarios in order to see the impact of the topology. Also, we examine the energy consumption in each of the three previous algorithms.

Figure 6.12 shows the average energy consumption per end-device for each algorithm in each scenario. First, we can notice that Alg-HR is the best compared to Alg-LB and Alg-LBHR. This is because in Alg-HR, the end-device is associated to the nearest gateway. In other words, the end-device is associated to the gateway that is the closest. Thus, the energy consumption decreases further because the time on air of frames decreases with the distance. It can also be noticed that the less time the end-device is in the active state, the lower its consumed energy. Here we can see the dependency between the energy consumption and the transfer delay. The less consumed energy needed to transmit data is related to the low delay.

Moreover, we can see that the urban scenario is the best in terms of energy consumption compared to the other two scenarios. In other words, the urban scenario gives the lowest consumed energy. This is due to the gateway deployment, since in urban scenario, all gateways are closer to the end-devices than in the other two scenarios, which yields to a reduction in the energy consumption compared to other scenarios.

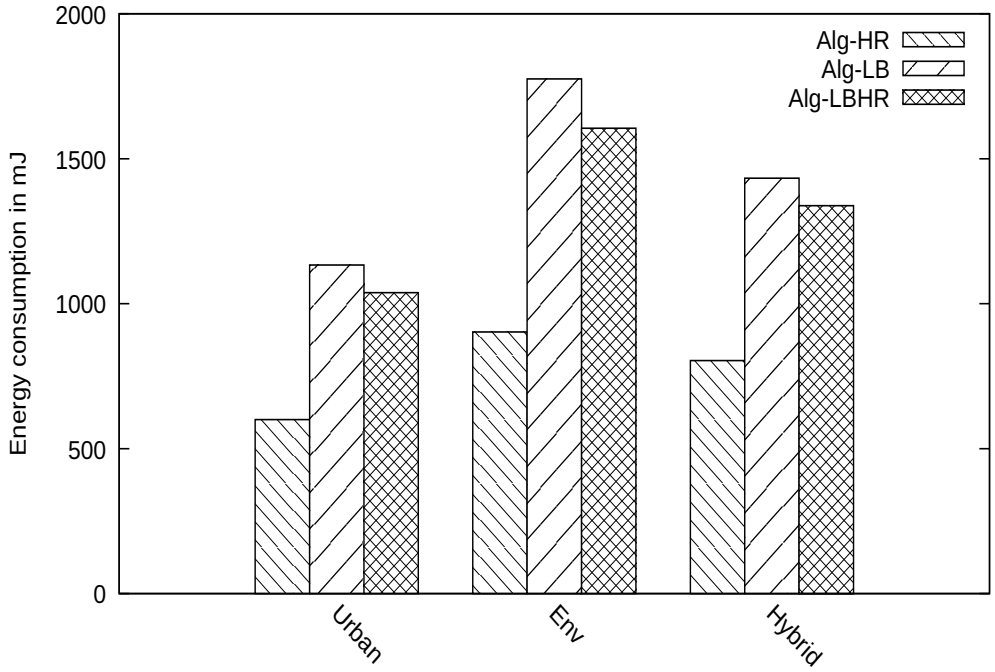


Figure 6.12: Average energy consumption per end-device with $N = 100$ end-devices and $g = 4$ gateways for the three algorithms in each scenario.

Figure 6.13 shows the average energy consumption per end-device according to the number of gateways with $N = 100$ end-devices for ENV scenario. We can see that when increasing the number of gateways in the network, the energy consumption decreases further. As the distance between the end-device and the gateway decreases, the time on air of the frame decreases as well, which leads to a reduction in energy consumption. Moreover, we can notice that with the increase of the number of gateways, the difference between Alg-LB and Alg-LBHR becomes more significant.

