Algorithms for Reliability Insurance and Traffic Differentiation in Low Power Internet of Things Communication Networks

Alessandro Aimi

HESAM UNIVERSITÉ

le cnam

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
Centre d'études et de recherche en informatique et communications

THÈSE

présentée par : **Alessandro AIMI**

soutenue le : **19 septembre 2023**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée au : **Conservatoire national des arts et métiers**

Discipline : **Informatique**

Algorithmes pour l'Assurance de la Fiabilité et la Différenciation du Trafic dans les Réseaux de Communication de l'Internet des Objets à Faible Consommation d'Énergie

Algorithms for Reliability Insurance and Traffic Differentiation in Low Power Internet of Things Communication Networks

THÈSE dirigée par :
M. SECCI Stefano Professeur, Cnam

et co-encadrée par :
M. ROVEDAKIS Stéphane Maître de conférences, Cnam
M. GUILLEMIN Fabrice Resp. programme de recherche (HDR), Orange Innovation

Jury

M^{me} Francesca CUOMO	Prof., DIET, Sapienza Univ. di Roma	Président
M. Paolo BELLAVISTA	Prof., DISI, Univ. di Bologna	Rapporteur
M^{me} Katia JAFFRÈS-RUNSER	Prof., IRIT, Univ. de Toulouse	Rapporteuse
M. Yassine HADJADJ-AOUL	Prof., IRISA/Inria, Univ. de Rennes	Examinateur
M^{me} Oana IOVA	MCF, CITI, INSA Lyon	Examinatrice
M^{me} Christelle CAILLOUET	MCF, I3S/Inria, Univ. Côte d'Azur	Invitée

THÈSE

... And I said, "Oh, yeah? Wait 'til you hear these ones" ...

The Burning Hell, Don't Believe the Hyperreal

Acknowledgements

First of all, I would like to address my thesis supervisors: Prof. Stefano Secci, Fabrice Guillemin, Stéphane Rovedakis and - even if only for the first year of this journey - Amina Boubendir. Thank you for giving me the opportunity to pursue this research endeavour, for your trust and patience, and for your invaluable lessons on methodology, technical rigor and scientific integrity; they will forever be parts of my personal and professional background. The amount of respect and gratitude I have for you is difficult to express with words.

Next, I would like to thank Prof. Paolo Bellavista and Prof. Katia Jaffrés-Runser for the time and effort invested in the review and evaluation of this manuscript. I also would like to thank Prof. Francesca Cuomo, Oana Iova and Prof. Yassine Hadjadj-Aoul for accepting to be part of my evaluation committee as examiners, and Christelle Caillouet for accepting to join as invited member.

I express my gratitude to Jean-François Maudoux, head of the TEAM équipe at Orange Innovation, who was also responsible for giving me this opportunity. You have welcomed and supported me since the first day, always setting the best standards of fairness and kindness for the team. I would also like to thank Aroussia Maadi, director of the Automation For Network department, and Nabil Charkani El Hassani, previously director of the Network Architecture & Automation department, for believing in me and providing financial support to share my work with the scientific community around the world.

Many thanks as well to all the researchers in the ANR INTELLIGENTSIA project, for the fruitful exchanges that ultimately led to the development of some of the contributions present in this manuscript.

Much gratitude goes to all the past and present members of the équipe TEAM and the people gravitating around it. Three years ago I arrived in France alone, and, since then, you have welcomed me as a second family; thank you Christian Destré, David Mathieu, Michel Ribeyron, Ilhem Fajjari,

Frederic Desnoes, Sihem Cherrared, Nour El Houda Nouar, Arzhang Shabazi, Timam Ghosh, Abderaouf Khichane, José Jurandir Alves Esteves, Wassim Aroui, Anping Zhao, Samuel Bertrand Dalechamps, Théo Claudien, Vitor Kussler Veronese, Sonia Boumodine, Sebastien Trancoso, Nedir Boukerras, Mohamed Ryad Cherifi, Mohamed Larbi Halet, Youssra Abrouk El Alami, Edgar Oblette and Salim Abdelmadjid Mariche. Same goes for all the members of the ROC team of the Cèdric laboratory of CNAM; therein, a special thought goes out to Maria Isabella Viola, Chi-Dung Phung, Nour El Houda Yellas, Mario Patetta and Mustapha Kamal Benramdane.

Now, to my friends. During these years, your presence helped overcome the toughest moments, and made good times memorable. For this, I am deeply grateful to Annie Gerard, Laiza de Lara Alves, Andrea Giuseppe Tommasi and Matteo Bernabè.

I thank my family - Sabrina Stecconi, Severino Aimi, Francesco Aimi and Grace - for their love and endless support in my pursuits.

Last but not least, I thank Chiara Colagiacomo for being the best thing to happen in my life.

Abstract

The ever-growing presence of the Internet of Things (IoT), characterized by a plethora of independent technologies at the network edge, calls for a seamless integration in the context of beyond 5G transformations. Novel principles and technologies such as softwarization and virtualization, multi-access edge computing, and artificial intelligence pave the way for the broader paradigms of network slicing and zero-touch management.

On the one hand, network slicing postulates the partition of networks into either partially or fully isolated horizontal slices for on-demand offering of specific services with guaranteed Service Level Agreements (SLAs). On the other hand, zero-touch management proposes achieving full automation with the introduction of smart loops for all support and maintenance operations to maintain SLAs.

Among the IoT technologies, Long Range Wide Area Networks (LoRaWANs) are rapidly gaining popularity thanks to their cheap and easy-to-operate nature. LoRaWAN is a low power wide area network technology targeting long range sensing and monitoring. Use cases are typically in the domain of agriculture (cattle tracking, soil properties monitoring) and smart cities (bicycle pool management, garbage monitoring, air quality monitoring, etc.)

One of the design choices in LoRaWAN is the uncoordinated access to the radio medium. This aims to minimize the power consumption of intermittently active devices by simplifying the communication protocol stack. Consequently, collisions are frequent in dense deployments, resulting in best effort operation, inherently inefficient utilization of resources and lack of quality. A key performance metric for an application using LoRaWAN is the Packet Delivery Ratio (PDR).

In this thesis we consider open issues regarding the integration of LoRaWAN in a network slicing and zero-touch management framework. LoRaWAN introduces several challenges in the radio access network: limited action in partitioning radio resources, sparse device control opportunities, and no

realistic requirement specification that can be integrated in SLAs to enable network slicing. In our contributions, we first investigate the introduction of PDR-based traffic quality differentiation in LoRaWAN, showing its feasibility and potential appeal in dense scenarios. Based on our findings, we proceed to improve our resource allocation with an online technique for PDR management which is both adaptive to changes and drastically more efficient. Finally, we capitalize on our extensive simulation efforts to produce an open-source tool to enable high-fidelity, end-to-end emulation of massive-IoT LoRaWAN services on real-world infrastructures.

Keywords: Internet of Things (IoT), LoRaWAN, Quality of Service (QoS), Reliable data delivery, Network management, Beyond 5G networks.

Résumé

La présence toujours croissante de l'Internet des Objets (IdO), caractérisée par une pléthore de technologies indépendantes à la périphérie du réseau, appelle à une intégration transparente dans le contexte des transformations au-delà de la 5G. De nouveaux principes et technologies tels que la softwarisation et la virtualisation, le multi-access edge computing et l'intelligence artificielle ouvrent la voie à des paradigmes plus larges de "slicing" du réseau et de gestion "zero-touch".

D'une part, le "slicing" du réseau postule la partition des réseaux en tranches horizontales plus ou moins isolées pour l'offre à la demande de services spécifiques avec des accords de niveau de service (SLAs) garantis. D'autre part, la gestion "zero-touch" propose une automatisation complète avec l'introduction de boucles intelligentes pour toutes les opérations de support et de maintenance afin de maintenir les SLAs.

Parmi les technologies IdO, les réseaux étendus à longue portée (LoRaWANs) gagnent rapidement en popularité grâce à leur nature bon marché et facile à utiliser. LoRaWAN est une technologie de réseau étendu à faible consommation qui vise la détection et la surveillance à longue distance : les cas d'utilisation se situent généralement dans le domaine de l'agriculture (suivi du bétail, surveillance des propriétés du sol) et des villes intelligentes (gestion des parcs à vélos, surveillance des ordures, surveillance de la qualité de l'air, etc.)

L'un des choix de conception de LoRaWAN est l'accès non coordonné au support radio. L'objectif est de réduire au minimum la consommation d'énergie des appareils à activité intermittente en simplifiant la pile de protocoles de communication. Par conséquent, les collisions sont fréquentes dans les déploiements denses et se traduisent par une opération "best effort", une utilisation intrinsèquement inefficace des ressources et un manque de qualité. Une mesure de performance clé pour une application utilisant LoRaWAN est le taux de livraison de paquets (PDR).

Dans cette thèse, nous examinons les questions ouvertes concernant l'intégration de LoRaWAN dans un cadre de découpage du réseau et de gestion "zero-touch". LoRaWAN introduit un certain nombre de défis dans le réseau d'accès radio : une action limitée dans le partitionnement des ressources radio, des opportunités de contrôle des appareils peu nombreuses, et aucune spécification d'exigence réaliste pouvant être intégrée dans les SLAs pour permettre le "slicing" du réseau. Dans nos contributions, nous étudions tout d'abord l'introduction de la différenciation de la qualité du trafic basée sur le PDR dans le réseau LoRaWAN, en montrant sa faisabilité et son intérêt potentiel dans des scénarios denses. Sur la base de nos résultats, nous améliorons notre allocation de ressources avec une technique en ligne pour la gestion du PDR qui s'adapte aux changements et qui est beaucoup plus efficace. Enfin, nous capitalisons sur nos vastes efforts de simulation pour produire un outil open-source permettant l'émulation haute fidélité et de bout en bout des services LoRaWAN ultra-denses sur des infrastructures du monde réel.

Mots-clés : Internet des Objets (IdO), LoRaWAN, Qualité De Service (QDS), Transmission fiable des données, Gestion du réseau, Réseau 5G et au-delà.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Context and Current Trends in IoT

Nowadays, novel paradigms and technologies such as softwarization and virtualization, Multi-access Edge Computing (MEC) and massive Internet of Things (IoT), automated management and orchestration, are enabling an evolutionary push in the design of communication network infrastructures towards "beyond 5G" and 6G architectures [1, 2, 3]. These envisioned transformations in networks are conceived to meet the changes and demands identified in most social and industrial use cases such as: the availability of on-demand services, connectivity and long-term computing capacities, smart loops for all support and maintenance operations, etc. This calls for the careful evaluation of the mentioned technologies and their integration, given the deep impact they will most likely have in the overall 6G system. In fact, some of them are still on a relatively low Technology Readiness Level (TRL) and will have complex interactions with the others.

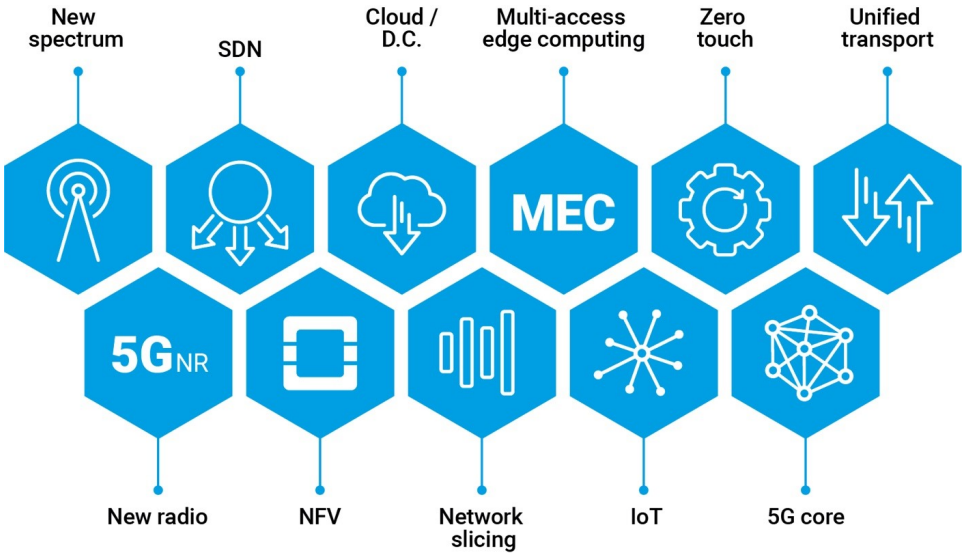


Figure 1.1: Evolutionary elements in the 5G network architecture [4, 5]. Acronyms and abbreviations: Software Defined Networking (SDN), Network Function Virtualization (NFV), Distributed Cloud (D.C.), MEC, IoT.

As shown in Figure 1.1, among 5G principles and technologies take place the broader paradigms of network slicing and zero-touch management. On the one hand, network slicing postulates the partition of networks into more or less isolated horizontal slices for on-demand offering of specific services with guaranteed Service Level Agreements (SLAs) [6]. On the other hand, zero-touch management proposes to achieve full automation with the introduction of smart loops for all support and maintenance

operations [7]. Together, they are designed to help network operators, service providers and end-users reach enhanced levels of efficiency, flexibility, automation, security and overall Quality of Service (QoS).

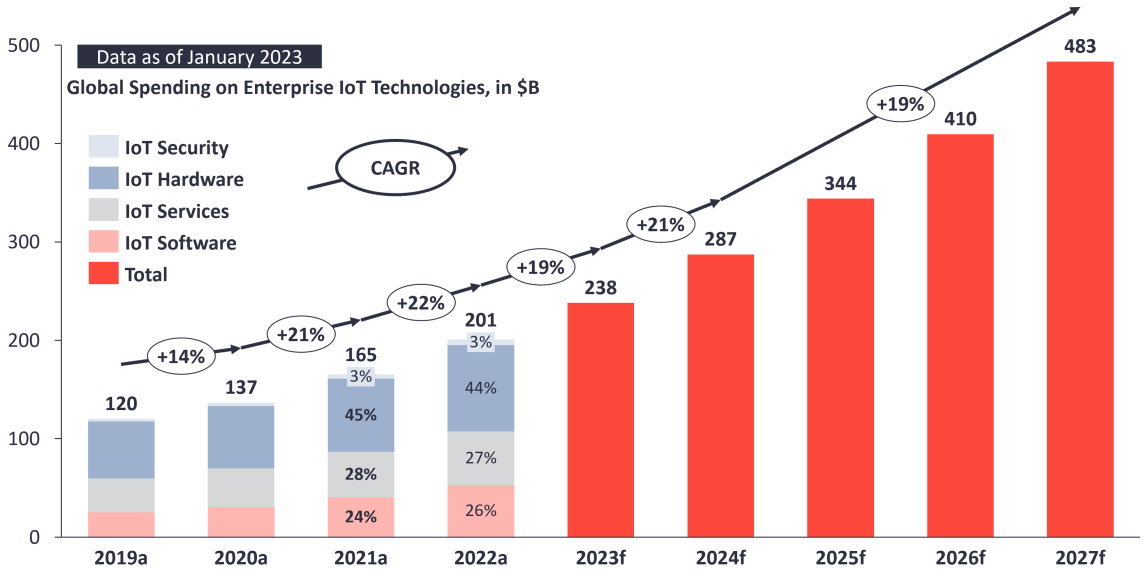


Figure 1.2: Enterprise IoT market 2019-2027 [8]. Yearly data is either actual (a) or a forecast (f). Market evolution over the years is measured in Compound Annual Growth Rate (CAGR). In the scope of the depicted market research, IoT is defined as a network of internet-enabled physical objects. Objects that become internet-enabled (IoT devices) typically interact via embedded systems, some form of network communication, or a combination of edge and cloud computing. The data from IoT-connected devices is often used to create novel end-user applications. Connected personal computers, tablets, and smartphones are not considered IoT, although these may be part of the solution setup. Devices connected via extremely simple connectivity methods, such as radio frequency identification or quick response codes, are not considered IoT devices.

One of the network segments whose evolution has seen a fast growth in number are IoT devices, leading to the coinage of the term massive IoT to highlight the impact they will have on the network Edge. As shown in Figure 1.2, the enterprise IoT market share has been steadily growing in recent years, and multiple market research sources forecast the possibility of an exponential trend in the years to come [9, 10, 11]. The emerging presence of the IoT is characterized by a plethora of independent technologies at the network Edge, some of which reached the market without necessarily complying with the global vision for 5G networks and beyond proposed by 3GPP, ETSI and other standardization bodies. Thus, a coherent evolution of communication networks requires a seamless integration of the IoT in the direction of beyond 5G transformations.

Among the IoT technologies, Long Range Wide Area Networks (LoRaWANs) have gained popularity thanks to their cheap and easy-to-operate nature [12]. LoRaWAN is a *non-3GPP* Low Power Wide Area Network (LPWAN) technology targeting long range sensing and monitoring: use cases are typically in the domain of agriculture (e.g., cattle tracking, soil properties monitoring, etc.) and smart cities (e.g., bicycle pool management, garbage monitoring, air quality monitoring, etc.) [13]. As shown in Figure 1.3, LoRaWAN currently represents a significant portion of existing LPWAN deployments.

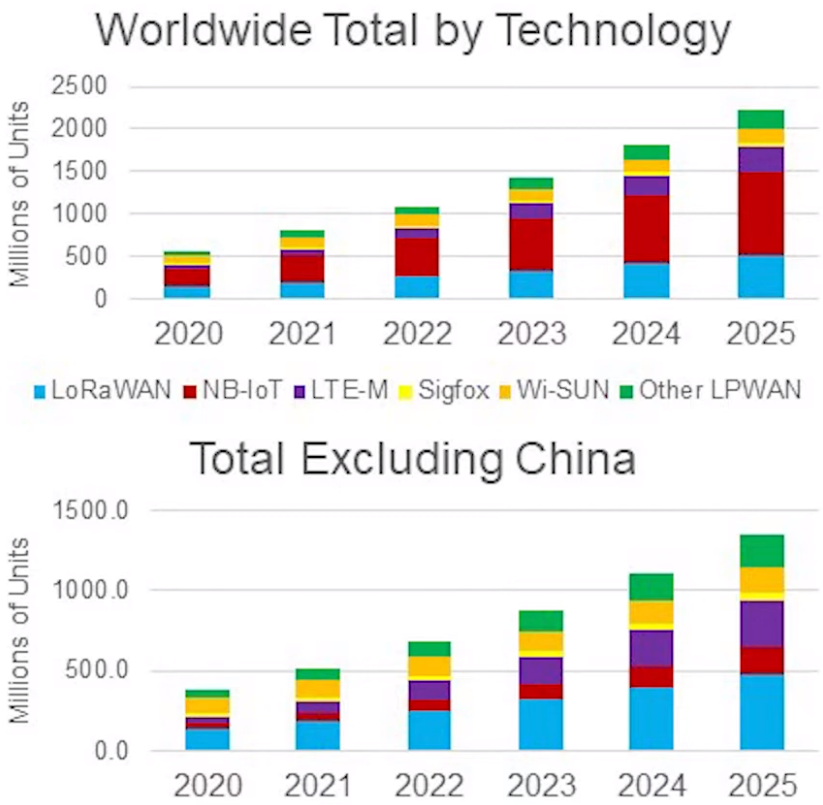


Figure 1.3: Yearly evolution in the number of installed LPWAN base units by technology [14]. Actual data for 2020, forecasts up to 2025. A Narrowband IoT (NB-IoT) market distortion is present due to Chinese government initiatives. Outside of China, LoRaWAN accounts for the largest share of installed units.

One of the design choices in LoRaWAN is the uncoordinated access to the radio medium. For example, in Europe only a simple duty-cycle limitation is present, establishing the average percentage of time that each device can spend occupying the channel. This is aimed at minimizing the power consumption of intermittently active devices by simplifying the communication protocol stack. As a consequence, collisions become more frequent in dense deployments and result in a best effort operation

and an inherently inefficient utilization of resources [15]. For this reason, a key performance metric for an application using LoRaWAN is the Packet Delivery Ratio (PDR) [16], measuring link reliability as the number of received packets divided by the number of sent packets over a period of time.

Given the current popularity of LoRaWAN and the expected growth of the IoT in the years to come, it is relevant to investigate how the technology could be integrated in the ecosystem evolution of 5G networks and beyond. In particular, the 5G Machine Type Communications (MTC) use cases that are currently covered by LoRaWAN often have different quality requirements, especially in terms of packet reliability (think, for instance, a fire alarm vs. a smart dishwasher vs. a humidity sensor in a smart city context [17, 18]). While quality differentiation in 5G is addressed by means of network slicing and automated resource allocation, LoRaWAN still manages traffic in a best effort way, providing no guarantees in term of reliability. To this day, the integration of LoRaWAN in the management automation framework envisioned for 5G MTC technologies remains an open research issue.

1.2 Problem Statement and Challenges

Network slicing as envisioned in 5G requires the partitioning of resources to create more or less isolated end-to-end (E2E) traffic flows. A notable challenge in network slicing is the idea to introduce full automation in the resource allocation process: operators seal with each slice client a SLA containing the desired traffic quality metrics, so that network orchestration components can automatically adapt resources to meet the requirements defined inside the agreement. While research on the topic has seen advancements for the 5G core network thanks to the advent of virtual and programmable network functions, it is still a mostly uncovered problem for what concerns the Radio Access Network (RAN) [3].

In this thesis, we consider open issues regarding the integration of LoRaWAN in a network slicing and automated management framework. Special attention is given to the problem of isolating groups of IoT devices and enforcing differentiated reliability requirements through resource allocation strategies. Existing research on LoRaWAN slicing concentrates on the RAN. In fact, for the most part, LoRaWAN core functions natively integrate the virtualization, programmability, and observability required by network slicing. On the contrary, a number of challenges appear when the RAN is considered: no realistic requirement specification that can be integrated in SLAs, limited action in partitioning radio

resources and sparse device control opportunities.

Related work on RAN traffic quality differentiation introduces QoS requirements only as optimization guidelines in the radio resource allocation process; a way to enforce strict quality requirements, and especially reliability in the form of PDR, is still missing in the traffic differentiation literature. Given the binding nature of SLAs in network slicing, this point is a key enabling aspect and it will be a central challenge considered in our contributions.

LoRaWAN operates on unlicensed bands, and devices generally share all the available bandwidth randomly choosing one of the available frequency channels at each transmissions. The limited number of independent discrete channels reduces the partitioning granularity of radio resources. As a consequence, isolating groups of devices is not as straightforward as it could be in other radio technologies, especially when introducing fairness considerations as commonly done in network slicing.

Finally, in LoRaWAN use cases, end-devices transmit sparse upstream packets in a periodical or sporadic fashion. In their most common operating mode, devices only admit one downstream configuration opportunity per upstream transmission. As a consequence, the introduction of smart feedback loops for the RAN automation as envisioned by zero-touch management requires careful dimensioning to account for the technical and physical limitations present in LoRaWANs.

1.3 Research Questions and Contributions

From the state of the art analysis carried out in this thesis and summarized in the above section the following research questions were produced:

1. Can LoRaWAN traffic differentiation be achieved with strict quality requirements?
2. Is it economically viable for LoRaWAN operators and users to adopt strict quality requirements?
3. Which approaches to adopt to go towards automated quality management in LoRaWAN?

Question 1 is addressed by proposing a first algorithmic approach that (i) allocates channels to classes of devices according to their total load and desired PDR requirement, and (ii) introduces a form of capacity-based access control to limit interference and reach the PDR objectives. Moreover, a variant with relaxed isolation among classes is provided. Results are produced using a well known

interference model in a dense network, and they show that our approach is able to satisfy the minimum PDR targets unlike state of the art proposals. The drawback of this approach is a reduction in partial maximum cell range, as the worse interferers happen to be the farthest devices. These results are presented in Chapter 4 and were also published in the following full conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Packet delivery ratio guarantees for differentiated LoRaWAN services," in *Proc. 2022 IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 2014–2019, doi: 10.1109/GLOBECOM48099.2022.10001145.

Question 2 is studied by testing a wider range of allocation strategies and introducing metrics for the user utility and the operator gain measured in terms of resource efficiency. Three modular techniques for channel allocation to classes are proposed, spanning over the fairness spectrum. Moreover, aside access control, a second traffic control variant is proposed using existing LoRaWAN primitives to control the duty-cycle of devices. Simulation results are carried-out in a denser urban environment, and they show that once again PDR targets are met. However, measuring the allocation efficiency leads to the conclusion that, for medium-low device densities, our proposals are less efficient than the state of the art without traffic control. Results show that this can be attributed to two main factors: (i) the limited amount of hardware resources in LoRaWAN access points affecting the isolation normally provided by different channels, and (ii) the adoption of a conservative mathematical model to produce capacity values used in the allocation, oftentimes leading the techniques to surpass the required PDR. These results are presented in Chapter 5 and were also published in the following full conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Traffic control and channel assignment for quality differentiation in dense urban LoRaWANs," in *Proc. 20th Int. Symp. Model. and Optim. in Mobile, Ad hoc, and Wireless Netw. (WiOpt)*, 2022, pp. 153–160, doi: 10.23919/WiOpt56218.2022.9930551.

Our third contribution aims both at solving the inefficiencies found throughout the investigation of Question 2, and at proposing a first approach to quality management automation as mentioned in Question 3. A feedback loop is introduced to dynamically configure the duty-cycle of devices according to the measured PDR. This allows the algorithm to "learn" the channel capacity instead of using a

static mathematical model. This estimation procedure is based on the well known bisection method to reduce at a minimum the number of configurations to be sent to devices. Our results show that the algorithm converges to the desired PDR value with a maximum overshoot of 2%. In a high density setting converges to the target in a maximum of 80 hours, and afterwards it is able to self-heal when faced with isolated changes in the pool of connected devices. These results are presented in Chapter 6 and will be submitted for publication in a journal article once joint with results of the algorithm running on a proof-of-concept network.

Finally, to capitalize on the considerable validation, implementation and code optimization efforts that were put into simulating LoRaWAN, *ELoRa* is proposed, an open source tool developed to emulate and study E2E massive IoT infrastructures. Thanks to an accurate implementation of packet serialization and protocol, simulations of the RAN running in real-time can be connected to real deployments of the LoRaWAN server stack. Traffic is bidirectionally translated between simulation and reality. The server stack is agnostic to the emulation, and simulated devices support all reconfiguration primitives that can be sent downstream. Two main use cases are identified for the tool: creating realistic anomalies to test core orchestration techniques, and testing radio resource allocation algorithms implemented in a real LoRaWAN server stack. *ELoRa* is presented in Chapter 7 and was published in the following demo conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "ELoRa: End-to-end emulation of massive IoT LoRaWAN infrastructures," in *Proc. 2023 IEEE/IFIP Netw. Operations and Manage. Symp. (NOMS)*, 2023, pp. 1–3, doi: 10.1109/NOMS56928.2023.10154373.

1.4 Document Structure

The chapters of this thesis are organized as follows. This introductory chapter provides an overview of the technological context surrounding the scope of the thesis, motivating the identified problems. After presenting a summary of the main challenges, the contributions of this thesis and the work framework are summarized.

Chapter 2 provides a background overview of the LoRaWAN technology, detailing all aspects needed to understand the chapters that follow. The network architecture, physical layer, Medium Access Control (MAC) protocol and device management actions are presented, together with some

considerations on current management practices in LoRaWANs.

Chapter 3 details the related work on the introduction of service differentiation and slicing in LoRaWANs. First, the general principles and motivations of the network slicing paradigm are introduced. Then, perspectives and challenges related to the integration of slicing in LoRaWANs are discussed. Finally, the state of the art on the topic of traffic quality differentiation in the radio access network is surveyed and compared.

Afterwards, the contributions summarized in Section 1.3 are presented as follows: Chapter 4 introduces a first approach to traffic quality differentiation with guarantees; Chapter 5 carries out a broader efficiency study based on multiple algorithms proposed for traffic control and channel assignment; Chapter 6 outlines a novel technique for adaptive packet delivery ratio management based on online estimation and duty-cycle control; Chapter 7 presents *ELoRa*, an emulator to test complex management actions for massive IoT services in real-world infrastructures. Finally, Chapter 8 concludes the dissertation with perspectives for future work improvements of the presented solutions.

Chapter 2

Long Range Wide Area Networks

Contents

2.1 Introduction

The IoT has steadily been growing for the past few decades and its spread becomes now exponential [10]. IoT technologies have a wide range of applications in the domains of wearables, smart home, office, city, industry, and agriculture. In general, IoT applications require low-cost, low-energy, reliable and, given the expected densification of connected devices, scalable network technologies [19]. Some IoT use cases require energy-efficient, long range wireless communications at low bandwidth/speeds; devices can then be battery powered to remove the costs introduced by the deployment and long term management of cables.

In this context, LPWANs have recently been hot topics for the research community. This class of technologies targets long range sensing and monitoring. Use cases are typically in the domain of agriculture, industry and smart cities [20]: they comprise, for instance, smart irrigation, wind monitoring, soil moisture monitoring, crop management, livestock monitoring in agriculture, street lighting, water level monitoring, road signalization, air quality monitoring, public transport and waste management in smart cities [18].

Among LPWAN technologies, LoRaWANs have rapidly gained popularity thanks to their cheap and easy-to-operate nature [12]. Business opportunities have emerged for new and existing network operators, interested in understanding its operational limits. Compared with NB-IoT, its main LPWAN competitor for the targeted use cases, LoRaWAN (i) operates on unlicensed sub-gigahertz radio frequency bands, contributing to its affordability and flexibility, (ii) does not require SIM cards and heavy authentication procedures, and (iii) consequently adopts a much simpler MAC layer, aimed at fully maximizing battery lifetime of intermittently and sporadically active devices [20]. In practice, LoRaWAN devices are highly energy efficient, at the cost of a more relaxed link layer reliability [21].

Technically speaking, LoRaWAN is an open standard promoted by the LoRa Alliance [22]; it defines a MAC and network management protocol for IoT wide area networks on top of the Long Range (LoRa) radio physical layer (often called LoRa PHY), which is proprietary to Semtech. Despite this distinction, the term LoRaWAN - not followed by the word 'protocol' - is commonly used to refer to the overall technology as an ensemble of the two layers.

In this chapter we give an overview of the LoRaWAN technology, with the intention to serve as preliminary background for the resource management techniques of following chapters. Section 2.2

presents its network architecture, followed by its physical and MAC layers in Sections 2.3 and 2.4, respectively. Then, Section 2.5 discusses management practices commonly adopted in LoRaWANs.

2.2 Network Architecture

The LoRaWAN architecture is represented in Figure 2.1, and the protocol layers of LoRaWAN communications are shown in Figure 2.2 on the next page. The radio access network can be described as a star of stars, where the sub-stars are composed of low-power IoT devices broadcasting messages through the radio medium to LoRaWAN gateways, while the core star consists of a LoRaWaN Network Server (LNS) function connected to multiple gateways through internet as a backhaul link.

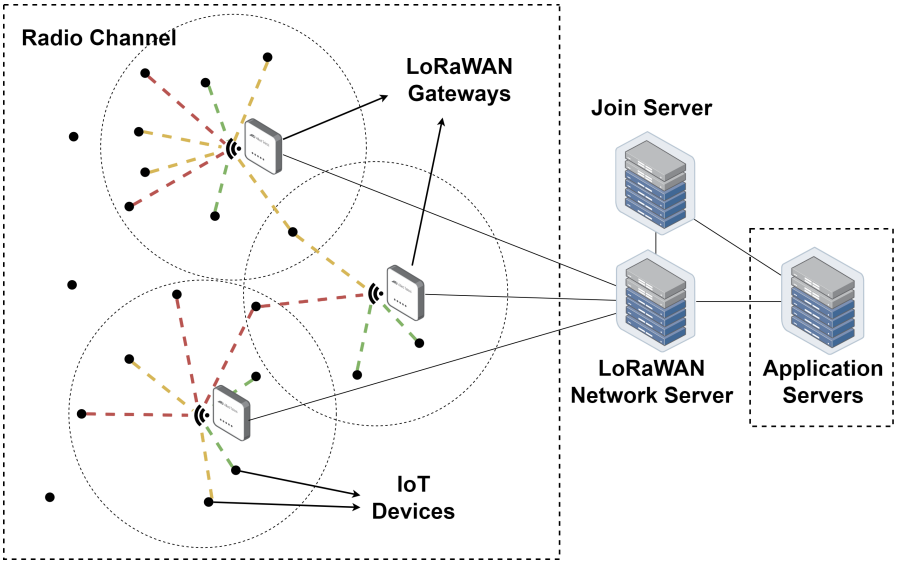


Figure 2.1: Overview of the LoRaWAN architecture.

IoT devices implementing the LoRaWAN MAC protocol generally consist of a light microprocessor (for example, an Arduino board) equipped with a sensor (humidity, pollution, heat, movement, etc.), and a LoRa chip for wireless communication. Transmission parameters are managed by the LNS using the MAC protocol so, theoretically, battery powered devices can last years without manual maintenance. In LoRaWAN, devices generate a light load of sparse or periodical traffic, which rarely exceeds 50 bytes per payload, with a periodicity lower than 10 minutes [24].

LoRaWAN gateways are in charge of forwarding frames between the devices and the LNS. They achieve access point radio modulation/demodulation of LoRa transmissions, and generally have

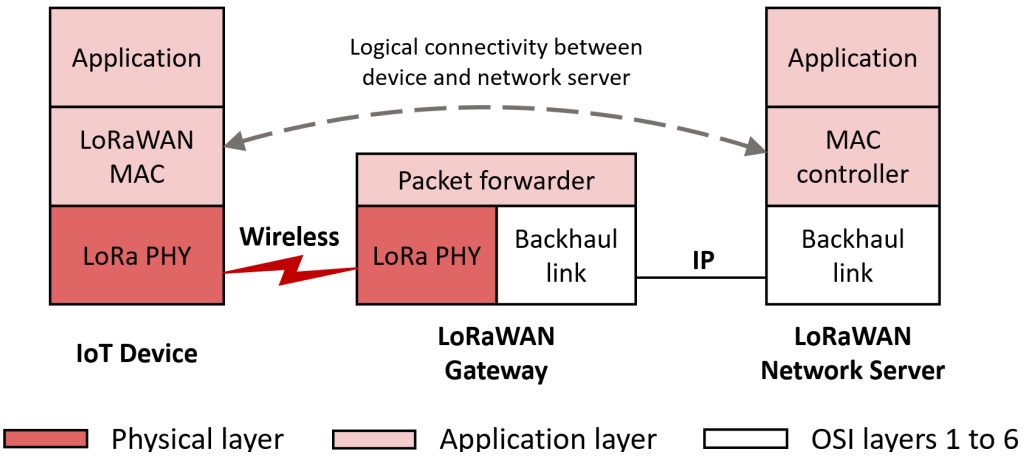


Figure 2.2: Protocol layers in LoRaWAN communications [23].

many parallel demodulation paths (8 in Semtech's SX1301 chips used for LoRaWAN outdoor macro-gateways [25]) always listening to the radio interface. Contrary to early beliefs [26], each demodulation path **cannot** be locked on a single central frequency, but it cycles over all the available ones configured on a gateway [25]. The maximum theoretical throughput that can be received by a single 8-path gateway from the radio interface is of 40.8 $^{\text{kbit}}/_{\text{s}}$ [1]. Gateways are connected to the LNS using IP, either via Ethernet, WiFi or 3G/4G/LTE, and they encapsulate LoRaWAN frames as payload of application protocols (Semtech UDP Packet Forwarder [28], MQTT [29, 30]) working at OSI layer 7. A common practice is to implement LoRaWAN gateways using a microcomputer, as for instance a Raspberry Pi.

The LNS is the network function in charge of managing the LoRaWAN radio access network. It validates authenticity and integrity of frames, manages duplication when the same transmission is received at multiple gateways, optimizes radio parameters of devices using downlink frames, and selects the best gateway for downlink transmission. The payload of LoRaWAN frames containing the application data is then sent to the correct Application Server for data decryption and end user exploitation. Finally, the Join Server is an optional function in charge of providing session keys for devices using over-the-air registration procedure. Most widely used implementations of the server stack functions are already containerized and cloud-native.

[1] Computed considering the unlikely scenario of multiple devices creating 8 perfect streams of consecutive packets on different frequencies using spreading factor 7, 230 B MAC payloads, 8 preamble symbols, 4/5 coding rate, explicit header and CRC (formula in [27]).

2.3 LoRa Physical Layer

A relevant aspect of the LoRa physical layer is its radio modulation technique, a derivation of the Chirp Spread Spectrum (CSS) technology. A LoRa transmission is a stream of symbols, called *chirps*: each chirp is a linear frequency-modulated sinusoidal pulse (see Figure 2.3) of time duration T_c and fixed bandwidth $B = f_1 - f_0$, with f_1 and f_0 frequency values such that $f_1 > f_0$. By varying the chirp duration, quasi-orthogonal signals, acting as virtual channel, can be created. In addition, the chirp duration leads to a trade-off between the throughput and the robustness against noise and interference.

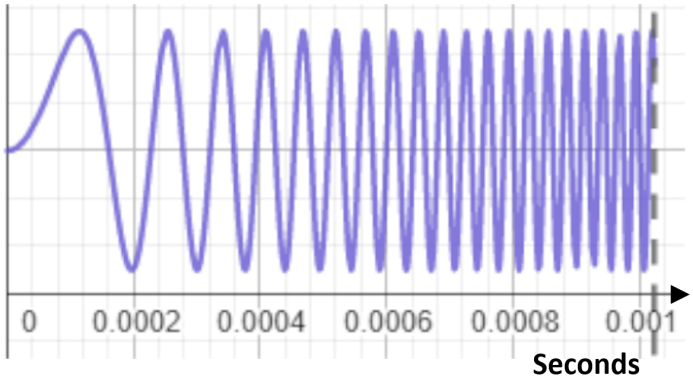


Figure 2.3: Being a derivation of Chirp Spread Spectrum modulation, LoRa symbols are so called *chirps*. The signal of a chirp is a linear frequency-modulated sinusoidal pulse. If the frequency is increasing they are called *upchirps*, otherwise *downchirps*. In the image: amplitude graph of an upchirp.

For a fixed T_c, the data symbols are coded by unique instantaneous frequency trajectory as shown in Figure 2.4 on the next page. These trajectories are obtained by cyclically shifting a reference chirp, in a manner not too dissimilar to dialing numbers in an old rotary dial telephone. These cyclic shifts, representing symbols, are discretized into multiples of chip-time $T_{\text{chip}} = 1/B$, while only 2^j possible edges in the instantaneous frequency exist. Therefore, each chirp represents j bits where j is referred to as SF. As a result, the modulation signal $m(t)$ of nth LoRa symbol can be expressed as:

$$m(t) = \begin{cases} f_1 + k \cdot (t - nT_{\text{chip}}) & \text{if } \ 0 \leq t \leq nT_{\text{chip}} \\ f_0 + k \cdot (t - nT_{\text{chip}}) & \text{if } \ nT_{\text{chip}} < t \leq T_c \end{cases} \tag{2.1}$$

where $k = (f_1 - f_0)/T_c$ is the rate of frequency increase over symbol duration $T_c = 2^{SF}/B$. Additional examples of how different values of n are encoded to symbols with the LoRa modulation are provided

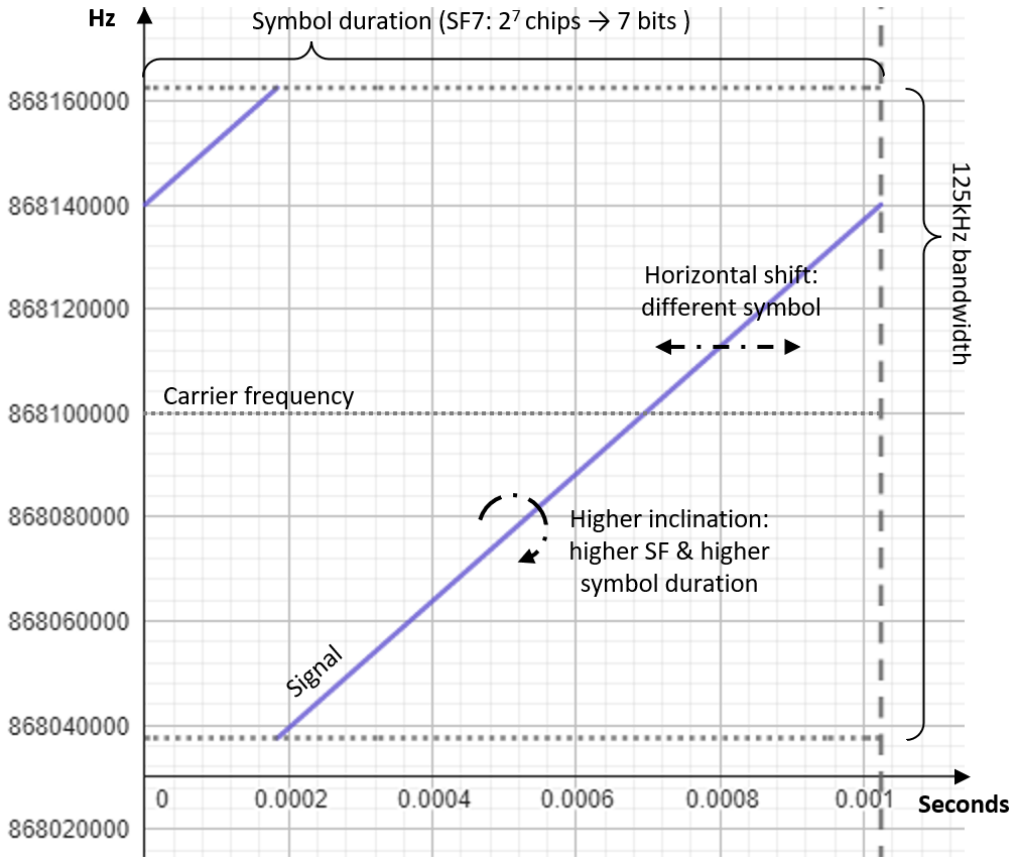
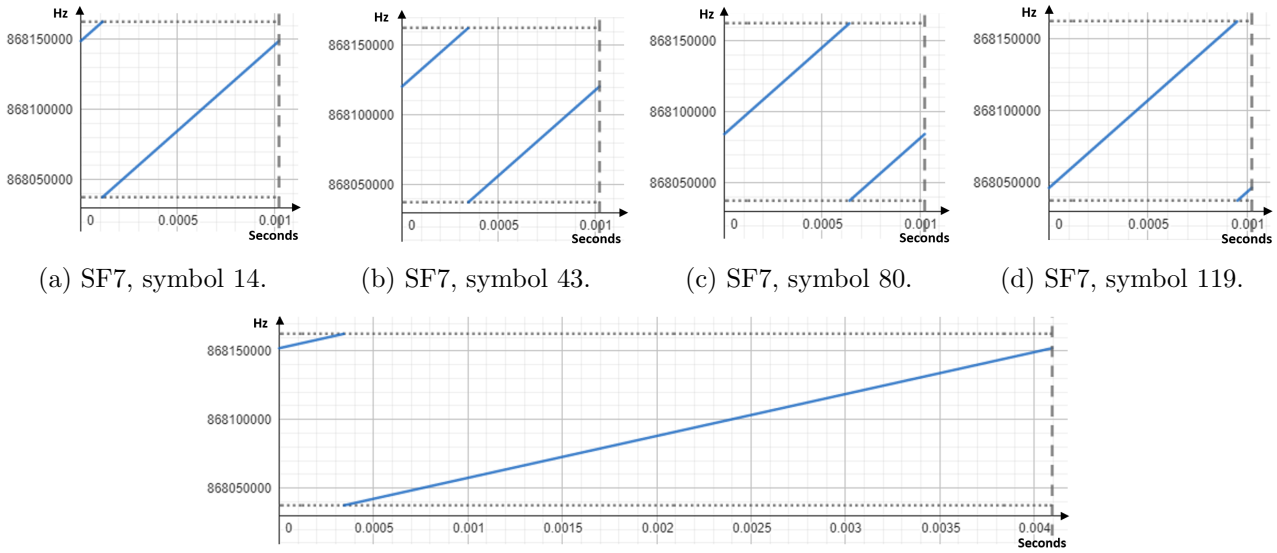


Figure 2.4: Overview of symbol encoding with LoRa chirp-based frequency modulation. The chirp duration consists of 2^{SF} *chips*, a minimal unit of time dependent on the bandwidth. Symbols are encoded by shifting a reference chirp signal by a desired number of chips. As a consequence, a LoRa chirp carries a number of bits equal to its Spreading Factor (SF). In the image: frequency graph of a single chirp carrying the symbol 23, assuming the reference chirp (carrying 0) is the diagonal signal going from corner to corner (shown in amplitude in Figure 2.3 on the preceding page).