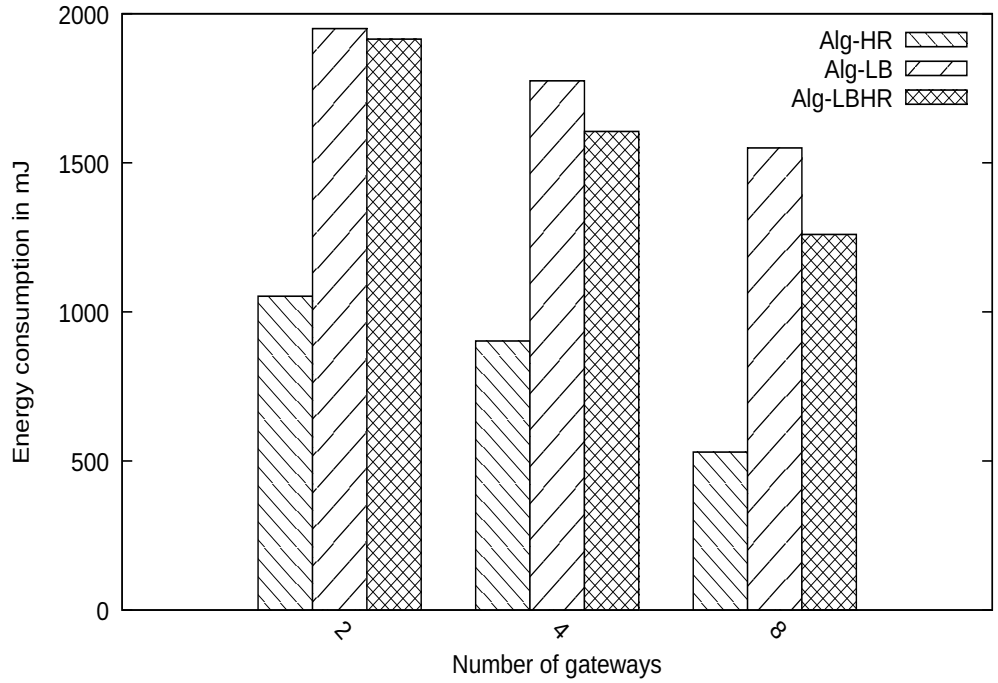


Figure 6.13: Average energy consumption per end-device according to the number of gateways with $N = 100$ *end-devices for ENV scenario.*

6.5 Conclusion

In this chapter, we presented and evaluated three algorithms for selecting the best gateway for downlink communications while increasing LoRaWAN throughput for three types of gateway deployments, as well as decreasing the energy consumption. First, we showed that the system throughput depends on this deployment. For instance, ENV and hybrid scenarios have the worst throughput compared to urban scenario. Second, we showed that balancing the load per gateway improves the throughput compared to the increase in the signal quality. Third, we showed that combining both the load and the signal quality does not improve further the throughput. Fourth, we showed that the energy consumption depends on the gateway deployment. For instance, the urban scenario has the lowest energy consumption compared to ENV and hybrid scenarios. In contrast to what we found for the throughput, we showed that choosing the gateway with the highest

signal quality reduces the energy consumption compared to choosing the gateway with the lowest load. This is because when the coverage area increases, the energy consumption also increases, due to the distance between the end-devices and the gateway. We have compared the three scenarios of gateway deployment, and showed their influence on the delay of the frames. We found that choosing the shortest gateway is much better than choosing the gateway with the lowest load.

Furthermore, we studied the number of frame collisions and ACK receptions that might arise under heavy load of end-devices, and explored the impact of the number of gateways on LoRaWAN network. In order to maximize the utilization of LoRaWAN while increasing the throughput, parameters such as the number of end-devices, the number of gateways, the scenario of gateway deployment, and the algorithm for gateway selection should be known in advance. Thus, the combination of these four parameters determines the LoRaWAN throughput for downlink communications.

Moreover, it is worth mentioning that there is a compromise between increasing the throughput and decreasing the energy consumption. In other words, it falls back to the application and its needs to choose which metric matters the most for it.

Part V

CONCLUSION AND PERSPECTIVES

Evaluating the protocols developed for IoT networks is essential before such networks can be deployed. This evaluation is necessary both to ensure that the protocol works correctly in different applications and scenarios, but also and above all to measure its performance, to compare it with other existing protocols and to improve its mechanisms and functionalities.

In this manuscript, after an introduction to the context of IoT, we have identified the directives of the existing standards, the parameters which have a direct impact on performance, and the dimensioning of these technologies. As the study concerns the LoRaWAN protocol in Europe (868 MHz), a comparison was made between this protocol and the other LPWAN protocols currently used by industrialists and researchers. Particular attention was paid to the characteristics and mechanisms which are used by each of these protocols. It is clear from the study presented in the state of art that LoRaWAN is suitable as an energy consumption technology in many possible applications. LoRaWAN is capable of providing good energy efficiency and good radio range.

Simulations were used to measure the performance of LoRaWAN. They are necessary to evaluate hypotheses or new concepts. In this thesis, it is a question of studying uplink and downlink LoRaWAN communications; it involves studying complex phenomena such as frame collisions which are evaluated using our developed simulator. In this context, to assess the robustness and performance of LoRaWAN, an interference simulation model has been developed. A model was built in the Java environment which allows to simulate the MAC layer. It allows to estimate the number of frame collisions and frame losses in LoRaWAN and the resulting degradations in the overall performance. The simulations carried out in this work deal in particular with interference in bidirectional-communications, allowing the interaction of end-devices with autonomous batteries.