(a) SF7, symbol 14. (b) SF7, symbol 43. (c) SF7, symbol 80. (d) SF7, symbol 119.

(e) SF9, symbol 43.

Figure 2.5: Comparative examples of symbol encoding. Reference for symbol 0 is taken from the diagonal signal going from corner to corner.

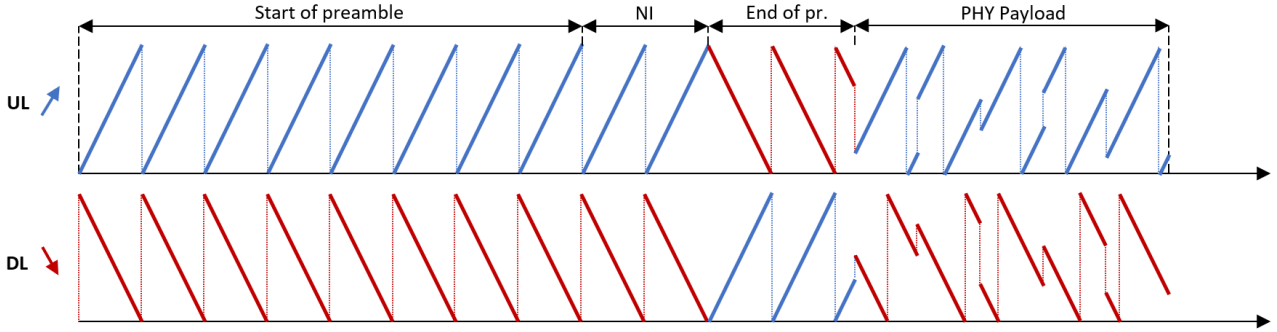


Figure 2.6: Up-link (UL) and down-link (DL) examples [31]. All LoRa transmissions begin with a synchronization preamble. Following the initial symbols (8 in this case) of value 0, there are 2 symbols encoding the Network Identification (NI), and 2.25 0-valued chirps of opposite 'chirpness' to indicate the end of the preamble. The actual frame payload starts at the end of the preamble, in this example 5 payload symbols are represented.

in Figure 2.5 on the preceding page. Moreover, a representation of concatenated symbols in the transmission preamble and payload is also shown in Figure 2.6 on the previous page.

Note that the chip rate remains the same, and equals to B, while the chirp duration (consequently time-on-air) increases drastically with the SF. On the positive side, higher SF yields higher processing gain, and, thus reduces the target Signal to Noise Ratio (SNR) for correct reception at the receiver. As a consequence, the range of a LoRa transmission can extend to several kilometers depending on the SF, transmission power and path loss of the considered scenario [32]. The SF values used by LoRa transmissions can go from 6 to 12, but SF_6 is rarely used. The sensitivity levels of a LoRa device are indicated in Table 2.1.

Table 2.1: Sensitivity of a LoRa device demodulator (125 kHz bandwidth) expressed in SNR and equivalent Received Signal Strength Indicator (RSSI) considering constant thermal noise (25°C) [27].

SF	SNR	RSSI
6	-5 dB	-121 dBm
7	-7.5 dB	-124 dBm
8	-10 dB	-127 dBm
9	-12.5 dB	-130 dBm
10	-15 dB	-133 dBm
11	-17.5 dB	-135 dBm
12	-20 dB	-137 dBm

LoRa operates on unlicensed sub-gigahertz industrial, scientific, and medical (ISM) radio bands. Depending on local regulations, the amount of available radio resources can vary; in the following we will exclusively consider European frequency plans. In Europe, LoRa is most commonly set to use the 863-870 MHz band. Frame transmissions take place on a central carrier Frequency Channel as shown in Figure 2.4 on page 16. Transmission bandwidth around the carrier frequency is chosen among one of the following values: 125 kHz, 250 kHz, 500 kHz.

Independent channels are created by choosing a combination of bandwidth/carrier frequencies that do not overlap; for example, two 125 kHz channels centered on the 868.1 MHz and 868.3 MHz frequencies are separated by 75 kHz to provide isolation. Higher bandwidths (250 kHz, 500 kHz) reduce the number of independent channels that can be created on a band, so the 125 kHz option is generally preferred in larger networks.

Network operators are free to add any frequency in the 863-870 MHz spectrum given that devices

Table 2.2: EU wide harmonised national radio interfaces from 863 MHz to 870 MHz (excerpt from [33]). Transmitters must either use a duty-cycle limitation, or implement Listen Before Talk (LBT) and Adaptive Frequency Agility (AFA).

Operational Frequency Sub-band	Maximum Effective Radiated Power (ERP)	Channel access and occupation rules
863 - 865 MHz	25 mW / 14 dBm	$\leq 0.1\%$ duty-cycle or LBT + AFA
865 - 868 MHz	25 mW / 14 dBm	$\leq 1\%$ duty-cycle or LBT + AFA
868 - 868.6 MHz	25 mW / 14 dBm	$\leq 1\%$ duty-cycle or LBT + AFA
868.7 - 869.2 MHz	25 mW / 14 dBm	$\leq 0.1\%$ duty-cycle or LBT + AFA
869.4 - 869.65 MHz	500 mW / 27 dBm	$\leq 10\%$ duty-cycle or LBT + AFA
869.7 - 870 MHz	5 mW / 7 dBm	No requirement
869.7 - 870 MHz	25 mW / 14 dBm	$\leq 1\%$ duty-cycle or LBT + AFA

are forced to use them in compliance with the restrictions indicated in the ETSI standard (shown in Table 2.2). These regulations impose a fair access policy on the band: devices must either implement polite spectrum access (LBT with AFA), or follow a duty-cycle restriction on the percentage of time they transmit. LoRa devices commonly favour the latter solution, as the former approach prevents coexisting transmissions on different SFs, a central design feature of LoRa. Furthermore, the current LoRaWAN specification exclusively uses duty-cycle limited transmissions to comply with ETSI regulations [34].

Devices must autonomously comply with duty-cycle regulations in each sub-band. No standard criteria is provided in specifications, but to ensure the duty-cycle is respected over short observation windows the following strategy is generally adopted: if the duration of a packet transmission is τ time units, then the transmission starting times of two consecutive packets on the same sub-band must be elapsed by at least $\frac{\tau}{\delta}$ time units, where δ is the duty-cycle.

2.4 LoRaWAN MAC Protocol

Specifications [22] define three classes of device behaviour: A, B, and C. In class A, after transmitting any uplink frame, a device must open two reception windows (RX1 and RX2) to possibly detect a downlink preamble and receive a message from the server, and then goes to sleep mode. This behaviour is shown in Figure 2.7 on the next page. On top of class A behaviour, class B devices periodically open reception windows synced with the server. Finally, class C devices are always listening when they are not transmitting. Class B and C give up some power efficiency in an attempt to broaden the

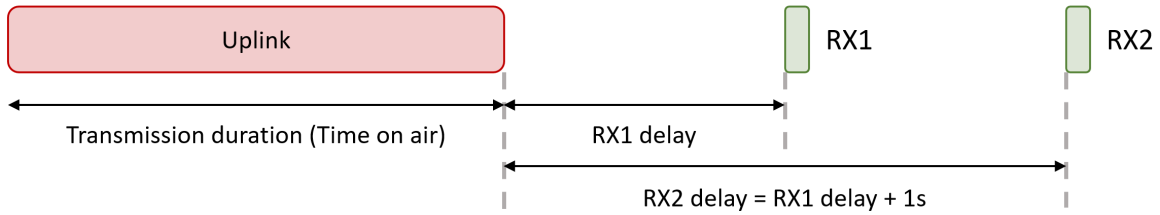


Figure 2.7: Overview of LoRaWAN class A behaviour. After uplink transmission, a device must open two reception windows (denoted RX1 and RX2). RX2 always opens 1 second after RX1.

range of use cases covered by LoRaWAN. The majority of existing devices are of class A, as B and C are later additions to LoRaWAN specifications. For this reason, in the following we will consider class A devices.

Devices rely on the selection of one among the available SFs for a transmission. In LoRaWAN, the term *data-rate* is used to identify a combination of a SF and a bandwidth (expressed in kHz) used for a transmission. The SF-based data-rates defined by the LoRaWAN specifications are listed in Table 2.3. An indicative maximum application payload size can be derived from the maximum MAC payload size, considering that it always contains an additional header of at least 8 Bytes (even more in presence of MAC primitives).

Table 2.3: LoRaWANtransmission data-rates and maximum payload. Transmission duration is provided for an example packet, computed following the formula in [27] with 8 preamble symbols, 4/5 coding rate, explicit header and CRC. Low data-rate optimization is active for SF_{11} and SF_{12}. We consider the minimum 51 B of headers/trailers of the LoRaWAN protocol and a 51 B application payload, the maximum size for SF_{12}. We also provide the maximum equivalent hourly goodput achievable with this payload by a device on a 1% duty-cycle sub-band without packet loss.

Data-rate	Configuration	Maximum MAC payload size	Example packet: 51 B application payload
0	SF_{12} / 125 kHz	59 B	2793.5 ms (6.6 kB/h)
1	SF_{11} / 125 kHz	59 B	1560.6 ms (11.8 kB/h)
2	SF_{10} / 125 kHz	59 B	698.4 ms (26.3 kB/h)
3	SF_{9} / 125 kHz	123 B	390.1 ms (47.1 kB/h)
4	SF_{8} / 125 kHz	230 B	215.6 ms (85.2 kB/h)
5	SF_{7} / 125 kHz	230 B	118.0 ms (155.6 kB/h)

In LoRaWAN, each device is assigned a set of channel frequencies and then does *frequency hopping*, i.e., it randomly picks up one frequency for each new transmission; this technique is claimed to help increasing the system robustness to interference. In Europe, the three default uplink channels are

868.1 MHz, 868.3 MHz and 868.5 MHz (1% duty-cycle) with a 125 kHz bandwidth. For downlink transmissions, the first receive window (RX1) uses the same settings (data-rate, frequency) as the uplink transmission, while the second (RX2) uses the 869.525 MHz frequency (10% duty-cycle) and SF_{12}.

There are two main types of frames in LoRaWAN, acknowledged and unacknowledged (also called, confirmed/unconfirmed traffic). The former requires an acknowledgement downlink transmission from the LNS for each upstream packet, while the latter does not. In addition, LoRaWAN provides a number of MAC primitives meant for control-plane communications (mainly, session management, parameter tuning and status requests). In the following we detail the management actions exposed by these commands.

2.4.1 Device Management Actions

In this section we list the device management actions offered by LoRaWAN, in the form of parameters that can be changed dynamically using the protocol primitives, i.e., without manual intervention. It must be noted that LoRaWAN also provides primitives for the initial procedure of activating a device and joining the network. These exist uniquely as an option to improve session security, so we do not consider them among management actions. In the following, we classify actions according to transmission and reception parameters.

2.4.1.1 Uplink transmission (TX) parameters:

- *TX data-rate.* It is set with `LinkADRReq` MAC primitive and it is encoded using 4 bits. The values previously shown in Table 2.3 on the preceding page consist in the minimal set of configurations that must be supported by any device. Only recently, the specifications introduced additional options with different modulation: SF_7 on 250 kHz bandwidth, Frequency Shift Keying (FSK) for an heavy trade-off (10x) between maximum range (↓) and bit-rate (↑) [35], and Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) for higher overall network capacity at lower device bit-rate [36]. It is not completely clear how these additional configurations integrate in the LoRaWAN management ecosystem, and they are currently not supported by default channels. For the scope of this thesis we will only consider data-rates from 0 to 5 as shown in Table 2.3 on the facing page.

- *TX power.* It is set with `LinkADRReq` MAC command and it is encoded using 4 bits. If it is set to a value higher than the device's maximum TX power, the device will operate at its maximum TX power. The encoding is region-specific and it is defined in the regional parameters document. In Europe it corresponds to the values in Table 2.4. The Effective Isotropic Radiated Power (EIRP) is the radiated output power referenced to an isotropic antenna radiating power equally in all directions. By default, the Max EIRP is considered to be 16 dBm. EIRP relates to ERP as follows: ERP = EIRP – 2.15 dB.

Table 2.4: Configurable TX power values in Europe.

TXPower	Configuration (EIRP)
0	Max EIRP
1	Max EIRP – 2 dB
2	Max EIRP – 4 dB
3	Max EIRP – 6 dB
4	Max EIRP – 8 dB
5	Max EIRP – 10 dB
6	Max EIRP – 12 dB
7	Max EIRP – 14 dB
8..14	Reserved for future use
15	Ignore field and keep current configuration

- *Number of transmissions for each uplink message.* It is set with `LinkADRReq` MAC command and defaults to 1 (i.e., no re-transmissions by default). The valid range is [1:15] (i.e., a 4 bits integer) and 0 makes the device ignore the field and keep the current configuration.

- *List of possible radio channels for uplink transmissions.* Channels can be added or modified using the `NewChannelReq` MAC command. In the EU863-870 band, LoRaWAN only supports a maximum of 24 channels. Each end device must include three default channels (868.1 MHz, 868.3 MHz and 868.5 MHz) that cannot be modified and guarantee a minimal common channel set between devices and gateways. For each channel the following information is stored:

  - *TX channel frequency.* For non-default channels is set with `NewChannelReq` MAC command. It is encoded using a 24 bits unsigned integer. The actual channel frequency in Hz is the value $\times 100$, whereby values representing frequencies below 100 MHz are reserved for future use. This allows setting the frequency of a channel anywhere between 100 MHz to 1.67 GHz

in 100 Hz steps. A value of 0 disables the channel.

- *TX channel min data-rate.* For non-default channels is set with `NewChannelReq` MAC command. It is encoded with 4 bits and it follows the same convention of the *TX data-rate* parameter.
- *TX channel max data-rate.* For non-default channels is set with `NewChannelReq` MAC command. It is encoded with 4 bits and it follows the same convention of the *TX data-rate* parameter. It must be higher of the *TX channel min data-rate* parameter.
- *RX1 frequency.* It can be set with `DlChannelReq` MAC command and defaults to the uplink frequency. It is encoded using a 24 bits unsigned integer and follows the same convention of a new *TX channel frequency.*

- *Mask of active radio channels for uplink transmissions.* It is set with `LinkADRReq` MAC command and is a subset of the list of possible uplink radio channels stored in the end device. The mask is 16 bits and each bit is set to 1 if the channel can be used for transmission. The bits represent the channels stored in the device in order. Subsequent `LinkADRReq` MAC commands can be used to set the channels mask if more than 16 channels are allowed. If this is the case, the other fields set by this MAC command will be updated using the last command in the batch. Finally, it exist an option in the `LinkADRReq` MAC command to disable the channel mask entirely.

- *Max aggregated TX duty-cycle.* It is set with `DutyCycleReq` MAC command and it corresponds to the TX duty-cycle over all sub-bands. The valid range for this parameter encoding is $d = [0 : 15]$ (i.e., a 4 bits integer) and the corresponding maximum duty-cycle is computed as: $\frac{1}{2^d}$.

2.4.1.2 Reception windows (RX*) parameters

- *RX1 delay from TX end.* It is set with `RXTimingSetupReq` MAC command and defaults to 1 second. This parameter is encoded using 4 bits, corresponding to the delay in seconds. Therefore, the range is between 1 and 15 seconds, with the value 0 defaulting to 1 second.

- *RX1 data-rate.* It is set with `RXParamSetupReq` MAC command and defaults to zero. It is encoded using 3 bits, so its value ranges between [0:5]. The actual data-rate is a function of the uplink data-rate and the 3 bits parameter (`RX1DROffset`) as given in Table 2.5 on the following page.

Table 2.5: Configurable RX1 data-rate offset from the TX data-rate.

`RX1DROffset`	**0**	**1**	**2**	**3**	**4**	**5**
Upstream data-rate	Downstream data-rate in RX1 slot					
DR0	DR0	DR0	DR0	DR0	DR0	DR0
DR1	DR1	DR0	DR0	DR0	DR0	DR0
DR2	DR2	DR1	DR0	DR0	DR0	DR0
DR3	DR3	DR2	DR1	DR0	DR0	DR0
DR4	DR4	DR3	DR2	DR1	DR0	DR0
DR5	DR5	DR4	DR3	DR2	DR1	DR0

- *RX2 data-rate.* It is set with `RXParamSetupReq` MAC command and follows the same 4 bits convention of *TX data-rate*. The default value is 0.

- *RX2 frequency.* It is set with `RXParamSetupReq` MAC command. It is encoded using a 24 bits unsigned integer and follows the same convention of a new *TX channel frequency*. The default frequency of the second reception window is 869.525 MHz (10% duty-cycle limitation).

2.5 Current Management Practices

The LoRaWAN protocol is often compared to pure ALOHA for its collision-prone nature and uncoordinated access to the radio medium [13], providing no guarantees in terms of reliability. Pure ALOHA and its enhancements (slotted, CSMA/CA) rely on feedback after transmission for collision observability [37]. Similarly, the LoRaWAN standard provides the option of using 'confirmed' (acknowledged) traffic: a downlink acknowledgment can be sent back to the device after every uplink transmission, hence possibly triggering re-transmissions. Many studies argue that this approach has limited viability in large networks [38], as it can quickly reduce the overall capacity. This is mainly attributed to mandatory duty-cycle limitations on gateways.

However, when looking at specifications [34], it is not clear if this imposition actually concerns gateways, which may be free to use polite spectrum access (LBT + AFA) in place of duty-cycle restrictions as indicated by regulations [33]. As a matter of fact, LBT is enabled by default by the Semtech UDP Packet Forwarder [28] that runs in most gateways. Assessing network capacity with confirmed traffic under this LBT assumption is an objective yet to be met by current research efforts.

Transmissions on different frequency channels do not interfere (i.e., they are orthogonal). Trans-

missions using different SFs on the same frequency are instead semi-orthogonal, that is, each one of them suffers of minor interference from the others [23]. In dense deployments, the increase of network traffic leads to a high number of collisions due to interference. This can be mitigated by distributing traffic over multiple SFs and regulating Transmission Power (TP), with the so called Adaptive Data Rate (ADR) algorithms.

Algorithm 1 Default LoRaWAN ADR algorithm [39].

$SNR_m \leftarrow$ max(SNR of last 20 frames)
$SNR_{req} \leftarrow$ demodulation floor of current data-rate (see Table 2.1 on page 18)
$deviceMargin \leftarrow 10$
$SNR_{margin} \leftarrow (SNR_m - SNR_{req} - deviceMargin)$
$steps \leftarrow \text{floor}(SNR_{margin}/3)$
while $steps > 0$ and $SF > SF_{min}$ **do**
    $SF \leftarrow SF - 1$
    $steps \leftarrow steps - 1$
while $steps > 0$ and $TP > TP_{min}$ **do**
    $TP \leftarrow TP - 3$
    $steps \leftarrow steps - 1$
while $steps < 0$ and $TP < TP_{max}$ **do**
    $TP \leftarrow TP + 3$
    $steps \leftarrow steps + 1$

By default devices are initially set to use SF_{12}. Then, existing implementations of ADR minimize the SF of each device according to the measured SNR of transmissions as shown in Algorithm 1. A simple heuristic based on the max SNR of the last 20 frames is used to determine whether SF or TP can be lowered. TP is only lowered if a device is already using SF_7. In addition to a higher bit-rate, using a lower SF has the added advantage of being more energy efficient thanks to the shorter duration of transmissions. In case devices fall out of range, they have a back-off procedure to first reset the TP and then the SF to the maximum.

Assuming a uniform distribution of devices per km^2, the classic SF-minimization approach for ADR is known to work best in situations where gateways are placed to maximize the coverage area, and many devices are too far to be using low SFs. However, in a scenario where every device is in range for SF_7, it has been shown that devices can be offloaded to higher SFs to increase even more the overall capacity at the expense of energy consumption [40]. On a network dimensioning note, this

of course also comes at a higher cost in terms of gateways per km^2, and thus its economical viability is still to be determined. In our work we only consider scenarios where the positioning of gateways is done to maximize coverage.