Moreover, in this thesis, an analysis of the physical properties and the system behavior of LoRaWAN in a constrained environment with interferences was carried out thanks, on the one hand, to a comparative study with the other competing protocols, and on the other hand, to simulation tests. Following this analysis, the parameters that can be improved have been identified and proposals for improvements have been made. They relate, among other things, to media access mechanisms, and the possibility of implementing new MAC protocol enabling new features to be offered such as frame recovery after collisions.

Contributions

Context of Efficient Decoding of Synchronized Colliding LoRa Signals

Our first contribution [51] is based on proposing an algorithm which decodes synchronized colliding frames in LoRa while improving the overall performance in the network. In Chapter 5, we have seen that collisions in LoRaWAN are damaging to the overall network performance. Indeed, collision is a factor that negatively impacts LoRaWAN throughput, which is already very limited (between 250 and 11000 bps). When a gateway receives several superposed LoRa signals with similar receive power levels, on the same channel and with the same spreading factor, LoRaWAN is unable to decode these signals which are hence lost.

In this thesis, we have proposed in Chapter 5 a collision resolution algorithm at the physical layer that enables to decode synchronized colliding frames in LoRa while mitigating the harmful effects of collisions (i.e frame retransmissions and losses). We also proposed a MAC algorithm in order to synchronize end-devices and to retransmit bitmaps in reply to guesses from the gateway instead of retransmitting whole frames. Based on our simulation results, we showed that our proposed MAC algorithm is able to significantly improve the network performance, in terms of system throughput and energy consumption. In addition, our proposed protocol enables significant delay reductions needed to successfully decode the colliding frames.

We have seen that the percentage of frame losses is less drastic using our MAC protocol. This is due to the fact that with our proposed decoding technique (i.e. the transmissions of bitmaps instead of complete frames), our MAC protocol is able to reduce the number of collisions, and hence the number of retransmissions with losses. In addition, the delay in both versions of LoRaWAN is greater than the delay in our protocol, which leads to further decrease the LoRaWAN throughput. Consequently, our MAC algorithm enables better collision decoding since it corresponds to sending short bitmaps.

These results contributed to the development of a new MAC protocol based on LoRaWAN, relying on the proposed collision resolution algorithm, and surpassing LoRaWAN.

Context of the Gateway Selection for Downlink Communication in LoRaWAN
Our second contribution [52] is based on selecting the best gateway among all candidates when replying in donwlink communications to each end-device. In Chapter 6, we study the selection of the gateway by the network server in order to reply to the end-device in downlink communications.

In Chapter 6, we have presented and evaluated three algorithms for selecting the best gateway for downlink communications while increasing LoRaWAN performance for three types of gateway deployment, as well as decreasing the energy consumption.

We have presented simulation results along with statistics based on a data-set containing all frames sent by end-devices in uplink communications to the network server. We evaluated the scalability, throughput, delay and energy consumption of LoRaWAN deployments by defining metrics such as the number of collided frames and the number of received ACK frames.

First, we showed that the system throughput depends on the gateway deployment. For instance, the environmental and hybrid scenarios have the worst throughput compared to urban scenario. Second, we showed that balancing the load per gateway improves the throughput compared to the increase in the signal quality. Third, we showed that combining both the load and the signal quality does not improve further the throughput.

Additionally, we showed that the energy consumption depends on the gateway deployment. For instance, the urban scenario has the lowest energy consumption compared to the environmental and hybrid scenarios. In contrast to what we found for the throughput, we showed that choosing the gateway with the highest signal quality reduces the energy consumption compared to choosing the gateway with the lowest load. This is because when the coverage area increases, the energy consumption also increases, due to the distance between the end-devices and the gateway. We have compared the three scenarios of gateway deployment, and showed their influence on the delay of the frames. We found that choosing the closest gateway is much better than choosing the gateway with the lowest load.

Furthermore, we studied the number of frame collisions and ACK receptions that might arise under heavy load of end-devices, and explored the impact of the number of gateways on LoRaWAN network. In order to maximize the utilization of LoRaWAN while increasing the throughput, parameters such as the number of end-devices, the number of

gateways, the scenario of gateway deployment, and the algorithm for gateway selection should be known in advance. Thus, the combination of these four parameters determines the LoRaWAN throughput for downlink communications. In all cases, we have noticed that increasing the number of gateways is a good solution for increasing the network performance. For example, the increase in the number of gateways decreases the load per gateway which leads to lower the frame collisions in downlink communications.

In our case study, we showed for instance that adding more gateways is very beneficial for improving LoRaWAN performance. Having less gateways in the environment on a small or large scale certainly constitutes a real challenge, but if these challenges are met, solutions are available to develop new implementations and enrich the offer of LoRaWANs in the IoT market. In addition, interesting solutions to use would be, for example, the use of bitmaps for the collision of synchronized LoRa frames. These two solutions would both extend the scope of LoRa in the IoT.

Moreover, it is worth mentioning that there is a compromise between increasing the throughput and decreasing the energy consumption. In other words, it falls back to the application and its needs to choose which metric matters the most for it.

Perspectives

At the end of the presentation of this work, we would like to present some research points which seem interesting to us to study in the future.

Short-term perspectives

During this thesis, we implemented using the Java programming language [125] all the functions necessary to make simulations and to test our hypotheses. One of the perspectives is to use a network simulator like NS-3 [129] to examine our work and contributions. We could thus obtain for example the accurate delay of the successful reception of the frames by the receiver in a more realistic simulation model.

In addition, another perspective in the context of the gateway selection for downlink communication in LoRaWAN is to propose an optimal algorithm based on Integer Linear Programming (ILP), and a heuristic based on our analyses in order to find the optimal gateway placement. Furthermore, retransmissions of lost frames can also be taken into

account in our simulations.

Moreover, it is worth considering real experiments to make real tests and implementations of our hypotheses on physical LoRa devices. Experiments are useful for examining the validity of our assumptions and for determining the effectiveness of our previously mentioned contributions.

Long-term perspectives

The perspectives of this thesis may include the simulation of interferences caused with other LPWANs, protocols and radio transceivers to further determine the radio robustness of LoRaWAN. Another perspective would be to use the models developed, in particular the adaptive data rate (ADR) model, to know the best parameter selection of LoRa modules from which the performance of the overall system can be improved. In addition, other time-related techniques and new MAC layer mechanisms might be simulated to overcome any persistent interference problems. With ADR, it is also possible to control the consumption of end-devices in typical activity scenarios for LoRaWAN autonomous battery-powered devices. Finally, it would be interesting to investigate (by simulation, radio test and / or probabilistic calculations) the possibilities for improving LoRaWAN in order to assess its relevance and overall impact on the system.

Last but not least, as many IoT applications (for example in healthcare, monitor patient, tracking objects, etc.) require mobile end-devices, where the mobility is a major parameter to consider, it would be interesting to study the impacts of mobility (low and high speed mobility) of LoRa end-devices on the overall system performance. Noting that several requirements are considered essential for LoRaWAN mobile networks such as the throughput, the maintenance of connectivity, and the energy consumption of end-devices. Moreover, data gathering from mobile sensors is more challenging than data gathering from static sensors. Hence, it is interesting to propose approaches that collect the environment conditions, and then adapt the end-device settings to reach better performance in mobile LoRa networks, and to support mobility throughout the system.

Finally, the study of LoRaWAN class B, which allows downlink communications with a limited delay, arouses interest. Therefore, we aim to test and analyze the class B specification in order to seek its limitations and improve them.

Bibliography

[1] M. Stočes, J. Masner, E. Kánská, J. Jarolímek, *et al.*, "Processing of Big Data in Internet of Things and Precision Agriculture.," in *Agrarian Perspectives XXVII. Food Safety-Food Security, Proceedings of the 27th International Scientific Conference, 19-20 September 2018, Prague, Czech Republic*, pp. 353–358, Czech University of Life Sciences Prague, Faculty of Economics and Management, 2018.

[2] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: anomaly detection of IoT cyberattacks in smart city using machine learning," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0305–0310, IEEE, 2019.

[3] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, pp. 1–35, 2019.

[4] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of IEEE 802.15. 4 using measurements and simulations," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, vol. 1, pp. 487–492, IEEE, 2006.

[5] "N. Sornin, M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM), and O. Hersent (Actility)," *LoRaWAN Specification V1*, 2015.

[6] A. Tiurlikova, N. Stepanov, and K. Mikhaylov, "Method of assigning spreading factor to improve the scalability of the LoRaWAN wide area network," in *2018 10th*

International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 1–4, IEEE, 2018.

[7] E. Alsaadi and A. Tubaishat, "Internet of Things: features, challenges, and vulnerabilities," *International Journal of Advanced Computer Science and Information Technology*, vol. 4, no. 1, pp. 1–13, 2015.