ADR is generally reserved for statically placed devices, as the channel conditions may change too quickly for it to be effective for mobile objects (e.g. in localization) and they are usually set to use SF_{12} [22]. Proposals for mobility support in ADR make use of trajectory estimation to anticipate the SF re-configuration [41].

In the literature, many different proposals exist to tackle the ADR problem, improving network scalability, throughput or energy consumption [42]. It must be noted that, despite the mitigation provided by ADR approaches, the capacity of each single SF remains limited, and thus packet loss still increases with the number of devices [26, 43]. Furthermore, the longer time-on-air on higher SFs results in higher collision probabilities and thus comparatively lower capacity.

2.6 Conclusion

In this introductory chapter we presented a detailed overview of the LoRaWAN technology. We introduced the network architecture, physical and MAC layer, paying special attention to concepts that will be useful in the following chapters, as the set of management actions provided by the MAC protocol. We also presented advantages and drawbacks of management approaches available in existing implementations on the market.

In all aspects, LoRaWAN seems to be designed towards maximal energy efficiency, a fit requirement for LPWAN use cases. On the other hand, for its simplicity, the technology provides no guarantees in terms of connection reliability. Due to the uncoordinated access to the radio medium, collisions and packet loss quickly increase with the number of connected objects. The majority of deployments mitigates these effects using an ADR algorithm to reduce the time-on-air of transmissions, thus the collision probability, also distributing devices over multiple quasi-orthogonal SFs. ADR alone, however, it seems not to be enough to solve the lack traffic quality guarantees in LoRaWAN, as packet loss still increases with the number of devices using a SF.

While this best effort operation may be fit for some LPWAN use cases, others may require guarantees in terms of QoS. This calls for the introduction of techniques to achieve some form of

traffic quality differentiation in the LoRaWAN RAN. It is also somewhat of an urgent aspect to be addressed, given the expected exponential densification of IoT networks [10]. For these reasons, a growing line of research inspired by the current evolution of 5G network management has started studying the introduction of network slicing principles in LoRaWAN.

Network slicing is a 5G network paradigm that postulates the creation of isolated network services with customized QoS. Clearly, the implementation of these principles in LoRaWAN introduces a number of technical challenges related to the specific characteristics of the RAN compared to 5G. In the following chapter, we will discuss these challenges after presenting the principles of network slicing, and providing additional context for their introduction in LoRaWAN. Finally, we will proceed to review the technical literature on the subject.

Chapter 3

Service Differentiation and Slicing in LoRaWAN

Contents

3.1 Introduction

In recent years, networks have been evolving towards the implementation of the design principles proposed by 3GPP and other standardization entities for 5G and beyond. In parallel to 5G development, the IoT has been growing exponentially and autonomously to become a panorama of technologies addressing a number of heterogeneous use cases and needs. In some specific instances, *non-3GPP* IoT network technologies reached the market and became popular, as happened for LoRaWAN [12]. A coherent vision of network evolution calls for the seamless integration of these existing technologies in the 5G and beyond ecosystem.

In the field of network management and orchestration, network slicing presented itself as the key paradigm to enable services with different requirements to share a common infrastructure. While all IoT traffic is sometimes relegated to the massive Machine Type Communications (mMTC) 5G slice prototype due to its general characteristics, many sub-taxonomies of applications with different requirements exist and need to be addressed [44]. In the domain of LPWANs, characterized by very low power consumption and long range requirements, LoRaWAN itself covers a plethora of different use cases [13, 45, 46, 18]. For this reason, a recent line of research has started investigating the adaptation of LoRaWAN to a network slicing paradigm for its integration into 5G and beyond architectures. These works give special attention to the radio resource allocation problem aimed at achieving differentiated traffic quality.

In this chapter we present existing related work on the introduction of service differentiation and slicing in LoRaWANs. In Section 3.2, we first present the general principles and motivations of the network slicing paradigm. Then, we proceed to discuss perspectives and challenges related to the integration of slicing in LoRaWAN in Section 3.3. Finally, Section 3.4 surveys and compares the state of the art on the topic of traffic quality differentiation in the radio access network.

3.2 Network Slicing Paradigm

Network slicing is a transformative paradigm in telecommunications and networking that allows for the creation of multiple virtual networks within a single physical infrastructure. It is a key enabler for the deployment of 5G and beyond, as it provides the flexibility, scalability, and efficiency required to support a wide range of use cases and diverse service requirements [6, 47].

At its core, network slicing involves partitioning a network into logically isolated segments, or slices, each tailored to specific applications or services. These slices are independent, virtual network instances that can be customized to meet the unique needs of different users, industries, or applications. Each slice can have its own set of resources, network functions, QoS requirements, and security policies. All these parameters are specified in SLAs between the network operator and its clients. According to [47], network slicing is built upon seven main principles: isolation, elasticity, automation, programmability, customisation, E2E and hierarchical abstraction.

Network slicing offers several advantages for network operators, service providers, and end users. Firstly, it enables the efficient sharing of network resources by allowing multiple slices to coexist on a common physical infrastructure. This leads to increased resource utilization and cost savings, as the network can dynamically allocate resources based on the requirements of each slice.

Secondly, network slicing enables the provisioning of customized and differentiated services. For example, a network operator can create a slice optimized for ultra-low latency applications, such as autonomous vehicles or industrial automation, while simultaneously offering a separate slice optimized for massive machine-type communications, such as the IoT. This level of customization ensures that each service receives the necessary network capabilities and performance, leading to improved user experience.

Furthermore, network slicing improves fault tolerance and security of services via isolation. Since each slice operates independently, they are agnostic to anomalies in services that would otherwise share the same resources. Also, providers can introduce new functionalities or experiment with innovative services without impacting the entire network. This flexibility encourages service innovation and reduces time-to-market for new offerings.

In terms of implementation, network slicing relies on virtualization technologies and SDN principles. Virtualization allows for the creation of virtual instances of network functions and resources (called Virtualized Network Functions (VNFs)), while SDN provides the centralized control and programmability needed to manage and orchestrate the slices. Together, these technologies enable dynamic provisioning, scaling, and management of network slices based on real-time demand and changing service requirements [47].

From an operator perspective, the concepts introduced by network slicing become especially

attractive when considering the contributions of a second, more recent, paradigm in network evolution, that is, Zero-touch network and Service Management (ZSM). ZSM proposes to achieve full automation with the introduction of smart loops for all support and maintenance operations needed to maintain SLAs [48]. In particular, ZSM specifications require this automation to *cover the management of the different technological domains such as Core, RAN and Transport domains, and also include the management of different types of resources such as VNFs, SDNs, virtual and physical resources, etc.* [49].

As the demand for diverse and specialized services continues to grow, network slicing holds considerable potential for various sectors. Despite this, in practice, some important aspects of network slicing are still at low level of maturity. Namely, network slicing is fairly mature for the core network, which is expected to lean on central and edge cloud for its virtualization/softwarization needs, but presents challenges of resource partitioning, isolation and virtualization in the RAN [6].

3.3 Slicing LoRaWAN: Perspectives and Challenges

The LoRaWAN technology allows for cheap [13], long range wireless communications (of the order of several kms [32, 45, 50]) of low throughput devices at minimal energy consumption. This enables battery powered devices to last for years [51, 46], removing the deployment and maintenance costs introduced by cables. The drawback of this design is of course the limited amount of data that each device can exchange, as shown in Table 2.3 on page 20.

Use cases for LoRaWAN are generally related to smart metering, environmental monitoring and localization in the domains of smart cities, smart buildings and agriculture [13, 46]. Therein, LoRaWAN is expected to support a number of specific use cases [45, 18]: smart parking, traffic sensors, air quality and water level monitoring, public transport and waste management for smart cities; house meters and appliances, health sensors and alarms in smart buildings; smart irrigation, wind and soil moisture monitoring, crop management, livestock monitoring in agriculture. See for instance a detailed list of urban MTC use cases in Table 3.1 on the facing page.

Given the diverse pool of possible applications, LoRaWAN use cases may necessitate different traffic requirements. Moreover, a separation of traffic flows could bring benefits in terms of resource utilization, scalability, security and fault isolation, as explained in Section 3.2 on page 30. These

Table 3.1: Device traffic parameters of urban MTC use cases based on data from New York in 2011 [17]. Original message sizes have been fragmented, consequently increasing the average inter-arrival time, to fit LoRaWAN application payload constraints (see Table 2.3 on page 20). PHEV stands for Plug-in Hybrid Electric Vehicle.

Application (Commercial, In-House)	Density [nodes/km²]	Average Message Inter-arrival Time [s]	Message Size [B]	Arrival Distribution	% of Total Traffic
Credit machine (grocery)	20.947	120	24	Poisson	0.110
Credit machine (shop)	2200.0	1800 (30 m)	24	Poisson	0.767
Roadway sign	316.47	30	1	Uniform	0.276
Traffic light	15.03	60	1	Uniform	0.007
Traffic sensor	15.03	60	1	Poisson	0.007
Movie rental machine	69.823	21600 (6 h)	38	Poisson	0.003
Home security system	3845.0	600 (10 m)	20	Poisson/uniform	3.350
Elderly sensor device	384.5	20	43	Poisson/uniform	21.609
Refrigerator	3845.0	3600 (1 h)	30	Poisson/uniform	0.838
Freezer	3845.0	86400 (24 h)	30	Poisson/uniform	0.035
Other house appliance	26915.0	86400 (24 h)	8	Poisson/uniform	0.065
PHEV charging station	7690.0	1400 (23 m, 20 s)	32	Poisson/uniform	4.595
Smart meter	11535.0	150	34	Poisson/uniform	68.338

reasons prompted researchers to investigate the integration of network slicing in LoRaWAN [52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63]. It is important to note that many of the existing works on the subject come from the same group of authors [52, 53, 54, 55, 56, 57, 58, 60], thus sharing ground assumptions and re-proposing some contributions.

According to the E2E principle of network slicing, LoRaWANs resources should be partitioned both in the 'core' network (network server, application servers, join server), and in the radio access part (end-devices, gateways). Considering the LoRaWAN server stack first, we see that existing implementations are generally cloud-native [30], thus already capable of the virtualization, programmability, scalability and observability requirements of network slicing. In [54], the authors first proposed an additional layer of programmable SDN switches between gateways and LNSs, also presented in [55, 56, 59].

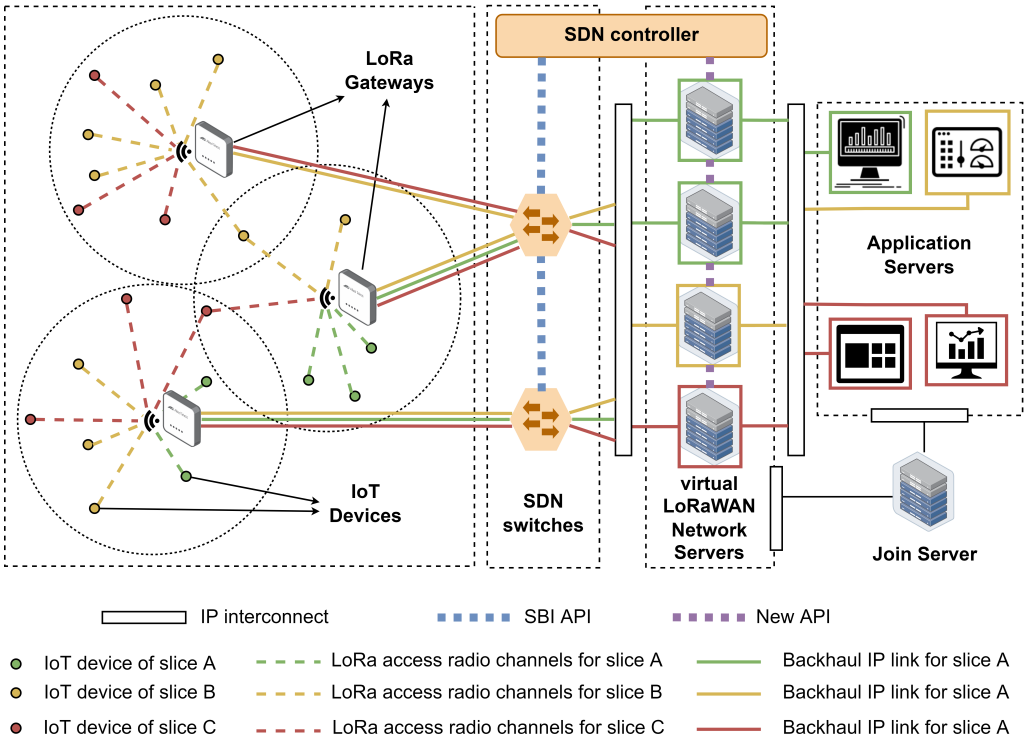


Figure 3.1: A high-level architecture to slice LoRaWAN using programmable SDN switches. Switches communicate with a SDN controller using a SouthBound Interface (SBI) Application Programming Interface (API).

An analogous architecture based on SDN switches is shown in Figure 3.1. A different virtual LoRaWaN Network Server (vLNS) is instantiated for each slice, with possible redundancy. Gateways receive traffic from devices of different slices, so switches are introduced to route traffic to the correct

vLNS. An SDN controller is in place to exchange details about device-slice membership between switches and vLNSs. This can be achieved using the information contained in the `DevAddr` field of the non-encrypted LoRaWAN protocol header: the `DevAddr` is a device identifier assigned by the LNS on connection, which also carries network membership information [22]. A similar mechanism is used by The Things Industries (a well known LoRaWAN stack operator) in the form of a centralized broker [64]. As co-located devices from different slices are affected by each other in the radio interface, management of radio parameters could be moved from the LNSs to gateways as done in [65]. Finally, E2E isolation is achieved by assigning different frequency channels to devices belonging to different slices.

It must be noted that LoRaWAN traffic is extremely light: as explained in Section 2.2 on page 13, by design, a single gateway receives much less than $40\,^{\text{kbit}}/_{\text{s}}$ from the radio interface, while covering multiple km^2 (more than 10 in urban setting [50]). For this reason, authors in [63] claim that slicing the LoRaWAN stack is not necessary. However, different LoRaWAN users may still require isolation for security reasons (for example, among co-located private and public networks). In conclusion, slicing the LoRaWAN stack is a problem with solutions within reach.

This changes when we consider the LoRaWAN RAN. Natively, LoRaWAN devices are expected to share all available radio resources. As mentioned in Section 2.4 on page 19, they randomly access one of the available frequency channels for each new transmission. Moreover, the absence of any form of access control results in a best effort operation where collisions quickly increase with the amount of devices in the network [16, 26, 20, 23, 43].

Integrating the principles of network slicing in the RAN requires the isolation of traffic flows coming from devices belonging to different slices. In addition, the network operator needs to regulate resources such that each traffic flow meets some radio quality requirement defined in SLAs. This second point alone arises a number of challenges due to the fully random access to the radio medium and the difficulties in achieving a fine grained partition of radio resources (the channels). Moreover, quality requirements need to be carefully designed taking into account the physical and protocol limitations present in LoRaWAN.

While network slicing is defined as an E2E paradigm, related works claiming to introduce LoRaWAN slicing only consider the radio access part [52, 53, 57, 58, 60, 61, 62, 63], or provide a high level architecture [54, 55, 56, 59]. For this reason, rather than addressing network slicing as a whole, we

focus in the following on 'traffic quality differentiation' in LoRaWAN, i.e., the integration of network slicing principles in the RAN.

3.4 Related Work on Traffic Quality Differentiation

In the following sections, we will first present how related works addressed the choice of quality requirements for LoRaWAN radio SLAs, together with our assumptions on the matter. Then, we will survey and compare existing techniques for traffic quality differentiation among clusters of LoRaWAN devices.

3.4.1 Quality Requirements

As explained in Section 3.2 on page 30, network slicing relies on the partition of the network into more or less isolated horizontal slices for accommodating services with guaranteed SLAs. To address the lack of quality requirements in LoRaWAN, authors in [52, 53, 54, 55, 56, 57, 58, 59, 60] adopt 5G MTC QoS objectives, generally consisting of three classes defined in terms of latency and reliability requirements as shown in Tables 3.2, and 3.3 to 3.4 on the next page. Latency and reliability bounds are then used as priority guidelines in the optimization of resources allocated to clusters of devices.

Table 3.2: IoT QoS Class Identifiers (QCIs) table used in [52, 53].

QCI	Slice ID	PDB	Services	Percentage of IoT flows
5	1	<100 ms	Surveillance and Emergency Alerting	10%
1-2	2	100-1000 ms	Health Sensors	15%
3-4	2	100-1000 ms	Home Security System	15%
6	3	>1000 ms	Smart Metering Applications	60%

Legend: Packet Delay Budget (**PDB**) is a one-way latency requirement.

The original targets for 5G MTC are in the order of hundreds of milliseconds (or less) for latency, and 99.9% (and higher) packet delivery ratios; they may, however, be difficult to meet in LoRaWAN, since this technology has very specific bandwidth, transmission speeds and medium access control. Indeed, such reliability targets are actually never met in the numerical results reported in [23, 15, 16, 13, 45]. Latency objectives could only be reached at the expense of decreasing allowed payloads and SFs to low values (compare for instance the values reported in Table 2.3 on page 20). Given the importance

Table 3.3: IoT QCIs table used in [54, 55, 58, 59, 60].

QCI	Slice Name	Resource Type	Priority	PDB	PER %	Example Services
71	URA	GBR	1	100 ms	10^{-3}	Real time, smart mobility
72	RA	GBR	2	200 ms	10^{-3}	Real time, eHealth and home security
73	BE	NGBR	3	300 ms	10^{-6}	Delay tolerant, smart agriculture

Legend: Urgency and Reliability Aware (**URA**), Reliability Aware (**RA**), Best Effort (**BE**); Guaranteed Bit-Rate (**GBR**), Non-Guaranteed Bit-Rate (**NGBR**) are originally defined for MTCs using Evolved Packet System (EPS) protocols [66] and indicate whether or not the class has bandwidth guarantees; **PDB** and Packet Error Rate (**PER**) are one-way latency and reliability requirements, respectively.

Table 3.4: QoS requirements for industrial slices used in [56, 57].

Slice Name	PDB	Reliability	Packet Size	Priority (urgency)	Applications
UCLE	50 ms	1-10^{-4}	24 B	1	Emergency action, safeguarding systems
HCLE	100 ms	1-10^{-4}	512 B	2	Scale readings
LCLE	500 ms	1-10^{-6}	250 B	3	Standard mobile robot

Legend: Ultra-high Critical of Latency and Efficiency (**UCLE**), High Critical of Latency and Efficiency (**HCLE**), Low Critical of Latency and Efficiency (**LCLE**); **PDB** is a one-way latency requirement.

of SLAs in network slicing, a review of objectives has to be done, in light of technological constraints, to be able to design techniques capable of meeting realistic requirements.

In [61], authors adopt class SLAs derived from packet success rate and network energy consumption requirements, without explicitly providing quantitative values. Note that here each class identifies an empty channel having quality requirements in place of devices: devices are then associated to one or more channels/classes so that channel requirements satisfaction is optimized. The association is carried out with K-means clustering paired with a coalitional game setting. In [63], three classes are defined to host and separate confirmed, unsolicited and unconfirmed traffic. While classes and requirement definition in this last work reflect the best effort nature of LoRaWAN, in this thesis we set out to investigate the possibility of introducing numerical objectives in SLAs that could eventually be met.

Similarly to [62, 67], we define in this thesis classes of services in terms of PDR [1]. Latency is a

[1]The PDR, also known as Packet Delivery *Rate*, Data Extraction Rate (DER) [16] or Packet Success Rate (PSR) [61] indicates the percentage of correctly received packets over a period of time and it can be computed in practice using frame counters. Sometimes it is also indicated with the opposite, as in 1−PER or 1−Packet Loss Rate (PLR).

secondary metric because LoRaWAN use cases are not time-critical [13], and throughput is closely related to PDR due to duty-cycle and packet size constraints [34]. Energy considerations are left out of the scope of this thesis for future work. It is known that energy consumption can be minimized by a suitable ADR allocation (see for instance [39]), however it could be also increased to improve the PDR, for instance using multiple packet re-transmissions [68].

Under the above assumptions, we adopt a three level approach to PDR differentiation by introducing ultra-high, high, and low PDR levels as in 5G MTCs for the Industrial IoT [56, 57]. To be closer to real LoRaWAN scenarios and capabilities, we consider 97% PDR, 90% PDR, and 70% PDR levels, based on the results in [45]. While some of the early related works introduce an initial step to assign devices to the class that best meets its own QoS requirements [52, 53, 55, 56], we assume cluster membership of devices as predetermined, based on the cluster PDR-requirements.

3.4.2 Traffic Quality Differentiation Techniques

In [52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63] the assignment of more or less independent interference domains to clusters of devices allows authors to achieve differentiated traffic quality. A comparison of related works on the subject is shown in Table 3.5 on the next page, classified of publications of first appearance. With the exception of [62], all these works adopt the requirements explained in Section 3.4.1 as guidelines for a resource optimization process, lacking numeric guarantees on the final outcomes.

These works generally adhere to a common methodology consisting of multiple steps as follows: (i) channel allocation, (ii) SF and TP assignment and (iii) traffic control. In most cases, these algorithmic steps are modular, namely, they can be combined with each other or deployed independently on top of the default LoRaWAN operation. In only one case [60], they are tested in an online setting where the time to reconfigure devices is taken into account. This point is of particular importance due to the sparsity of LoRaWAN traffic, where devices seldom transmit more than each 10 minutes [24]. For each algorithmic step, we review in the following the techniques adopted by related works.