[8] M. Bagheri and S. H. Movahed, "The effect of the Internet of Things (IoT) on education business model," in *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 435–441, IEEE, 2016.

[9] M. Pooja, M. M. Singh, and M. K. Gandhi, "Interconnected Smart Objects: Era of Internet of Things," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 3, no. 6, 2014.

[10] D. Infso, "Internet of Things in 2020: A Roadmap for the Future, INFSO D. 4 Networked Enterprise & RFID and INFSO G. 2 Micro & Nanosystems in co-operation with RFID Working Group of the European Technology Platform on Smart Systems Integration (EPOSS)," in *Tech. Rep.*, 2008.

[11] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, and M. Eisenhauer, "Internet of Things strategic research roadmap," *Internet of things-global technological and societal trends*, vol. 1, no. 2011, pp. 9–52, 2011.

[12] S. Talari, M. Shafie-Khah, P. Siano, V. Loia, A. Tommasetti, and J. Catalão, "A review of smart cities based on the Internet of Things concept," *Energies*, vol. 10, no. 4, p. 421, 2017.

[13] A. Rghioui, S. Sendra, J. Lloret, and A. Oumnad, "Internet of Things for measuring human activities in ambient assisted living and e-health," *Network Protocols and Algorithms*, vol. 8, no. 3, pp. 15–28, 2016.

[14] J. Lloret, J. Tomas, A. Canovas, and L. Parra, "An integrated IoT architecture for smart metering," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 50–57, 2016.

[15] D. Ismail, M. Rahman, and A. Saifullah, "Low-power wide-area networks: Opportunities, challenges, and directions," in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, p. 8, ACM, 2018.

[16] L. Petter and N. Johanna, "A Study of Low-Power Wide-Area Networks and an In-Depth Study of the LoRaWAN Standard." Available: http://www.diva-portal.org/smash/get/diva2:1141920/FULLTEXT01.pdf, 2017.

[17] J. Finnegan and S. Brown, "A comparative survey of LPWA networking," *arXiv preprint arXiv:1802.04222*, 2018.

[18] R. Want, B. N. Schilit, and S. Jenson, "Enabling the Internet of Things," *Computer*, no. 1, pp. 28–35, 2015.

[19] E. Ruano Lin, "LoRa protocol. Evaluations, limitations and practical test," Universitat Politècnica de Catalunya, 2016.

[20] J. Chase, "The Evolution of the Internet of Things," *Texas Instruments*, vol. 1, pp. 1–7, 2013.

[21] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[22] WiFi, [Online]. Available: http://www.ieee802.org/11/.

[23] A. I. Al-Alawi, "WiFi technology: Future market challenges and opportunities," *Journal of computer Science*, vol. 2, no. 1, pp. 13–18, 2006.

[24] A. Powell, "WiFi publics: producing community and technology," *Information, communication & society*, vol. 11, no. 8, pp. 1068–1088, 2008.

[25] K. Pothuganti and A. Chitneni, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *Advance in Electronic and Electric Engineering*, vol. 4, no. 6, pp. 655–662, 2014.

[26] O. Horyachyy, "Comparison of Wireless Communication Technologies used in a Smart Home: Analysis of wireless sensor node based on Arduino in home automation scenario." [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1118965/FULLTEXT02.pdf, 2017.

[27] J. Rossey, I. Moerman, P. Demeester, and J. Hoebeke, "Wi-Fi helping out Bluetooth Smart for an improved home automation user experience," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, pp. 1–6, IEEE, 2016.

[28] M. D. Corner, B. N. Levine, O. Ismail, and A. Upreti, "Advertising-based measurement: A platform of 7 billion mobile devices," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pp. 435–447, 2017.

[29] P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "CADWAN: A control architecture for dense WiFi access networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 194–201, 2018.

[30] Bluetooth, [Online]. Available: http://www.bluetooth.com.

[31] P. Bhagwat, "Bluetooth: technology for short-range wireless apps," *IEEE Internet Computing*, vol. 5, no. 3, pp. 96–103, 2001.

[32] E. Mackensen, M. Lai, and T. M. Wendt, "Bluetooth Low Energy (BLE) based wireless sensors," in *SENSORS, 2012 IEEE*, pp. 1–4, IEEE, 2012.

[33] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15. 4," *Sensor network operations*, vol. 4, pp. 218–237, 2006.

[34] I. Howitt and J. A. Gutierrez, "IEEE 802.15. 4 low rate-wireless personal area network coexistence issues," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 3, pp. 1481–1486, IEEE, 2003.

[35] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, M. Naeve, B. Heile, and V. Bahl, "Home networking with IEEE 802.15. 4: a developing standard for low-rate wireless personal area networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 70–77, 2002.

[36] S. Ahamed, "The role of ZigBee technology in future data communication system.," *Journal of Theoretical & Applied Information Technology*, vol. 5, no. 2, 2009.

[37] V. P. Rao, "The simulative investigation of Zigbee/IEEE 802.15. 4," *Department of Electrical Engineering And Information Technology Dresden University of Technology. Master of Science*, 2005.

[38] S. C. Ergen, "ZigBee/IEEE 802.15. 4 Summary," *UC Berkeley, September, Technical report*, vol. 10, p. 17, 2004.

[39] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on ZigBee technology," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, pp. 297–301, IEEE, 2011.

[40] A. Hussain and M. Ivanovic, *Electronics, Communications and Networks IV: Proceedings of the 4th International Conference on Electronics, Communications and Networks (CECNET IV), Beijing, China, 1215 December 2014*. CRC Press, 2015.

[41] S. Kim and Y. Yoo, "Contention-aware adaptive data rate for throughput optimization in LoRaWAN," *Sensors*, vol. 18, no. 6, p. 1716, 2018.

[42] N. El Rachkidy, A. Guitton, and M. Kaneko, "Collision Resolution Protocol for Delay and Energy Efficient LoRa Networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 535–551, 2019.

[43] R. Trüb and L. Thiele, "Increasing Throughput and Efficiency of LoRaWAN Class A," in *UBICOMM 2018. The Twelfth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 54–64, International Academy, Resarch, and Industry Association (IARIA), 2018.

[44] P. J. Marcelis, V. S. Rao, and R. V. Prasad, "DaRe: Data recovery through application layer coding for LoRaWAN," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 97–108, IEEE, 2017.

[45] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, p. 2104, 2018.

[46] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, p. 2364, 2017.

[47] D. F. Carvalho, P. Ferrari, E. Sisinni, A. Depari, S. Rinaldi, M. Pasetti, and D. Silva, "A test methodology for evaluating architectural delays of LoRaWAN implementations," *Pervasive and Mobile Computing*, vol. 56, pp. 1–17, 2019.

[48] D. F. Carvalho, P. Ferrari, A. Flammini, and E. Sisinni, "A test bench for evaluating communication delays in LoRaWAN applications," in *2018 Workshop on Metrology for Industry 4.0 and IoT*, pp. 248–253, IEEE, 2018.

[49] D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "On the evaluation of application level delays in public LoRaWAN networks," in *2019 IEEE International Symposium on Measurements & Networking (M&N)*, pp. 1–6, IEEE, 2019.

[50] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of latency and MAC-layer performance for class a LoRaWAN," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, 2017.

[51] S. Abboud, N. E. Rachkidy, and A. Guitton, "Efficient Decoding of Synchronized Colliding LoRa Signals," pp. 1–5, IEEE Vehicular Technology Conference (VTC) 2020.

[52] S. Abboud, N. El Rachkidy, A. Guitton, and H. Safa, "Gateway selection for downlink communication in LoRaWAN," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2019.

[53] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.

[54] O. Khutsoane, B. Isong, and A. M. Abu-Mahfouz, "IoT devices and applications based on LoRa/LoRaWAN," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6107–6112, IEEE, 2017.

[55] C. Goursaud and J.-M. Gorce, "Dedicated networks for IoT: PHY/MAC state of the art and challenges," *EAI endorsed transactions on Internet of Things*, 2015.

[56] SIGFOX, Accessed: Aug. 6, 2017. http://www.sigfox.com.

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

[58] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, "DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication," in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 54–59, IEEE, 2015.

[59] WEIGHTLESS, [Online]. Available: http://www.weightless.org.

[60] INGENU, [Online]. Available: https://www.ingenu.com/technology/rpma.

[61] NGMN, [Online]. Available: http://www.ngmn.org.

[62] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.

[63] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.

[64] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[65] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range IoT technologies: The dawn of LoRa^{TM}," in *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pp. 51–58, Springer, 2015.

[66] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[67] L. García-García, J. M. Jiménez, M. T. A. Abdullah, and J. Lloret, "Wireless technologies for IoT in smart cities," *Network Protocols and Algorithms*, vol. 10, no. 1, pp. 23–64, 2018.