3.4.2.1 Channel Allocation

In each gateway, a portion of the available frequencies is reserved for each cluster of devices to introduce isolation. In [52], this partition of frequencies is done proportionally to the average device

Table 3.5: Comparison of LoRaWAN quality differentiation approaches. Many of these techniques are often re-used in more than one of the presented works, so, for clarity, here we indicate the reference of first appearance and group similar approaches.

Reference	RoR	CA	ADR	TC	Technique	M	OT
[52, 53]	Guideline	✓			H	✓	
[54, 58]	Guideline		✓		MCDA	✓	
[55]	Guideline	✓			GT	✓	
[56]	Guideline	✓			MBGD	✓	
[57, 59]	Guideline		✓		DRL	✓	
[60]	Guideline		✓		DRL	✓	✓
[61]	Guideline	✓			KC+GT	✓	
[62]	Strict	✓		✓	GA		
[63]	Guideline		✓		H	✓	

Legend: Role of Requirements (**RoR**), Channel Allocation (**CA**), SF and TP allocation (**ADR**), Traffic Control (**TC**), Modular (**M**), Online execution is Tested (**OT**). Modular means that CA/ADR/TC are designed to not necessarily need one another. - Heuristic (**H**), Game Theory (**GT**), Multi Criteria Decision Analysis (**MCDA**), Mini-Batch Gradient Descent (**MBGD**), Deep Reinforcement Learning (**DRL**), K-means Clustering (**KC**), Genetic Algorithms (**GA**).

bit-rate of clusters. Then, each frequency channel is further classified based on radio quality, using weights obtained via an analytical hierarchy process approach. Afterwards, each device is assigned to the best single-channel among the ones selected for the cluster: this is done greedily by establishing a priority among devices based on how far their one-way latency is from the requirement of their class. In [53], the frequency partitioning method is refined by considering, for each gateway, only those devices in range for the average device bit-rate calculation, instead of the totality of devices in the network. Partitioning resources proportionally to the average device bit-rate (instead, for instance, of the total cluster bit-rate) can introduce a fairness issue when there is high disparity in the number of devices among different groups, and thus it is not desirable.

In [55], the authors propose a two-step game-theoretical method for channel set partitioning and single-channel assignment based on bankruptcy games and one-to-one matching theory, respectively. In [56], an iterative approach based on mini-batch gradient descent is used to estimate the portion of frequencies for each cluster. As previous works, these techniques do not consider cluster quality requirements in the channel set partitioning process, but only in a second intra-cluster single-channel assignment step based on measured channel radio quality. However, compared to previous works

they consider the demands of a class based on the total bit-rate of devices in the class, instead of the average device bit-rate.

Whilst not explicitly specifically addressing requirements for quality differentiation, it is worth mentioning that authors in [69] also propose a method to partition channels for multi-operator deployments; an iterative approach is proposed to maximize the total network throughput while maintaining proportional-fairness of data extraction among operators. In [61], the authors simply assign one channel per class of requirements, paying more attention to the association of devices to classes as explained in Section 3.4.1. Finally, the authors in [62] provide devices with a discrete probability distribution to determine which frequency channels to use and how much. Contrary to other works, these probabilities are optimized with genetic algorithms, taking into consideration PDR requirements.

3.4.2.2 SF and TP Allocation

An assignment of SFs and TP to devices is achieved for the different clusters, according to their needs. At this point, clusters operate on different frequencies and this second step can effectively be done by an ADR algorithm with a different optimization objective for each cluster. In [54, 58], this task is executed using a multi-criteria decision analysis approach. According to the class urgency, reliability and energy consumption requirements, SF and TP configurations are ranked, and the best one is assigned to every device in the class.

In [57], this problem is addressed via deep federated Q-learning, in [59] is addressed via a 'transfer learning-based multi-agent deep deterministic policy gradient' algorithm, and in [60] via multi-agent deep Q learning. Due to the sparse characteristics of LoRaWAN traffic, deep reinforcement learning approaches seem to require a large amount of training time to learn a specific scenario (thousands of observations from a device just to surpass the base ADR algorithm [60]), and their generality is not tested in multiple randomized cases.

In [63], authors study two simple criteria to separate acknowledged, unacknowledged and unsolicited traffic: (i) SF_7 for acknowledged, SF_{11} for unsolicited, the rest for normal traffic, and (ii) SF_7 for unsolicited, SF_{12} for acknowledged, and again the rest for normal unacknowledged traffic. While this approach may work in low-density networks, the partial isolation provided by different SFs [70, 23] can lead to packet loss caused by a different class, which may not be enough to fulfill the isolation

and security requirements of network slicing. The other works all adopt the default ADR algorithm detailed in [39].

3.4.2.3 Traffic Control

A technique is defined to constrain the amount of traffic that devices can put through the radio channel. This can be done to limit the PDR degradation due to collisions or the energy consumption of a device, the latter being in major part a consequence of devices' transmission and reception states. In [62], the authors block devices in a secondary, best-effort, cluster such that the PDR goal is able to be reached by a priority group. Results show that PDR goals can be precisely met in the priority cluster, at the expense of a high percentage of blocked best-effort devices. This traffic control solution is non-standalone, as it requires two clusters, a priority rule, and a specific technique for channel selection. It also requires a modification of the LoRaWAN specifications to allow non-uniform frequency hopping. Given that we aim at introducing strict requirements for the traffic in different clusters, we will also experiment with forms of traffic control.

3.5 Conclusion

In this chapter we have discussed the topics of service differentiation and network slicing in LoRaWAN. After presenting the principles of network slicing as defined for 5G networks, we explained how, in theory, separated traffic flows could be used to improve LoRaWAN service differentiation and customization. This of course relies on how well resources can be partitioned and customized. We discussed how most of the challenges in the introduction of network slicing in LoRaWAN come from the RAN, as the network server stack already has most of the required properties (i.e., virtualization, programmability, scalability, observability).

The term LoRaWAN slicing identifies an E2E paradigm, but most of the related works on the subject only touch the radio resource allocation problem contained therein. To avoid confusion, we coin the terminology LoRaWAN traffic quality differentiation to refer this resource allocation problem derived from network slicing, which is also the main topic explored in this thesis.

A notable challenge introduced by traffic quality differentiation is the choice of effective quality requirements. Network slicing postulates service customization through the definition of SLAs

containing QoS targets, but natively LoRaWAN is a best effort technology. Most existing proposals in the state of the art adopt quality requirements that are related to 3GPP-specific technologies (5G MTCs), far from LoRaWAN capabilities both in terms of latency and reliability. In fact, these objectives are then only used as guidelines for best effort resource optimization.

In our contributions we aim at introducing realistic numerical objectives for SLAs that could eventually be met. In our work we define classes in terms of packet reliability, or PDR, because it is a key metric normally not guaranteed in LoRaWAN. Related works show that it is not uncommon to measure PDRs well under 90%, thus other metrics quickly become less relevant.

Similarly to most related works we adopt disjoint set of frequency channels to achieve isolation between clusters of devices. Unlike referenced works, we aim at integrating requirements directly in the channel partitioning problem among clusters. Moreover, we consider the integration of traffic control techniques in LoRaWAN to reduce the effects of interference. Limiting the offered traffic seems to be the only option that allows us to regulate PDR while also changing as little as possible the energy-efficient design that popularized LoRaWAN. Given that we focus on regulating interference, an important assumption in our work is that the gateways placements is always dense enough to provide near perfect coverage to any device: this is not too unrealistic given that we only consider static connected objects. On these assumptions, in the following chapter we present a first contribution to traffic quality differentiation with guarantees.

Chapter 4

A First Approach to Traffic Quality Differentiation with Guarantees

Contents

4.1 Introduction

In LoRaWAN, despite SF and TP optimizations, quality does degrade when a high number of devices transmit in the vicinity of a radio gateway [26, 43]. Resource allocation algorithms currently proposed do not consider limiting device access and/or traffic to prevent performance degradation. Such actions may be in contradiction with the openness and best effort principles of LoRaWAN, but seem to be inevitable in light of the expected densification of connected objects in quality demanding applications.

In this chapter, we propose a radio resource management scheme to integrate quality differentiation in LoRaWAN in dense scenarios. For this purpose, we exploit the isolation provided by the available radio frequencies to group devices into clusters with different PDR targets, and to optimize behavior configuration in order to reach the desired PDR objectives. With respect to existing proposals, we introduce two new contributions:

- The share of radio resources, i.e., the number of channels, assigned to each cluster is scaled according to the PDR requirement. In [52, 53, 55, 56], this share is only based on cluster throughput. Thus, compared to previous proposals we propose a more fine grained approach that integrates requirements in the channel assignment problem.

- We limit the number of devices using each SF in a cluster according to the number of channels assigned. This is important to allow more precise control over the satisfaction of requirements, and to address network congestion if the population of devices around a gateway increases. For example, this could happen in a smart city scenario due to bicycle tracking. To the best of our knowledge, in LoRaWAN literature this has not been considered before. While a similar idea is presented in [62], the two works were made public in a concurrent time frame.

We consider three classes of services defined in terms of PDR (97%, 90%, and 70%) as motivated in Section 3.4.1 on page 36. Moreover, we evaluate two variants of the approach at different levels of isolation between clusters. Simulation results of our proposal show that PDR-differentiation is fairly achieved as a trade-off between device density and maximum cell range.

The results presented in this chapter were published in the following full conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Packet delivery ratio guarantees for differentiated LoRaWAN services," in *Proc. 2022 IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 2014–2019, doi: 10.1109/GLOBECOM48099.2022.10001145.

The chapter is organized as follows. Section 4.2 presents our PDR differentiation framework. Section 4.3 describes the numerical evaluation. Concluding remarks are presented in Section 4.4.

4.2 LoRaWAN Packet Delivery Differentiation

Since our proposal aims at differentiating the PDR of devices according to their cluster membership, we first present an estimation of the PDR followed by the algorithmic framework to assign radio resources and transmission parameters.

4.2.1 Offered Traffic Based on PDR Estimation

We rely on the capacity model in [43]. The average number of concurrent transmissions on a channel is assessed via the offered traffic defined as $\nu = \tau\lambda$, where τ is the average transmission duration, λ is the arrival rate of frames. From the perspective of a frame arriving at a gateway, the probability of meeting k additional overlapping transmissions is

$$P(X = k) = \frac{(2\nu)^k e^{-2\nu}}{k!}, \tag{4.1}$$

as there should be no other transmission events for a time 2τ to avoid overlap. The PDR is then derived as the sum of the probability of no collisions ($k = 0$), and the probability of having a non destructive collision ($k = 1$) as

$$PDR = e^{-2\nu} + \frac{2\nu}{\gamma + 1} e^{-2\nu} \stackrel{\text{def}}{=} h(\nu), \tag{4.2}$$

where $\gamma = 1\,\text{dB}$ is the difference in received power necessary to capture one of two overlapping transmissions on a SF [70], and $(\gamma + 1)^{-1}$ is the probability that an exponential random variable is γ times greater than another one.

From Equation (4.2), we get the estimate ν of total offered traffic on a frequency and SF to respect the desired PDR. As the result is the same for each SF, the maximum traffic supported by a gateway can eventually be estimated as 6ν multiplied by number of frequencies used. By inverting

Equation (4.2) we obtain

$$h^{-1}(PDR) = -\frac{1}{2} \cdot \mathcal{W}_{-1}\left(-\frac{\xi}{e^{\xi}} \cdot PDR\right) - \frac{\xi}{2} = \nu, \tag{4.3}$$

where $\mathcal{W}_{-1}$ is the lower branch of the Lambert W function and $\xi = \gamma + 1$.

The duty cycle sets an upper bound for the offered traffic ν_d of a single device. In real LoRaWAN scenarios, devices usually transmit much less than 1% of time [24]. Thus, supposing to know a priori the maximum throughput t_d needed by a device d, we get a better offered traffic estimate with

$$\nu_d = \frac{p}{dr_{SF}}\lambda = \frac{t_d}{dr_{SF}}, \tag{4.4}$$

where p is the packet length and dr_{SF} is the data-rate of the SF to be used by the device. We use this estimate for the resource allocation of the proposed technique.

4.2.2 Parameter Allocation Technique

Our approach is based on four algorithmic steps to ensure that PDR targets are closely met for clusters by means of device configuration. We design our procedure to be re-executed whenever the network detects a significant change in device population around a gateway.

We assume that devices transmit periodically, with complete heterogeneity in terms of inter-transmission period and packet size. As a prerequisite, each device is assigned to a cluster and has to provide the network operator with its maximum planned throughput ($^{\text{bit}}/_{\text{s}}$). We further assume that each device is positioned in the radio range of a gateway and that we already cold started devices to collect SNR data. Note that existing LoRaWAN network servers already use SNR to identify for each device the best downstream gateway and the best SF and TP configurations.

Let D be the set of devices; $d \in D$ has declared throughput t_d and belongs to cluster c_d. Clusters are defined in terms of a minimum PDR. The set of clusters is denoted by C, cluster c has required PDR noted PDR_c; G is the set of gateways and F the set of frequencies. See Table 4.1 on the next page for a definition of the various parameters.

The four main steps of our approach are (1) device grouping, (2) computation of spectrum shares, (3) frequency allocation and (4) assignment of frequencies and SFs to devices; they are detailed hereafter.

Table 4.1: Glossary of notation used in this chapter.

Symbol	Description
ν	Offered traffic
τ	Frame transmission duration
λ	Frame arrival rate
$h^{-1}(PDR)$	Capacity of a SF on a frequency at desired PDR
d, D	Device, set of devices
c, C	Cluster, set of clusters
g, G	Gateway, set of gateways
F	Set of frequencies
$SNR_{g,d}$	SNR measured from device d at gateway g
c_d	Cluster of device d
t_d	Throughput needed by device d
PDR_c	PDR requirement of cluster c
$D_{g,c}$	Subset of devices of cluster c assigned to gateway g
$w_{g,c}$	Resource demand of cluster c at gateway g
$w'_{g,c}$	Ideal bandwidth share of cluster c at gateway g
$\overline{w}_{g,c}$	Number of frequencies assigned to cluster c at gateway g
$F_{g,c}$	Subset of frequencies assigned to cluster c at gateway g
dr_{SF}	Data-rate of SF

4.2.2.1 Step 1: Device Grouping

We subdivide devices according to the gateway g measuring the highest SNR. Then, each device is assigned to a set $D_{g,c}$ according to the cluster c_d they belong to: it so minimizes the transmission parameters of devices to reduce interference at farther gateways.

4.2.2.2 Step 2: Computation of Spectrum Shares

For each (g, c) gateway-cluster pair, we compute a weight $w_{g,c}$ and we use it to determine the share of the total number of frequencies to be allocated to cluster c at gateway g. The weight $w_{g,c}$ is proportional to the total resource demands of devices in each cluster. Then, if the number of devices accessing the network exceeds the capacity, we exclude a uniform percentage of them between the clusters.

Devices at the highest levels of quality need more resources to maintain a low number of collisions. Hence, we determine the transmission bit-rate needed to achieve the declared throughput t_d while respecting the offered traffic constraint obtained with the capacity model of Equation (4.3) at the

desired PDR_c. Then, we sum them up so that

$$w_{g,c} = \sum_{d \in D_{g,c}} \frac{t_d}{h^{-1}(PDR_c)} \quad \forall g \in G, \forall c \in C. \tag{4.5}$$

Finally, the shares $w_{g,c}$ are proportionally normalized to sum up to $|F|$, the total number of frequencies,

$$w'_{g,c} = |F| \cdot \frac{w_{g,c}}{\sum_{c' \in C} w_{g,c'}} \quad \forall g \in G, \forall c \in C. \tag{4.6}$$

4.2.2.3 Step 3: Frequency Allocation

Because the quantity $w'_{g,c}$ is real and possibly less than one, this value cannot be directly used as the number of frequencies to allocate to cluster c at gateway g, which has to be an integer larger than or equal to 1. To overcome this problem, we propose the two following methods of fixing the set $F_{g,c}$ of frequencies such that $|F_{g,c}| = \overline{w}_{g,c}$ and $\bigcup_{c \in C} F_{g,c} = F$.

Hard Isolation We do not mix devices of different clusters in a same frequency. The number of frequencies is obtained by minimizing the sum of the gaps between the continuous frequency share values $w'_{g,c}$ computed in the previous step, and the final discrete numbers of frequencies $\overline{w}_{g,c}$ to be assigned to each cluster in a gateway:

$$\min \sum_{c \in C} |w'_{g,c} - \overline{w}_{g,c}| \quad \forall g \in G \tag{4.7}$$

Exact solutions are obtained in $O(|G| \cdot |C| \log(|C|))$ complexity with a trivial greedy algorithm that (i) reserves $\lfloor w'_{g,c} \rfloor$ to each cluster, and (ii) assigns the remaining frequencies one at a time to the clusters ordered by highest decimal part of $w'_{g,c}$.

Soft Isolation In this policy we exploit the fact that we can upgrade devices to higher PDR clusters without breaking the requirements. In each gateway, starting from the highest PDR cluster c we allocate $\overline{w}_{g,c} = \lceil w'_{g,c} \rceil$, and we randomly choose devices to be upgraded (without changing their cluster membership) from the next lower cluster, so to fill the $\overline{w}_{g,c} - w'_{g,c}$ difference. In line with Equation (4.5), each new device d occupies an amount equal to $t_d / h^{-1}(PDR_c)$. This procedure is repeated for all following clusters in terms of PDR until we have frequencies to allocate.

4.2.2.4 Step 4: Assignment of Frequencies and SF to Devices

We assign all frequencies in $F_{g,c}$ to each device in cluster c of gateway g. We maintain frequency hopping to mitigate the impact of external interference, and TP minimization in SF_7 as in existing approaches [39].

We progressively assign a SF to each device by allocating higher SFs to devices $D_{g,c}$ ordered by descending SNR. Based on the model of Equation (4.3), each SF in a frequency of cluster c has a maximum offered traffic equal to $\nu = h^{-1}(PDR_c)$. So, starting from the lowest SF and the device $d \in D_{g,c}$ with the highest SNR, we allocate the SF to d and subtract from ν a quantity equal to the offered traffic share $\nu_d = t_d/dr_{SF}$ of d. If the capacity of the last SF is depleted we need to avoid more devices around a gateway (e.g., by limiting the duty cycle to an extremely small value using MAC primitives).

Algorithm 2 LoRaWAN clusters configuration.

Inputs: D, C, G, F, $SNR_{g,d}$, c_d, t_d, PDR_c, `hard` (flag)

```
Use SNR_{g,d} and c_d to insert d in D_{g,c}                    ∀d ∈ D
for all g ∈ G do
    Compute cluster shares w'_{g,c}                             ∀c ∈ C
    if hard then
        Hard Isolation to get the number of frequencies w̄_{g,c}
    else
        Soft Isolation to get the number of frequencies w̄_{g,c}
    Reserve F_{g,c} frequencies based on w̄_{g,c}                ∀c ∈ C
    for all c ∈ C do
        Order D_{g,c} by descending SNR_{g,d}
        Assign all F_{g,c} frequencies to each d ∈ D_{g,c}
        Assign a SF to each d ∈ D_{g,c}
```

A summary of the overall approach is given in Algorithm 2. If we consider a uniform density of $^{\text{nodes}}/_{\text{km}^2}$, we obtain an estimate of the gateway range based on the last device served. Hence, if path loss data of an area is available, the technique can be used beforehand by the operator to dimension the number of gateways in the network based on maximum device density and offered levels of PDR.

4.3 Numerical Results

To evaluate our proposed scheme, we have developed a lightweight simulator for the LoRa uplink traffic physical layer in the ns-3 simulation environment [71]. Interference and path loss computations follow the state of the art model from [26].

4.3.1 Simulation Setup

Seven gateways, one per cell, are placed using hexagonal tiling as in Figure 4.1, where circle radius is 7.5 km. This results in a total area of 1114.71 km^2, used to obtain densities in the following sections of this chapter. Devices are uniformly placed in range of gateways, and they transmit with a periodical traffic pattern, interfering with other devices both in the same and other cells. To simulate heterogeneity, the inter-transmission time and payload of each device are extracted from a truncated Gaussian random variable with mean 600 s and variance 300 s, and with mean 31 B (13 B for headers) and variance 10 B, respectively. The average application payload (18 B) is taken from the results in [24], and the inter-transmission time is selected to reproduce the bit-rate of common outdoor sensors [17]. In addition, devices check duty-cycle limitations before sending, and eventually postpone the transmission. The physical layer parameters of devices are listed in Table 4.2.

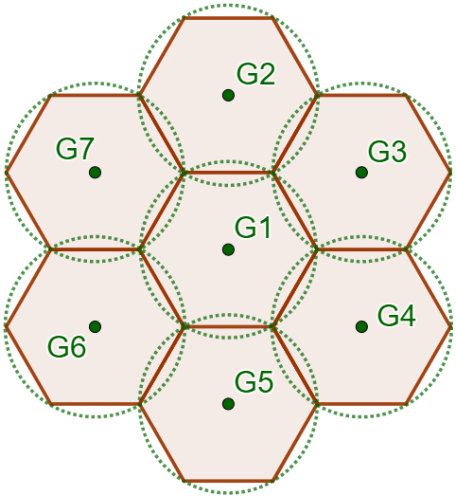


Figure 4.1: Placement of gateways and devices in simulations.

Table 4.2: Physical layer parameters of devices [34, 33, 27].