[68] "Libelium Comunicaciones Distribuidas, Waspmote Sigfox NetworkingGuide," [Online]. Available: www.libelium.com, 1 2015, v4.1.

[69] 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.

[70] U. Noreen, A. Bounceur, and L. Clavier, "A study of LoRa low power and wide area network technology," in *Advanced Technologies for Signal and Image Processing (ATSIP), 2017 International Conference on*, pp. 1–6, IEEE, 2017.

[71] M. Bor, J. E. Vidler, and U. Roedig, "LoRa for the internet of things," 2016.

[72] Semtech, "AN1200.22 LoRa™ Modulation Basics," [Online]. Available: https://www.semtech.com/uploads/documents/an1200.22.pdf.

[73] 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.

[74] A. Jebril, A. Sali, A. Ismail, and M. Rasid, "Overcoming limitations of LoRa physical layer in image transmission," *Sensors*, vol. 18, no. 10, p. 3257, 2018.

[75] Semtech, "SX1272/3/6/7/8: LoRa Modem Designer's Guide, Corporation Advanced Communications and Sensing Products Division, 200 Flynn Road, Camarillo, CA93012, 2013a. AN1200.13.," [Online]. Available: https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/knowledge-items/application-notes-for-the-internet-of-things/LoRa%20Design%20Guide.pdf.

[76] G. Ferré and A. Giremus, "LoRa Physical Layer Principle and Performance Analysis," in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 65–68, IEEE, 2018.

[77] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, "Improving LoRa network capacity using multiple spreading factor configurations," in *2018 25th International Conference on Telecommunications (ICT)*, pp. 516–520, IEEE, 2018.

[78] LoRa Alliance, "LoRa alliance wide area networks for IoT," *LoRa Alliance [Online]. Available: https://www. lora-alliance. org/[Accessed: 06 February 2016]*, 2016.

[79] N. Blenn and F. Kuipers, "LoRaWAN in the wild: Measurements from the things network," *arXiv preprint arXiv:1706.03086*, 2017.

[80] M. Pulpito, P. Fornarelli, C. Pomo, P. Boccadoro, and L. A. Grieco, "On fast prototyping LoRaWAN: a cheap and open platform for daily experiments," *IET Wireless Sensor Systems*, vol. 8, no. 5, pp. 237–245, 2018.

[81] K. Kousias, G. Caso, Ö. Alay, and F. Lemic, "Empirical Analysis of LoRaWAN Adaptive Data Rate for Mobile Internet of Things Applications," in *Proceedings of the 2019 on Wireless of the Students, by the Students, and for the Students Workshop*, pp. 9–11, ACM, 2019.

[82] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in LoRa networks," *arXiv preprint arXiv:1611.00688*, 2016.

[83] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 59–67, ACM, 2016.

[84] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *2016 International Conference on Engineering and Telecommunication (EnT)*, pp. 10–14, IEEE, 2016.

[85] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of LoRaWAN channel access with capture effect," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, IEEE, 2017.

[86] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *European Wireless 2016; 22th European Wireless Conference*, pp. 1–6, VDE, 2016.

[87] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, 2017.

[88] A. Rahmadhani and F. Kuipers, "When LoRaWAN Frames Collide," 2018. https://fernandokuipers.nl/papers/WiNTECH2018.pdf.

[89] A. Lavric and V. Popa, "Performance evaluation of LoRaWAN communication scalability in large-scale wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[90] G. Ferre, "Collision and packet loss analysis in a LoRaWAN network," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2586–2590, IEEE, 2017.

[91] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN-design, analysis, and deployment," *Sensors*, vol. 19, no. 4, p. 838, 2019.

[92] T.-H. To and A. Duda, "Simulation of lora in ns-3: Improving lora performance with csma," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2018.

[93] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop LoRa networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21430–21446, 2017.

[94] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 309–321, 2017.

[95] N. El Rachkidy, A. Guitton, and M. Kaneko, "Decoding superposed LoRa signals," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pp. 184–190, IEEE, 2018.

[96] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–7, IEEE, 2018.

[97] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[98] K. Mikhaylov, J. Petäjäjärvi, and A. Pouttu, "Effect of downlink traffic on performance of LoRaWAN LPWA networks: Empirical study," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2018.

[99] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, ieee, 2017.

[100] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.

[101] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

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

[103] Semtech, https://semtech.my.salesforce.com/sfc/p/E0000000JelG/a/44000000MDnR/Et1KWLCuNDI6MDagfSPAvqqp.Y869Flgs1LleWyfjDY.