Parameter	Value(s)
Antenna ERP (dBm)	14, 12, 10, 8, 6, 4, 2, 0
Frequency (MHz)	868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, 867.9
Spreading Factor	7-12
Bandwidth (kHz)	125
Coding rate	4/5
Preamble length	8
Explicit header	Enabled
CRC	Enabled
Low data rate optimization	Enabled (SF_{11}/SF_{12})

Packet transmission time is computed following the SX1272/73 transceiver datasheet [27]. Gateways are modeled on Semtech's SX1301 chips for LoRaWAN outdoor macro-gateways and therefore have

Table 4.3: Gateway sensitivity levels (dBm) required for correct packet reception on the different SFs [25].

SF_7	-126.5
SF_8	-129.0
SF_9	-131.5
SF_{10}	-134.0
SF_{11}	-136.5
SF_{12}	-139.5

Table 4.4: Signal to Interference Ratio (SIR) thresholds (dB) of reference SFs (rows) against interference coming from other SFs [70].

	SF_7	SF_8	SF_9	SF_{10}	SF_{11}	SF_{12}
SF_7	1	-8	-9	-9	-9	-9
SF_8	-11	1	-11	-12	-13	-13
SF_9	-15	-13	1	-13	-14	-15
SF_{10}	-19	-18	-17	1	-17	-18
SF_{11}	-22	-22	-21	-20	1	-20
SF_{12}	-25	-25	-25	-24	-23	1

8 parallel reception paths. Levels of gateway to SF Sensitivity are in Table 4.3. In interference calculations, we adopt the empirical Signal-to-Interference thresholds matrix shown in Table 4.4. We simulate the network running for 10 h and we replicate simulations 30 times to be able to draw figures with 95% confidence intervals.

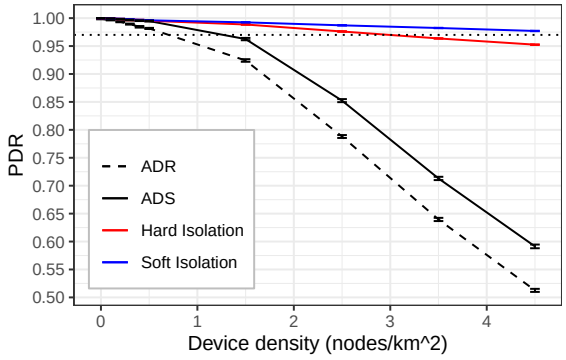
4.3.2 Results Analysis

In the experiments we evaluate the Hard and Soft Isolation policies for frequency assignment proposed in Section 4.2 against increasingly high network densities, up to 45 $^{\text{nodes}}/_{\text{km}^2}$ based on the scalability study in [23] (<65% PDR after 200 nodes per channel in a 7.5 km cell). We use three clusters with the PDR requirements (97%, 90% and 70%) motivated in Section 3.4.1 on page 36. Devices assignment to clusters (10%, 30% and 60%, respectively) follows the percentages proposed in [53].

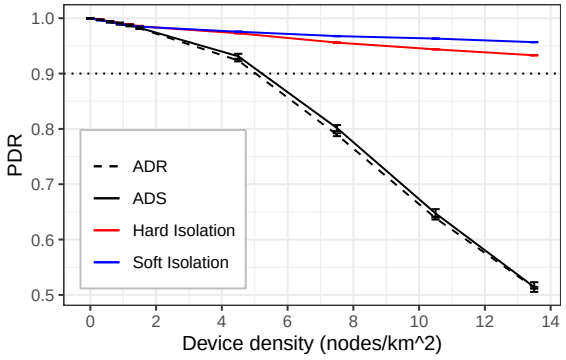
For comparison, we implement the technique for channel allocation proposed in [53], called Adaptive Dynamic Slicing (ADS). In terms of PDR optimization, ADS has proven to be comparable to the other proposals [55, 72]. As a baseline, we also implement the classical ADR configuration currently used in LoRaWANs [39]. While we commonly express PDR values in percentage (0-100%), they are left in decimal form (0.0-1.0) in the following figures.

4.3.2.1 PDR Compliance of Clusters

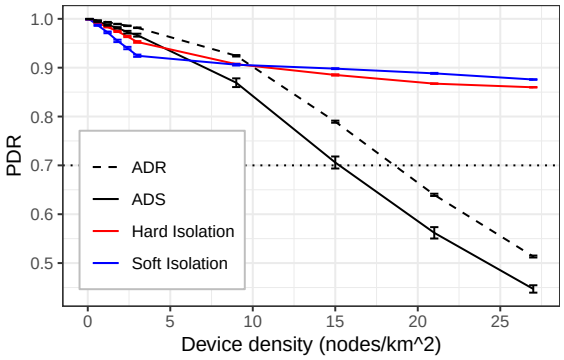
We measure the PDR of traffic for each cluster over the density of devices they serve. The density range of each cluster reflects the percentage of input devices assigned to them.



(a) Cluster 1 (97% PDR target).



(b) Cluster 2 (90% PDR target).



(c) Cluster 3 (70% PDR target).

Figure 4.2: PDR performance of devices in different clusters. The device densities for each plot is related to the number of devices of the associated cluster.

Measurements for the 97% PDR cluster are displayed in Figure 4.2a. Soft Isolation shows better performance and it is able to comply with the PDR requirement up to the maximum density. Hard isolation goes under 97% PDR after 2 $^{\text{nodes}}/_{\text{km}^2}$. ADS is able to outperform ADR but falls under 97% PDR after 1.5 $^{\text{nodes}}/_{\text{km}^2}$.

Measurements for the 90% PDR cluster are displayed in Figure 4.2b. Both Hard and Soft Isolation remain compliant with the PDR objective, Soft Isolation shows a better performance. ADS and ADR show comparable degradation and they both go under 90% PDR after 6 $^{\text{nodes}}/_{\text{km}^2}$.

Measurements for the 70% PDR cluster are displayed in Figure 4.2c on the facing page. Hard and Soft Isolation remain over 82% PDR. From this we understand that our technique over-assigns resources at lower PDR levels. ADR and ADS fall under the 70% threshold after 21 $^{\text{nodes}}/_{\text{km}^2}$ and 17.5 $^{\text{nodes}}/_{\text{km}^2}$.

From the results we conclude that Soft Isolation is the technique obtaining best performances, by being able to maintain differentiated PDR levels in scenarios with the same network density of current state-of-the-art scalability estimates. Comparison with state-of-the-art proposals for traffic quality differentiation show the gain in terms of requirements satisfaction that can be achieved by introducing access control.

4.3.2.2 Network Throughput and Gateway Range

We measure the throughput of the network and the range of the gateways against the density of devices served by the network. The range is measured as the maximum distance from the closest gateway among served devices. In our technique the range is reduced by access control. We compare to ADR and ADS, where the range limit is a consequence of path loss. Results are displayed in Figure 4.3 and in Figure 4.4 on the following page for throughput and range respectively.

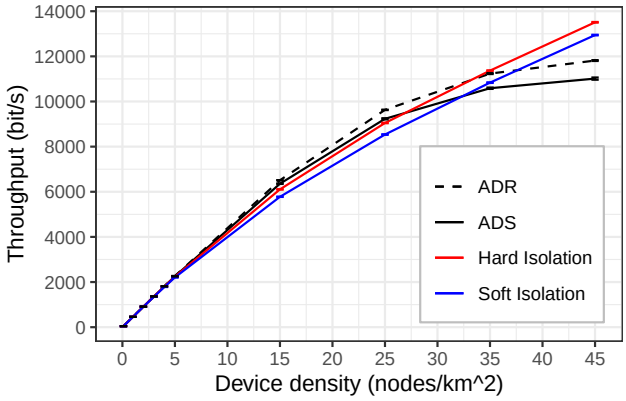


Figure 4.3: Throughput of the network under different loads.

Throughput results show that access control is able to balance out packet loss caused by higher interference in ADR and ADS, especially at high density (>40 $^{\text{nodes}}/_{\text{km}^2}$). By comparing ADS and our technique to ADR, we see that traffic differentiation generally comes at the cost of slightly lower throughput values. Also, higher precision in PDR control as in Soft Isolation results in lower

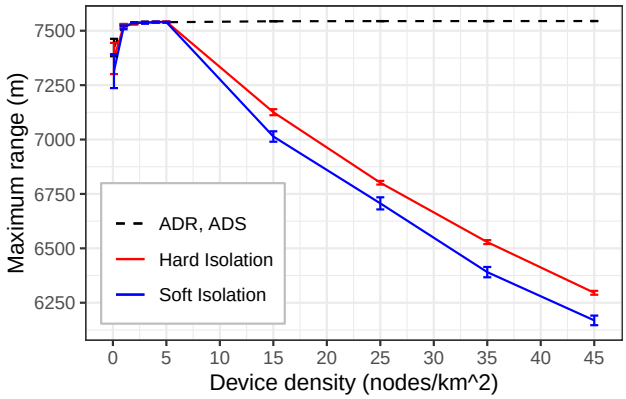


Figure 4.4: Maximum gateway range under different loads.

throughput with respect to Hard Isolation, which is instead comparable to ADS. Figure 4.4 highlights the trade-off of the proposed technique: the capacity-based policy reduces the range in which devices can be served when the density increases. Overall Soft Isolation is more expensive in terms of range, losing almost 1300 m against ADR and ADS. As a consequence, in large and dense deployments PDR differentiation comes at the expense of an higher number of gateways than best effort operation.

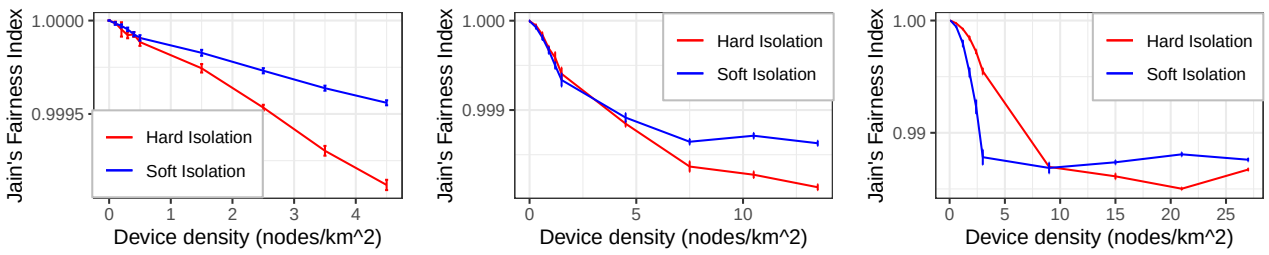
4.3.2.3 Intra-cluster Fairness

Let us now evaluate how well the PDR of clusters is distributed between their devices; in fact, even if the average PDR in a cluster is over the required level, the PDR distribution over the devices in the same cluster may still be unbalanced. We produce the Jain's Fairness Index for the devices PDR in each cluster, a common metric to evaluate whether participants are receiving a fair share of system resources. Results are displayed in Figure 4.5 on the facing page.

We can see that the level of fairness is related to the number of served devices by a cluster. This results in high PDR clusters achieving near perfect fairness and the best effort cluster being less precise. Soft Isolation is more fair and the overall fairness does not fall under 97% which means our technique is highly fair among single devices in a cluster.

4.4 Conclusion

We have proposed an approach to differentiate the PDRs of LoRaWAN device clusters under dense settings by creating independent interference domains and limiting the number of transmitting



(a) Cluster 1 (97% PDR target). (b) Cluster 2 (90% PDR target). (c) Cluster 3 (70% PDR target).

Figure 4.5: Inter-device fairness of the PDR in each cluster computed with the Jain's Fairness Index $\frac{(\sum_{i=1}^{N} x_i)^2}{N \cdot \sum_{i=1}^{N} x_i^2}$. Maximum fairness is 1 and minimum is $1/N$, with N being the number of participants and x_i the PDR of participant i.

devices with a form of access control. Our results show that the proposed approach is able of achieving PDR differentiation in high density scenarios identified by state-of-the-art scalability studies. The technique performs better than common heuristics for parameter allocation, at the expense however of decreasing the maximum cell range. Thus, we conclude that a trade-off can be found between the PDR levels to guarantee and the range of a cell. Finally, the approach presents high fairness levels in requirements satisfaction between devices.

In this chapter, we shed light on the feasibility of PDR-based, traffic quality differentiation in LoRaWAN. Now, a major future work perspective concerns evaluating the resource efficiency of this type of approaches from an operator standpoint. Another one is to consider much denser networks, as for instance in an urban environment, where gateways need to be closer apart. Furthermore, the duty-cycle tuning primitive already present in the LoRaWAN protocol could be introduced to adapt the traffic related to every device in place of access control, which is not directly supported. We set out to investigate these aspects in the following chapter.

Chapter 5

Traffic Control and Channel Assignment: Proposed Algorithms and Viability

Contents

5.1 Introduction

In this chapter, we carry out a broader analysis on the effectiveness of PDR-based quality differentiation. Compared to the previous work presented in Chapter 4, we consider dense urban LoRaWANs as well as the introduction of fairness in the spectrum allocation among clusters. We adopt three classes of services defined in terms of PDR (97%, 90%, and 70%) as motivated in Section 3.4.1 on page 36. We use disjoint sets of channels to isolate clusters of devices as done in the previous chapter and in most related works (see Section 3.4.2 on page 38).

To limit interference in clusters, we introduce and compare two traffic control solutions (via access control and duty-cycle control). Then, to evaluate the performance impact of using independent interference domains, we define three distinct frequency assignment policies (priority to high requirements, proportional-fairness, traffic maximization) spanning over the fairness spectrum. We avoid mixing devices of different classes as done in the Soft Isolation technique proposed in Chapter 4: it does not bring a substantial performance improvement, while making it more difficult to clearly partition resources for evaluation purposes. If needed, our Soft Isolation technique can be easily adapted (in the form of upgrading devices) to further improve any channel allocation.

We design our techniques to work on pre-existing transmission parameter allocations (e.g., ADR), such that further requirements and optimization objectives may be subsequently integrated. For evaluation purposes, we define several performance metrics to measure the efficiency of the proposal in terms of requirements satisfaction for devices and gain for the network operator. Numerical results show that significant improvements to PDR levels can be achieved at the expense of penalizing devices with low requirements. Also, to achieve comparable gains operators have to charge for quality differentiation with the exception of high density scenarios.

The results presented in this chapter were published in the following full conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Traffic control and channel assignment for quality differentiation in dense urban LoRaWANs," in *Proc. 20th Int. Symp. Model. and Optim. in Mobile, Ad hoc, and Wireless Netw. (WiOpt)*, 2022, pp. 153–160, doi: 10.23919/WiOpt56218.2022.9930551.

The chapter is organized as follows. We introduce our framework for packet delivery control as

well as the traffic control policies and the performance criteria used in the evaluation in Section 5.2. We detail the techniques considered for frequency allocation in Section 5.3. Simulation results are presented in Section 5.4. Concluding remarks are presented in Section 5.5.

5.2 Packet Delivery Control

Our objective is to separate and control levels of PDR for different groups of devices by means of parameter allocation schemes. For this purpose, we introduce an estimation of the PDR in a cell, followed by two methods of enforcing a desired level of PDR. Finally, we introduce two metrics to evaluate the utility of devices and the operator gain with an allocation scheme.

5.2.1 PDR Estimation for Urban Environments

Scenarios with a high density of devices are very likely to happen in various domains, notably smart cities [17, 9, 11]. Before considering the effect of interference in the calculation of PDR, fading caused by dense building environments has to be considered and is typically modeled by using Rayleigh fading. This effect has a significant impact on PDR independently of interference [43, 23]. The coverage probability is expressed as

$$\mathbb{P}_H = exp\Big(- \frac{\mathcal{N} q_j}{P g(d)}\Big), \tag{5.1}$$

where $\mathcal{N} = -117\,\text{dBm}$ is the constant thermal noise for a 125 kHz-wide band [25, 23], q_j is the SNR threshold for reception at SF j [43], P is transmission power, and $g(d)$ is average path loss at distance d.

Therefore in the following we assume that devices are placed at a maximum distance from gateways such that $\mathbb{P}_H \geq 0.98$ on SF_{12} at 14 dBm ERP power (max in Europe [33, 34]). With the Okumura-Hata path loss model for large urban environments (widely used in the literature [43]) we obtain a maximum distance of 2.5 km. Assuming ADR is used, the SNR margin for SF and TP assignment can be increased [39]. Thus, we derive the margin from Equation (5.1) to grant $\mathbb{P}_H \geq 0.98$ and benefit from the usage of multiple SFs and TP combinations.

We base our PDR estimation on the model proposed in [43], correlating the effects of fading with

co-SF interference. Depending on the SF $j = 7, \ldots, 12$, the model formula is

$$e^{-g_t - 2\nu_j} + \frac{1 + \gamma(1 - e^{\frac{1}{\gamma} - g_t})}{\gamma + 1} \cdot 2\nu_j e^{-g_t - 2\nu_j} \stackrel{\text{def}}{=} h(\nu_j), \tag{5.2}$$

where $h(\nu_j)$ is the PDR as a function of ν_j, offered traffic per frequency on SF j, $g_t = \frac{\mathcal{N} q_j}{Pg(d)}$ is the maximum thermal noise gain that allows successful reception, and $\gamma = 1\,\text{dB}$ is the difference in received power necessary to capture one of two overlapping transmissions for a given SF [70]. Under the established condition $\mathbb{P}_H \geq 0.98$, from Equation (5.1) we can set $g_t = -\ln(\mathbb{P}_H)$ for every SF.

By inverting Equation (5.2) we obtain the maximum frequency offered traffic ν for a SF to respect the desired PDR

$$h^{-1}(PDR) = -\frac{1}{2} \cdot \mathcal{W}(-\frac{\xi}{e^{\xi}} \cdot e^{g_t} \cdot PDR) - \frac{\xi}{2} = \nu_j, \tag{5.3}$$

where $\mathcal{W}$ is the Lambert function and

$$\xi = \frac{\gamma + 1}{1 + \gamma(1 - e^{\frac{1}{\gamma} - g_t})}. \tag{5.4}$$

The offered traffic of a single device is usually considered to be $\delta = 0.01$ because, with duty-cycle limitations, they cannot transmit for more than 1% of time and thus their arrival rate λ will always be $\frac{\delta}{\tau}$ with τ average transmission duration. The problem with this approach is that in real LoRaWAN scenarios devices usually transmit much less than a δ fraction of time. Thus, supposing to know a priori the bit rate β in $^{\text{bit}}/_{\text{s}}$ of a device, we can obtain a better estimate of the offered traffic with

$$\delta = \frac{p}{dr_{SF}} \lambda = \frac{\beta}{dr_{SF}}, \tag{5.5}$$

where p is the packet length in bits and dr_{SF} is the data-rate in $^{\text{bits}}/_{\text{s}}$ of the SF used by the device.

We use the above estimate for allocating resources in the proposed technique, which constrains traffic on each SF so that the total offered traffic is less than $h^{-1}(PDR)$ times the number of frequencies assigned.

5.2.2 Traffic Control Schemes

We consider two alternative techniques to constrain traffic: the first one consists of limiting the population of active devices, and the second one of uniformly lowering the duty-cycle.

5.2.2.1 Access Control

Assuming that a group of devices can transmit on m frequencies with a desired PDR value, we can formulate the following problem to maximize the global amount of traffic under the PDR constraint: for each SF $j = 7, \ldots, 12$,

$$\max \sum_{i=1}^{N^j} d_i^j \delta_i^j, \tag{5.6a}$$

$$\sum_{i=1}^{N^j} d_i^j \delta_i^j \leq m \cdot h^{-1}(PDR), \tag{5.6b}$$

$$d_i^j \in \{0, 1\}, \qquad i = 1, \ldots, N^j, \tag{5.6c}$$

where N^j is the number of devices using SF j, δ_i^j is the duty-cycle of device i using SF j, and d_i^j is a binary variable that indicates whether a device is enabled to transmit. This is a subset sum problem, which is known to be NP-hard and to have pseudo-polynomial time approximated solutions in N (number of variables) and the solution's precision [73].

5.2.2.2 Duty-Cycle Control

Alternatively, we consider a solution exploiting the LoRaWAN primitives to constrain the duty-cycle of devices. The parameter `MaxDutyCycle` $= [0 : 15]$ sets the maximum duty-cycle to $\delta = 1/2^{\texttt{MaxDutyCycle}}$. In the EU 863-870 MHz band, only values between 7 and 15 are useful because they yield $\delta < 1\%$, from 0.0078 to $3.05 \cdot 10^{-5}$.

We lower the maximum duty-cycle of all devices using SF j to the same value δ^j as follows: for each $j = 7, \ldots, 12$,

$$\delta^j = \max \left\{ \delta \in \overline{\delta} : \delta \leq \min \left\{ \delta_{\max}^j, \frac{m \cdot h^{-1}(PDR)}{N^j} \right\} \right\}, \tag{5.7}$$

where $\overline{\delta} = \left\{ \delta_{\max}^j, \frac{1}{2^7}, \ldots, \frac{1}{2^{15}} \right\}$ and $\delta_{\max}^j$ is the maximum offered traffic value among the devices using SF j. The latter is introduced as an upper bound indicating no duty-cycle limitation.

5.2.3 Performance Metrics

In the following we present two performance metrics to evaluate the device utility and the total operator gain, respectively. The former is built on the amount of expected resources, and the latter as the portion of network resources directed towards the satisfaction of requirements.

5.2.3.1 Device Utility

To understand the satisfaction of devices with a resource allocation we model their utility. The utility of a device is the ratio of the amount of radio resources received to the expected amount of resources, which depends on the PDR specified in the SLAs and the device maximum offered traffic δ.

Maximum utility of a device is achieved when the actual PDR^* of the device is equal or greater than the PDR requirement of its cluster. Otherwise, we model utility as the ratio of radio resources used at PDR^* to the resources needed for PDR. The exponential decrease of resource needs at low PDRs introduces a strong penalty for not complying with SLA when compared, for instance, with modeling utility as the direct ratio of PDR^* to PDR. Also, under duty-cycle control described in the previous section, some devices have lower offered traffic δ^* than the one they expected, δ, and consequently use less resources.

Radio resources are estimated with the capacity model $h^{-1}(\cdot)$ and expressed in terms of bandwidth. There is no direct way of obtaining the bandwidth occupied by a single device because in LoRaWAN multiple devices concurrently transmit on the same bandwidth with different traffic patterns. We derive the fraction of bandwidth b traceable to each device from the proportion of device offered traffic to maximum carried traffic on a same-PDR bandwidth:

$$b = \frac{\delta}{J \cdot h^{-1}(PDR)} \cdot B, \tag{5.8}$$

where δ is the device offered traffic, and $J \cdot h^{-1}(PDR)$ is the maximum carried traffic on a frequency channel of bandwidth $B = 125\,\text{kHz}$. This is obtained by considering $J = 6$ SFs, each contributing $h^{-1}(PDR)$ to the channel capacity. From Equation (5.8) we obtain b, the bandwidth of the device according to desired PDR and desired offered traffic δ, and b^*, computed from the achieved PDR^* and offered load δ^*.

We can thus define device utility u in terms of bandwidth requirement satisfaction,

$$u = \frac{\min\{b^*, b\}}{b}, \tag{5.9}$$

where again $\min\{b^*, b\}$ is the amount of bandwidth directly contributing to requirement satisfaction (the above ratio is equal to 100 % if $b^* \geq b$).

5.2.3.2 Total Operator Gain

To evaluate the impact of resource allocations on the operator, we define a metric to measure the gain it receives from the network. Assuming that the operator is fairly charging users according to their resource demand and satisfaction, we define the total gain as the sum of $u_i \cdot b_i = \min\{b_i^*, b_i\}$ over all devices $i = 1, \ldots, N$ in the network. This is the amount of well-assigned bandwidth, directly contributing to the satisfaction of requirements. Thus, it corresponds to the resources that the users are charged for by the operator. Then we divide it by the total bandwidth used by the network (B times the number of frequency channels F) to obtain

$$\eta = \frac{\sum_{i=1}^{N} \min\{b_i^*, b_i\}}{B \cdot F}, \tag{5.10}$$

a metric for resource allocation efficiency directly related to the operator gain.

With the PDR estimation of Equation (5.2) and the presented traffic control policies, our proposal consists of allocating frequencies to clusters. Different policies can be adopted to achieve this task as described in the next section.

5.3 Frequency Allocation Policies

Our proposal aims at differentiating the PDR of devices depending the cluster they belong to. Differentiation is achieved by assigning disjoint sets of frequencies to devices according to clusters present around a gateway. We group devices according to the gateway measuring the best radio conditions (SNR) of its transmissions, and we define the frequency allocation problem for the group of devices around to a gateway.

We grant a minimum level of service by assigning at least one frequency to each cluster. Formally,

$$\sum_{k=1}^{K} m_k = F \tag{5.11a}$$

$$m_k \geq 1,\ m_k \in \mathbb{N}, \qquad k = 1, \ldots, K, \tag{5.11b}$$

where K is the number of clusters, F the total number of frequencies, and m_k represents the number of frequencies assigned to cluster k.

We quantify the resource demand of clusters through the parameters w_k for $k = 1, \ldots, K$ defined

by

$$w_k = \max_{j \in \{7,\dots,12\}} \left\{ \frac{\sum_{i=1}^{N_k^j} \delta_{k,i}^j}{h^{-1}(PDR_k)} \right\}, \tag{5.12}$$

where N_k^j is the number of devices using SF j in cluster k and $\delta_{k,i}^j$ is the duty cycle of device i using SF j in cluster k. For the resources to be enough independently of the SF, we select the maximum frequency requirement among the SFs $j = 7, \dots, 12$ in a cluster. We obtain the estimated frequency requirement by dividing the total offered traffic on the SF by $h^{-1}(PDR_k)$, the SF estimated capacity on a single frequency. For each cluster k, PDR_k is the required PDR level.

In the following, we introduce three different frequency allocation policies: (i) giving priority to clusters with strict requirements, (ii) doing a proportional-fair allocation with right to resource demands of clusters, and (iii) maximizing the amount of network traffic. It is worth noting that it is possible to host sufficiently more devices at lower PDR to result in higher global traffic [43]. Intuitively, such techniques can be placed on a concave fairness curve going from being biased towards high PDR demands to favouring low resource consuming devices.

5.3.1 Priority to High Requirements

After reserving one frequency channel per cluster, we assign the remaining frequencies starting from the cluster demanding highest PDR. We select the minimum number of frequencies in order to fully satisfy the capacity demands of the considered cluster $k = 1, \dots, K$ as

$$m_k = \min \begin{cases} \lceil w_k \rceil, \\ \\ F - \sum_{k'<k} m_{k'} - (K - k), \end{cases} \tag{5.13}$$

where clusters are ordered by descending PDR.

Equation (5.13) also recursively ensures that conditions (5.11a) and (5.11b) are respected, and it is solvable in $O(K)$ time. Finally, we maximize traffic in each cluster by using the policies of Sections 5.2.2.1 and 5.2.2.2.

5.3.2 Proportional-fair Allocation

We adapt the optimization problem for proportional fairness described in [74]. In our case, we assume that clusters are charged proportionally to the expected resource consumption to serve all

devices. The optimization problem becomes:

$$\max \sum_{k=1}^{K} w_k \cdot \log m_k \tag{5.14a}$$

$$\sum_{k=1}^{K} m_k = F \tag{5.14b}$$

$$m_k \geq 1,\ m_k \in \mathbb{N}, \qquad k = 1, \ldots, K. \tag{5.14c}$$

This problem presents a non-linear objective that is usually solved with heuristic algorithms. In our case, however, the solution space size $\binom{F-1}{K-1}$, being a K-composition of F elements, is fairly small due to the limited number of frequencies (rarely $F \geq 8$, with a maximum of 18 channels fitting in the EU 863-870 MHz band at 1% duty-cycle [33]). Therefore, we can always tackle the problem directly in reasonable time (i.e., $O(2^F/\sqrt{F})$ iterations using Stirling's approximation of the factorial to bound binomial coefficients [75]). After determining m_k, $k = 1, \ldots, K$, we can optimize traffic as detailed in Sections 5.2.2.1 and 5.2.2.2.

5.3.3 Network Traffic Maximization

In this policy, we set the optimization objective to maximize the amount of network traffic, considering the effect of expected PDR on traffic. Therefore, we integrate the traffic optimization in the problem. It follows from Section 5.2.2.1, detailing access control,

$$\max \sum_{k=1}^{K} \left(PDR_k \sum_{j=7}^{12} \left(dr_j \sum_{i=1}^{N_k^j} d_{k,i}^j \delta_{k,i}^j \right) \right) \tag{5.15a}$$

$$\sum_{i=1}^{N_k^j} d_{k,i}^j \delta_{k,i}^j \leq m_k \cdot h^{-1}(PDR_k), \qquad \begin{cases} j = 7, \ldots, 12, \\ k = 1, \ldots, K, \end{cases} \tag{5.15b}$$

$$\sum_{k=1}^{K} m_k = F, \tag{5.15c}$$

$$m_k \geq 1,\ m_k \in \mathbb{N}, \qquad k = 1, \ldots, K, \tag{5.15d}$$

$$d_{k,i}^j \in \{0, 1\}, \qquad \begin{cases} j = 7, \ldots, 12, \\ k = 1, \ldots, K, \\ i = 1, \ldots, N_k^j. \end{cases} \tag{5.15e}$$

For the recall, K is the number of clusters, PDR_k is the PDR target of cluster k, m_k is the number of frequency channels assigned to k, N_k^j is the number of devices using SF j in cluster k, $\delta_{k,i}^j$ is the duty-cycle of device i using SF j in cluster k, and $d_{k,i}^j$ is a binary variable that indicates whether

the device is enabled to transmit. To obtain the total bit-rate on each SF, the cumulative offered traffic is multiplied by dr_j, the fixed data-rate of SF j. This problem is NP-complete, but given the limited number of possible frequency allocations (see, Section 5.3.2) it could be split into a number of pseudo-polynomial subset sum sub-problems (see Section 5.2.2.1).

When considering duty-cycle control (Section 5.2.2.2 on page 61), the problem can be rewritten as

$$\max \sum_{k=1}^{K} \left(PDR_k \cdot \sum_{j=7}^{12} \left(dr_j \cdot N_k^j \cdot \sum_{l=0}^{9} x_{k,l}^j \overline{\delta}_l \right) \right), \tag{5.16a}$$

$$\sum_{l=0}^{9} x_{k,l}^j \overline{\delta}_l \leq \min \left\{ \delta_{k,\max}^j, \frac{h^{-1}(PDR_k)}{N_k^j} \cdot m_k \right\}, \qquad \begin{cases} j = 7, \ldots, 12, \\ k = 1, \ldots, K, \end{cases} \tag{5.16b}$$

$$\sum_{l=0}^{9} x_{k,l}^j = 1, \qquad \begin{cases} j = 7, \ldots, 12, \\ k = 1, \ldots, K, \end{cases} \tag{5.16c}$$

$$\sum_{k=1}^{K} m_k = F, \tag{5.16d}$$

$$m_k \geq 1,\ m_k \in \mathbb{N}, \qquad k = 1, \ldots, K, \tag{5.16e}$$

$$x_{k,l}^j \in \{0, 1\}, \qquad \begin{cases} j = 7, \ldots, 12, \\ k = 1, \ldots, K, \\ l = 0, \ldots, 9, \end{cases} \tag{5.16f}$$

where $x_{k,l}^j$ are binary variables used to indicate whether a duty-cycle setting $\overline{\delta}_l$ is used on SF j of cluster k. Similarly to the previous problem, this one can be split into sub-problems by evaluating all possible frequency allocations. Then, the solution of each sub-problem can be trivially found by checking each duty-cycle setting.

5.4 Numerical Results

For evaluation purposes, we reuse our lightweight simulator for the LoRa uplink traffic physical layer first presented in Section 4.3.1 on page 50. As before, interference computations follow the state of the art model from [26]. Path loss follows the Okumura-Hata model for large urban environments with Rayleigh fading [43, 32, 23]. Solutions to integer programming problems are obtained using the CBC solver in the OR-Tools suite [76].

5.4.1 Simulation Setup

Seven gateways are placed using hexagonal tiling as previously illustrated in Figure 4.1 on page 50, where circle radius is 2540.29 m. This results in a total area of 127.88 km^2, used to obtain densities in the following sections of this chapter. Devices are uniformly placed in range of gateways, and they transmit with a periodical traffic pattern, interfering with other devices in the same and other cells. Gateways and devices are all placed at the same height of 15 m from the ground. Remaining details on device behaviour, parameters and interference are the same already provided in Section 4.3.1 on page 50.

SFs and TP are configured with ADR [39], the scheme currently implemented in most LoRaWAN deployments. We simulate the network running for 10 hours and we replicate simulations 30 times to be able to draw figures with 95% confidence intervals. We use three clusters with the PDR requirements at 97%, 90% and 70%, as motivated in Section 3.4.1 on page 36. Devices assignment to clusters at 10%, 30% and 60%, respectively, follows the one proposed in [53].

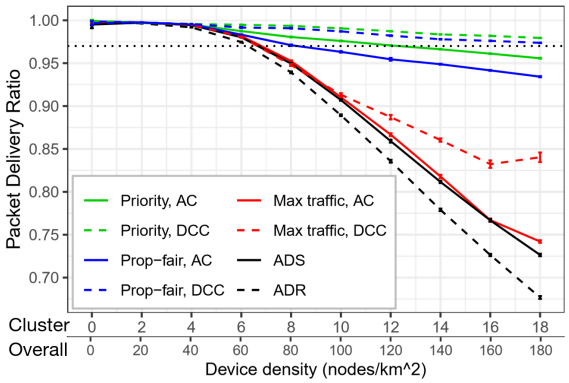
5.4.2 Result Analysis

To measure the effectiveness of our proposal, we ran simulations of the Access Control and Duty Cycle Control policies of Section 5.2 in combination with the proposed Priority, Proportional-fair, and Traffic Maximization frequency allocation policies detailed in Section 5.3. For bench-marking our solution, we implement the ADS frequency allocation proposed in [53], which does not limit traffic. Results in [55, 72] show that ADS is comparable to other proposals in terms of PDR optimization. As a baseline, we plot ADR with devices using all frequencies.

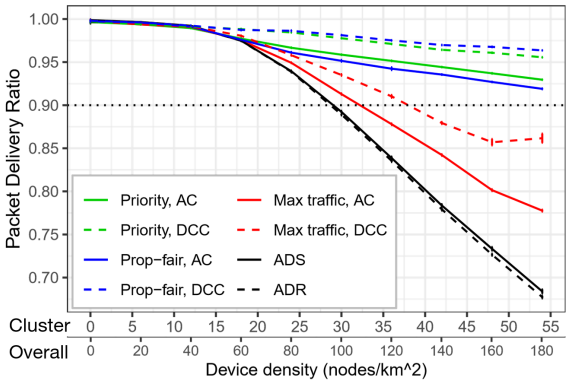
Multiple scenarios are tested, as we progressively increase the density of devices (nodes) in the network range. As we are interested in the scalability under very heavy loads, we simulate up to 180 $^{nodes}/_{km^2}$ based on the study in [23] (<50% PDR after 433 nodes per channel in a 2.5 km cell). In the following sections, metrics refer to traffic over all 7 gateways, occasionally shown per cluster in the same simulation. While we commonly express PDR values in percentage (0-100%), they are left in decimal form (0.0-1.0) in the following figures.

5.4.2.1 Packet Delivery Ratio

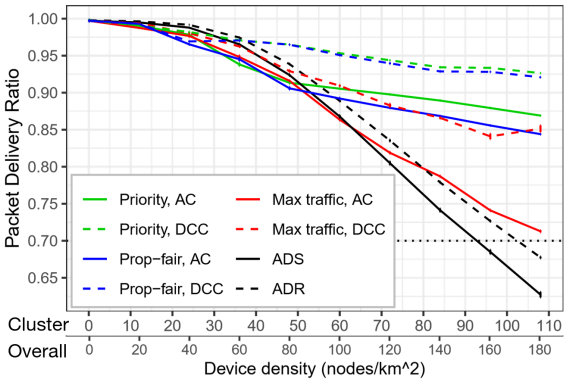
The PDR of the three clusters is shown in Figure 5.1 with their respective PDR target highlighted. The node density range of each cluster reflects the percentage of input devices assigned to them.



(a) Cluster 1 (97% PDR target).



(b) Cluster 2 (90% PDR target).



(c) Cluster 3 (70% PDR target).

Figure 5.1: Packet delivery ratio comparison, per cluster in the same simulation, for Access Control (AC) and Duty-Cycle Control (DCC) paired with the three presented frequency allocation methods. ADR [39] and ADS [53] are represented as well. Horizontal dotted black lines denote the required PDR level.

Densities above 60 $^{\text{nodes}}/_{\text{km}^2}$ are critical for the network: a difference in the performance of the various policies begins to appear at high densities. Priority and Proportional-fair allocations are actively able to prevent from quality degradation due to interference and limited reception paths, with the exception of the 97% PDR cluster, for which only with duty-cycle control they are able to satisfy the requirement at all densities. As expected, duty-cycle control is more conservative than access control because it lowers traffic in steps defined by the protocol primitive.

Table 5.1: Percentage of lost frames in Prop-fair, AC, Max traffic, AC, and ADR. Loss is caused by Interference (I), no available reception paths in a Congested gateway (C), and Under sensitivity (U) due to fading. Offered Traffic (OT) (in Erlang) is included.

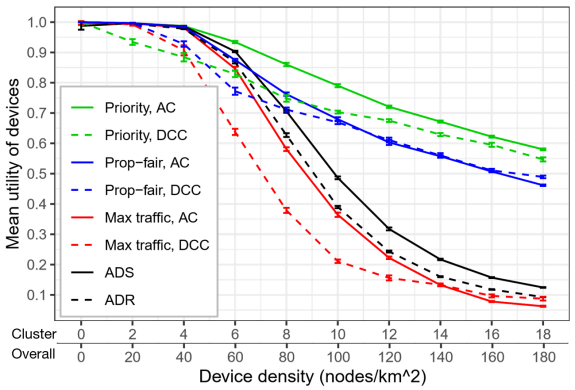
Scenario		60 $^{\text{nodes}}/_{\text{km}^2}$	120 $^{\text{nodes}}/_{\text{km}^2}$	180 $^{\text{nodes}}/_{\text{km}^2}$
Prop-fair, AC	I	3.39%	5.10%	6.21%
	C	1.16%	3.63%	5.79%
	U	0.21%	0.25%	0.27%
	OT	7.01	9.66	11.27
Max traffic, AC	I	2.13%	5.18%	7.04%
	C	1.42%	10.07%	18.81%
	U	0.22%	0.23%	0.23%
	OT	7.34	13.57	18.58
ADR	I	0.77%	3.44%	6.97%
	C	1.55%	12.41%	24.94%
	U	0.21%	0.22%	0.21%
	OT	7.51	14.96	22.60

With ADR and ADS, the PDR quickly falls down to values lower than 70% at maximum density. The results of ADS show that frequency allocation alone is not enough. Allocating frequencies with Traffic Maximization yields similar results to ADR and ADS in requirement satisfaction. We conclude that traffic control is necessary, but not sufficient to mitigate traffic quality degradation.

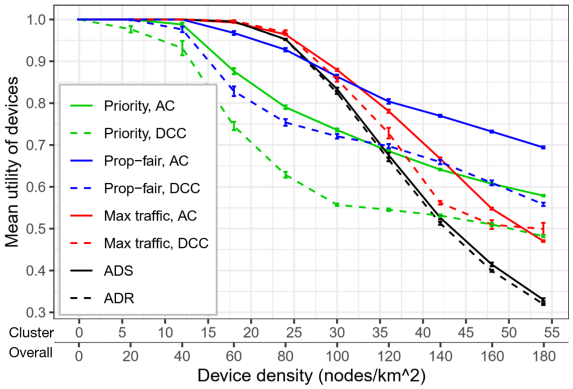
To understand why Traffic Maximization is more similar to configurations without traffic limitation, we compare causes for packet loss. As shown in Table 5.1, under heavy traffic conditions the limited number of reception paths in gateways creates a bottleneck. A higher value of carried traffic is obtained with lower PDR constraints [43], so they are favoured by Traffic Maximization. Increase of traffic in the 70% cluster for Traffic Maximization impacts the other clusters by occupying reception paths. Common implementations of LoRaWAN gateways do not allow more than 8 parallel reception paths, limiting the scalability of the technology in dense scenarios. The impact of limited reception paths is mitigated by Proportional-fair and Priority allocations.

5.4.2.2 Device Utility and Fairness

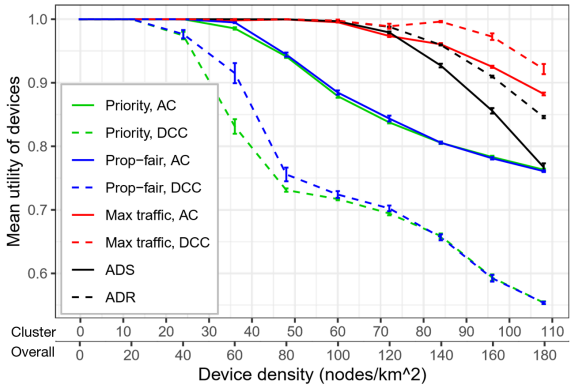
Mean devices utility is illustrated in Figure 5.2 on the following page. As expected, Priority allocations favor the high PDR cluster while Traffic Maximization favors low requirements with similar results to ADR and ADS. Globally, Proportional-fair allocation with access control is the



(a) Cluster 1 (97% PDR target).



(b) Cluster 2 (90% PDR target).



(c) Cluster 3 (70% PDR target).

Figure 5.2: Mean utility of devices, per cluster in the same simulation, for the three presented frequency allocation methods with Access Control (AC) and Duty-Cycle Control (DCC), ADR [39], and ADS [53]. Devices excluded by Access Control (AC) have 0 utility and are considered in the computation of the mean.

most balanced and is able to bring higher utility to the 97% and 90% PDR clusters than techniques without traffic control. Duty-cycle control falls behind access control; this was expected because duty-cycle control is more conservative as discussed in the PDR results section. Interestingly, with Traffic Maximization it yields the best results for low PDR requirements; this is due to the more relaxed offered traffic constraints of the cluster.

Globally speaking, Figure 5.2 on the preceding page illustrates the distortion introduced by quality differentiation in terms of utility (or satisfaction for devices). Traffic control and frequency allocation policies can preserve a rather high utility for the higher PDR clusters while the lower PDR cluster is more penalized. The ADR and ADS schemes, which cannot meet the requirements of the high PDR clusters, have the inverse impact. Thus, the price to pay to offer differentiated quality is to penalize a significant fraction of devices in order to satisfy the fraction of most demanding devices. This is viable only if the network can benefit of quality differentiation; this point is discussed in the next section. It is worth noting that the fraction of excluded devices with access control can take up to 25% of devices (Figure 5.3).