[104] M. Afshang and H. S. Dhillon, "Poisson cluster process based analysis of HetNets with correlated user and base station locations," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2417–2431, 2018.

[105] J.-B. Park and K. S. Kim, "Load-Balancing Scheme With Small-Cell Cooperation for Clustered Heterogeneous Cellular Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 633–649, 2018.

[106] A. K. Gupta, H. S. Dhillon, S. Vishwanath, and J. G. Andrews, "Downlink multi-antenna heterogeneous cellular network with load balancing," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 4052–4067, 2014.

[107] Y. Sun, W. Xia, S. Zhang, Y. Wu, T. Wang, and Y. Fang, "Energy Efficient Pico Cell Range Expansion and Density Joint Optimization for Heterogeneous Networks with eICIC," *Sensors*, vol. 18, no. 3, p. 762, 2018.

[108] C. Gomez, A. Shami, and X. Wang, "Machine Learning Aided Scheme for Load Balancing in Dense IoT Networks," *Sensors*, vol. 18, no. 11, p. 3779, 2018.

[109] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless communications*, vol. 18, no. 3, pp. 10–21, 2011.

[110] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in loRaWAN," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2019.

[111] J. So, D. Kim, H. Kim, H. Lee, and S. Park, "LoRaCloud: LoRa platform on openstack," in *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, pp. 431–434, IEEE, 2016.

[112] P. Jörke, S. Böcker, F. Liedmann, and C. Wietfeld, "Urban channel models for smart city IoT-networks based on empirical measurements of LoRa-links at 433 and 868 MHz," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2017.

[113] V. A. Dambal, S. Mohadikar, A. Kumbhar, and I. Guvenc, "Improving LoRa signal coverage in urban and sub-urban environments with uavs," in *2019 International Workshop on Antenna Technology (iWAT)*, pp. 210–213, IEEE, 2019.

[114] S. Bertoldo, L. Carosso, E. Marchetta, M. Paredes, P. Savi, and M. Allegretti, "A LoRa-based distributed network for environmental monitoring using public transport," 2018.

[115] N. Harris, J. Curry, *et al.*, "Development and range testing of a LoRaWAN system in an urban environment," *International Journal of Electronics and Communication Engineering*, vol. 12, no. 1, 2018.

[116] O. Iova, A. Murphy, G. P. Picco, L. Ghiro, D. Molteni, F. Ossi, and F. Cagnacci, "LoRa from the city to the mountains: Exploration of hardware and environmental factors," 2017.

[117] M. Luvisotto, F. Tramarin, L. Vangelista, and S. Vitturi, "On the use of LoRaWAN for indoor industrial IoT applications," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[118] P. Romero-Diaz, L. Garcia, S. Sendra, and J. Navarro-Ortiz, "Performance of LoRaWAN Networks in Outdoor Scenarios," in *Proc. of the 18th Int. Conf. on Networks (ICN 2019)*, pp. 1–6, 2019.

[119] H. Tarab, "Real Time Performance Testing of LoRa-LPWAN Based Environmental Monitoring UAV System," 2018.

[120] Bouleogeorgos, Alexandros-Apostolos A and Diamantoulakis, Panagiotis D and Karagiannidis, George K, "Low power wide area networks (LPWANs) for internet of things (IoT) applications: Research challenges and future trends," *arXiv preprint arXiv:1611.07449*, 2016.

[121] F. Yu, Z. Zhu, and Z. Fan, "Study on the feasibility of LoRaWAN for smart city applications," in *2017 IEEE 13th International conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 334–340, IEEE, 2017.

[122] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, "Design and Implementation of Open LoRa for IoT," *IEEE Access*, vol. 7, pp. 100649–100657, 2019.

[123] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, and N. Koteli, "IoT agriculture system based on LoRaWAN," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–4, IEEE, 2018.

[124] M. Kouvatsou, "LoRa protocol analysis and performance evaluation using PyCom equipment," Master's thesis, Πανεπιστήμιο Πειραιώς, 2019.

[125] J. Gosling, B. Joy, G. Steele, and G. Bracha, *The Java language specification*. Addison-Wesley Professional, 2000.

[126] Y. Okumura, "Field strength and its variability in VHF and UHF land-mobile radio service," *Rev. Electr. Commun. Lab.*, vol. 16, pp. 825–873, 1968.

[127] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, 1980.

[128] "SX1272/73—860 MHz to 1020 MHz Low Power Long Range Transceiver," [Online]. Available: http://www.semtech.com/images/datasheet/sx1272.pdf (accessed on 22 May 2017).

[129] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*, pp. 15–34, Springer, 2010.