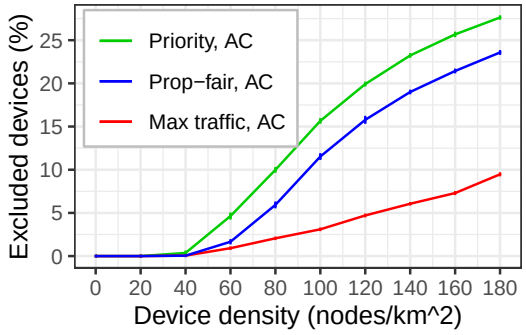


Figure 5.3: Percentage of excluded devices of the three presented frequency allocation methods with Access Control (AC).

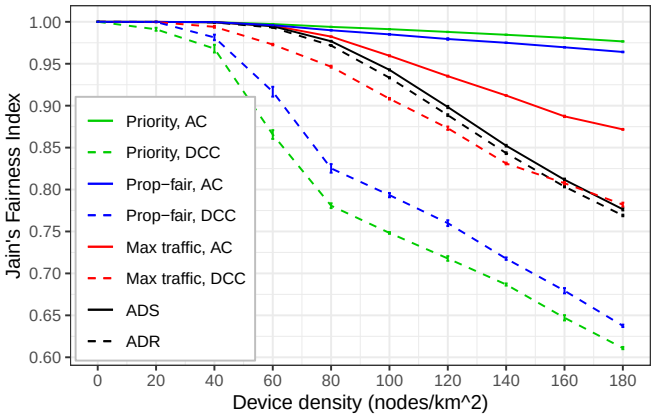


Figure 5.4: Jain's fairness index $\frac{(\sum_{i=1}^{N} u_i)^2}{N \cdot \sum_{i=1}^{N} u_i^2}$ of the utility for active devices. The three presented frequency allocation methods with Access Control (AC) and Duty-Cycle Control (DCC), ADR [39], and ADS [53] are shown.

To better understand how balanced is the utility value between devices, we evaluate the Jain's Fairness Index of the utility for active devices. Results are displayed in Figure 5.4. According to this metric, access control is reasonably fair.

5.4.2.3 Traffic Quality Differentiation Cost

We evaluate the impact of quality differentiation on the operator by showing the resource allocation efficiency metric defined in Section 5.2.3. This metric, shown in Figure 5.5, is directly related to how much the operator is charging users (resources and requirements satisfaction) and thus it is an indicator of operator gains that can be expected. We see that access control is able to maintain considerable gains that exceed ADR and ADS at higher density. This is the result of the trade-off of serving devices with better traffic quality, with Proportional-fair being the best allocation.

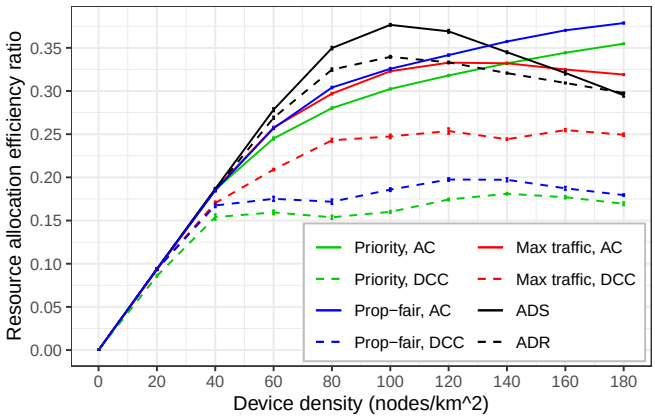


Figure 5.5: Share of resources (bandwidth) contributing to satisfaction of requirements. If the operator charges users according to demands satisfaction, higher values indicate higher gain for the operator. The three presented frequency allocation methods with Access Control (AC) and Duty-Cycle Control (DCC), ADR [39], and Adaptive Dynamic Slicing ADS [53] are shown.

We can conclude that with quality differentiation the operator uses resources less efficiently and thus needs to charge a premium to high requirement devices. Furthermore, network throughput (Figure 5.6 on the facing page) does not suffer significant degradation with access control. Energy considerations are outside of the scope of this work but can be expected to be on par with the results in [58] for ADR (therein called DA).

5.5 Conclusion

To achieve quality differentiation in dense urban LoRaWANs, we introduced various traffic control and frequency allocation policies for devices in clusters defined in terms of target PDR. It turns out that some policies can meet PDR requirements, even with high levels of PDR (97%) and high network

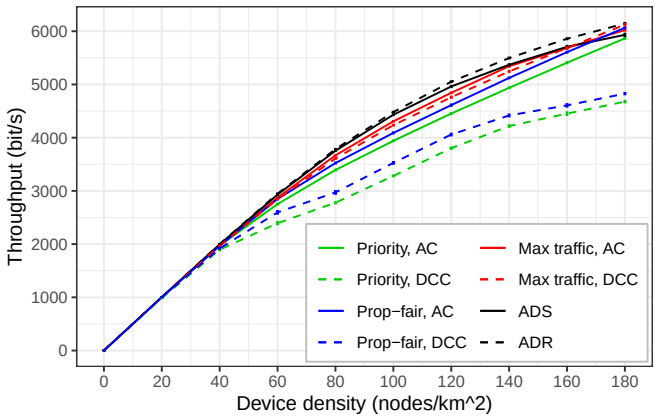


Figure 5.6: Total network throughput. The three presented frequency allocation methods with Access Control (AC) and Duty-Cycle Control (DCC), ADR [39], and ADS [53] are shown.

densities, proving the importance of traffic control. In this respect, the proposed policies perform much better (in terms of PDR and utility) than allocations introduced earlier in the technical literature without traffic control. Access control joint with proportional-fair allocation yields the best results overall.

The counterpart of introducing differentiated quality is that the utility (or satisfaction) of devices is worse for low PDR clusters than that offered by existing approaches. Yet, we find that favouring low PDR requirements is not ideal as it causes a bottleneck in the whole system. Finally, the usage of resources of the network can be lower with quality differentiation except in the case of very high device densities. This can be compensated by the network only via ad-hoc pricing. This last point can raise attractiveness issues for potential customers, making the improvement of resource utilization a crucial perspective.

Resource utilization is maximized by having a final PDR value as close as possible to the target. The techniques proposed in this chapter often underestimate the amount of offered traffic needed to meet the PDR objectives. For instance, our access control strategy in Figure 5.1c on page 68 surpasses the targeted value by 15% at a density of 180 $^{\text{nodes}}/_{\text{km}^2}$, while results for ADS [53] clearly show that a value of 70% PDR is reachable if more traffic was introduced. This is caused by the approximations present in the static capacity model we adopted to constrain the offered traffic. In the next chapter we analyze the impact of online execution, automatic estimation of the channel capacity, and changes in the pool of connected devices. Moreover, we address the drawbacks of duty-cycle tuning to be less

conservative and viably accurate.

Chapter 6

A Contribution to Adaptive Packet Delivery Ratio Management

Contents

6.1 Introduction

In Chapters 4 and 5 we qualify a trade-off between reliability and intensity of radio traffic, which can be exploited for PDR control in dense settings. In this chapter, we go one step further and address the problem of maintaining precise PDR levels by introducing a form of automated collision control. For this purpose, we develop a lightweight adaptive algorithm, based on the bisection method [77], in order to optimize the amount of traffic produced at each access point. This technique is integrated in a feedback loop that alternates between a monitoring phase and a reconfiguration phase. To constrain device traffic, we exploit the *aggregated duty-cycle* parameter for traffic shaping already present in the LoRaWAN protocol, and we develop an optimization problem to obtain a max-min configuration scheme for the system. Once convergence is achieved, the loop remains to enforce possible reconfiguration when devices appear or leave the network.

Our approach uses preexisting transmission parameter allocations so that further requirements and optimization objectives (for instance, related to energy and one-way latency) may be integrated. Using realistic parameters and a state-of-the-art simulation environment, we evaluate the convergence of our technique and we compare its accuracy and efficiency against those of earlier proposals for traffic control, which are based on a stochastic geometry model and static configurations. We also show that this approach can be used to achieve precise traffic quality differentiation for clusters of devices in a same infrastructure. Finally, we evaluate the impact of changes in the pool of connected devices via automated reconfiguration. Numerical results show that our approach is able to keep the PDR within 2% over the target value, resulting in higher carried traffic per frequency compared to other proposals. This, however, comes at the expense of having to train the algorithm beforehand.

This chapter is organized as follows. We compare our study to the state-of-the-art, in Section 6.2. We introduce our framework for collision control Section 6.3. We detail the considered simulation setup and scenarios in Section 6.4. Simulation results are presented in Section 6.5. Concluding remarks are presented in Section 6.6.

6.2 Positioning of This Contribution

In light of the review of existing work in Chapter 3, it is evident that ADR is an essential element for efficient LoRaWAN operation and that channel allocation can be used to enable differentiated

quality. On the contrary, the problem of meeting strict quality objectives as required by SLAs (for instance in the framework of network slicing) is still at an early stage of exploration.

Due to the ALOHA-like nature of LoRaWAN, the increase of traffic leads to higher collision probability [26], resulting in channel congestion and packet loss. A reasonable solution is to introduce a third step to shape traffic with the objective of limiting channel congestion, expressed by means of PDR. This yields two sub-problems: (i) finding the optimal amount of traffic that achieves a desired PDR objective, and (ii) which criteria should be used to reduce traffic among devices. Our proposal resides in this novel class of traffic control solutions.

Unlike the works presented in Chapter 3, or the techniques detailed in Sections 4.2.2.3 on page 48 and 5.3 on page 63, in this chapter our proposal for collision control does not involve the reconfiguration of channels, SF, or TP of devices. Instead, our proposal considers eachSF in an interference domain and manages the collision probability by fine tuning the duty-cycle of devices. It is designed to be applied on top of an existing channel and ADR allocation, as an additional long-term step to maintain a required PDR level.

Depending on the configuration, devices on different SFs can be subject to different levels of interference, and thus have different PDRs. When considering a permanent set of devices using a given SF (on a same set of frequencies) and transmitting according to LoRaWAN principles, we have an ergodic system and long-lived devices suffer from the same collision probability [43, 23]. This further allows us to measure device PDR by sampling the overall process, as it reflects the average one of each device thanks to ergodicity. Contrary to previous works, which considered cluster-level PDR, we therefore aim in this paper at guaranteeing a minimum PDR level for every device, by considering each SF separately.

Related work indicates that our proposal in Chapters 4 and 5 *requires to modify the behavior of end-devices, and thus does not follow the LoRaWAN specifications* [63]. However, the specifications themselves provide the *DutyCycleReq* command which *is used by the Network Server for traffic shaping, to limit the transmissions from a given end-device* [22]. The current proposal exploits this duty-cycle tuning primitive. In addition, we assume to be able to mute a device at application level allowing only once-per-day keep-alive transmissions. This simple measure of last resort is required in case duty-cycle tuning is not enough, as seen in Sections 4.3.2.1 on page 52 and 5.4.2.1 on page 68.

We aim at obtaining an accurate trade-off between input traffic and PDR targets by means of online PDR estimation, using a feedback loop based on the bisection method that performs iterative re-configuration until convergence. Furthermore, unlike previous work, we evaluate the impact of online device reconfigurations (in terms of time and packet loss), and we design our proposal to adapt to changes in the pool of devices connected to the network.

To the best of our knowledge, no other work at the state-of-the-art addresses PDR level agreement satisfaction with an online PDR estimation technique for adaptive control, independently from a specific channel and ADR configuration.

6.3 Proposed Collision Control

In a scenario where the radio coverage is good enough thanks to the placement of a sufficiently large number of radio gateways (and thus preventing devices from being in uncovered areas), degradation of the PDR (or probability of successfully transmitting a packet) is mainly a consequence of interference with other LoRa transmissions or gateways unavailability. This second case happens when a gateway is sending a downlink frame (e.g., for reconfiguration purposes in the common scenario of unidirectional traffic), or when the density of uplink frames is so high that all parallel reception paths (a.k.a. demodulators) of gateways are busy.

Given that we want to keep reconfiguration as sparse as possible, the majority of time they do not influence PDR. Instead, the limited number of demodulators can cause heavy packet loss (see, Section 5.4.2.1 on page 68) and affect full, channel-based, quality differentiation. As this is simply a present-time hardware limitation, we do not consider it in this chapter. As a matter of fact, commonly available gateways have 8 reception paths, but gateways with 16 paths are already appearing on the market, and, alternatively, there is the option of multiple co-located gateways set to listen to different frequency sets. Loss due to interference is always a possibility when traffic in the network is high.

LoRaWAN uplink traffic is usually modeled as a Poisson process, as it is the superposition of many sporadic processes. Therefore, it is often characterized in terms of offered traffic, i.e., the average frequency of transmissions times the mean volume of data sent per packet. In theory, it has been shown that on a singleSF there is a continuous and monotonic decreasing function between offered traffic and the probability for a device to successfully transmit a packet [23]. More specifically, the

success probability with self-interference is exponentially decreasing when offered traffic increases [43]. Inter-SF interference has a smaller impact when compared with self-interference, but can have a significant impact on PDR depending on the configuration [23].

If we knew the correct amount of traffic to have for each SF, we could control the impact of interference on link quality. To solve this complex issue, we propose a fast-converging iterative procedure based on the bisection method to estimate the offered traffic needed to get a desired PDR. The bisection method is a well known root-approximation technique that iteratively evaluates points of a continuous function on an interval [77]. Then, we define an optimization problem to obtain a max-min configuration of the duty-cycles of the devices for a SF given a total offered traffic value.

6.3.1 Online Offered Traffic Optimization

In the literature, mathematical models commonly establish a relationship between offered traffic and PDR [43, 23]. Some models [43] can be inverted directly (albeit with non-elementary functions) to obtain an offered traffic bound for a target PDR (see, Sections 4.2.1 on page 45 and 5.2.1 on page 59). However, due to conservative assumptions, this approach tends to underestimate the bound at which a PDR level is reached. As a consequence, high PDR values can be guaranteed but too much traffic is rejected (see, Sections 4.3.2.1 on page 52 and 5.4.2.1 on page 68). To overcome this problem, an empirical traffic bound for a desired PDR could be estimated online by re-configuring devices and continually measuring their PDR.

In LoRaWAN, we can use frame counters of transmissions to estimate the PDR during network run time. Due to long periods of time elapsed between transmissions, precise measurements could take considerable integration time. Furthermore, re-configuring a set of devices can take time as well because downlink frames can only be sent as a reply to uplink ones, and gateways are duty-cycle limited too. Therefore, we must keep the number of re-configurations as low as possible to allow fast convergence. For this purpose, we propose an online bisection method to find the offered traffic bound. This method is biased by low-quality PDR measurements, but it allows convergence in a limited number of reconfiguration cycles compared to stochastic gradient descent or more sophisticated online estimation techniques.

Let us consider static devices using a common SF j in a cell. At join time, we assume they declare to the operator the amount of data they intend to transmit over time (either an exact value for

periodic devices or an average for sporadic devices). Alternatively, we can infer it from received frames. Knowing the SF (thus, the transmission speed), we compute the maximum total offered traffic of these devices, denoted by E. We call $x_t \in [0, E]$ the offered traffic on the SF at time t, controlled by the operator with duty-cycle configurations.

Using the transmission frame counter, we periodically sample the PDR of these devices. Assuming high coverage probability, the average success probability, i.e., the PDR, depends on offered traffic x, say, as a function $f(x) \in [0, 1]$. We know that higher traffic yields lower PDR and vice-versa, but a precise definition of $f(x)$ is lacking in the technical literature. Some estimates are given for instance in [43]. We can empirically build an estimate $\widehat{f}(x_t)$ by allowing a certain amount of traffic x_t and then sampling PDR between time t and $t + 1$.

Now, consider the task of optimizing x_t such that the average success probability is in an interval $[S - \varepsilon, S + \varepsilon]$, where S is a desired PDR level and $\varepsilon > 0$. As long as $\widehat{f}(E) > S$, nothing needs to be done. Otherwise, we want to approximate the root of

$$L(x) \stackrel{\text{def}}{=} S - \widehat{f}(x) = 0. \tag{6.1}$$

When $L(E) > \varepsilon$ is measured for the first time, we set $a_0 = 0$, $b_0 = E$. The bisection method cuts in half the interval $[a, b]$ on each iteration. To speed up the process, the first assignment of x (i.e., x_0) is set to $X_0 = h^{-1}(S) \cdot m \cdot \beta$, where $h^{-1}(S)$ is formula for static configuration from Equation (5.3) (obtained by inverting the model proposed in [43]), m is the number of available frequencies and $\beta > 1$ is a multiplicative constant to avoid overshooting the PDR objective in the first reconfiguration.

Once we reconfigure the devices and sample $\widehat{f}(x_0)$, we update the variables as follows,

$$a_{t+1} = \begin{cases} x_t & \text{if } L(x_t) < 0, \\ a_t & \text{otherwise;} \end{cases} \tag{6.2a}$$

$$b_{t+1} = \begin{cases} x_t & \text{if } L(x_t) > 0, \\ b_t & \text{otherwise;} \end{cases} \tag{6.2b}$$

$$x_{t+1} = \frac{a_{t+1} + b_{t+1}}{2}. \tag{6.2c}$$

We stop when $|L(x)| \leq \varepsilon$ because we are in an acceptable range of the objective. Also, given a small tolerance value `TOL`, we stop updating if $\frac{b_t - a_t}{2} < \texttt{TOL}$ because this means that the traffic values are no more varying.

In case of changes in the pool of connected devices, we configure offered traffic to the same x_t

Algorithm 3 Bisection method for traffic optimization.

Inputs: $L(x)$, E, ε, TOL, X_0

$a_0 \leftarrow 0$, $b_0 \leftarrow E$, $x_0 \leftarrow \frac{E}{2}$
if $\frac{E-X_0}{2} \geq$ TOL **then**
    $x_0 \leftarrow X_0$
Configure offered traffic to x_0 and sample $L(x_0)$
$t \leftarrow 0$
while $\frac{b_t-a_t}{2} \geq$ TOL **or** $|L(x_t)| > \varepsilon$ **do**
    $a_{t+1} \leftarrow a_t$, $b_{t+1} \leftarrow b_t$
    if $L(x_t) < 0$ **then**
        $a_{t+1} \leftarrow x_t$
    else
        $b_{t+1} \leftarrow x_t$
    $x_{t+1} \leftarrow \frac{a_{t+1}+b_{t+1}}{2}$, $t \leftarrow t+1$
    Configure offered traffic to x_t and sample $L(x_t)$
return x_t

and sample $L(x_t)$ again. Moreover, if $\Delta x_t < 0$ (i.e., we decrease the offered traffic) and no change was made on $L(x)$, we revert to x_{t-1} and stop updating. The global procedure is summarized in Algorithm 3. This procedure is executed in parallel on all SFs. In particular, $L(x)$ sampling must happen in the same time window for all SFs to prevent the gateway unavailability during downlink transmission from tampering PDR measurements.

6.3.2 Max-min Duty-cycle Configuration

Let $N^{(j)}$ denote the number of devices using SF $j \in \{7, \ldots, 12\}$. We want to impose a total traffic $x_t^{(j)}$ offered by this set of devices. By default each device $i \in \{1, \ldots, N^{(j)}\}$ contributes a quantity δ_i to the total offered traffic, based on its behaviour. This contribution is equal to their declared bit-rate divided by the instantaneous data-rate on SF j as shown in Equation (5.5). By using protocol primitives, we can set the maximum duty-cycle of devices to any value among $\{\frac{1}{2^0}, \ldots, \frac{1}{2^{15}}\}$. In the EU868 frequency plan only values lower than 0.01 are useful. With this information we build a vector of parameters

$$\overline{\delta}_i = \left(\delta_i, \frac{1}{2^7}, \cdots, \frac{1}{2^{15}}, 0\right), \tag{6.3}$$

where $\overline{\delta}_{i,0} = \delta_i$ is the default (change nothing) setting of device i, and $\overline{\delta}_{i,l}$ with $l \in \{1, \ldots, 9\}$ are the other available duty-cycle parameter values. Moreover, in case the lowest value offered by the protocol

is not enough, we assume to be able to mute a device at application level ($\overline{\delta}_{i,10} = 0$) allowing only once-per-day keep-alive transmissions.

LoRaWAN allows configurations only as a reply to uplink frames. As a consequence, lowering the duty-cycle impacts the time it takes to reconfigure a set of devices. For this reason, we maximize the minimum duty-cycle value among devices instead of maximizing the sum. We lose in precision, while gaining in overall reconfiguration time and fairness. To address this precision loss, we introduce a straightforward maximization as a secondary objective.

We define a mixed integer linear programming (MILP) problem to maximize the minimum duty-cycle parameter among devices without breaking the offered traffic constraint $x_t^{(j)}$ as follows.

$$\max \ \theta + \frac{\alpha}{N^{(j)}} \cdot \sum_{i=1}^{N^{(j)}} \sum_{l=0}^{10} \left(y_{i,l} \cdot \overline{\delta}_{i,l}\right), \tag{6.4a}$$

$$\text{s.t. } \theta \leq \sum_{l=1}^{10} \left(y_{i,l} \cdot \overline{\delta}_{i,l}\right) + 1 \cdot y_{i,0} \qquad \forall i, \tag{6.4b}$$

$$\sum_{i=1}^{N^{(j)}} \sum_{l=0}^{10} \left(y_{i,l} \cdot \overline{\delta}_{i,l}\right) \leq x_t^{(j)}, \tag{6.4c}$$

$$\sum_{l=0}^{10} \left(y_{i,l} \cdot \overline{\delta}_{i,l}\right) \leq \delta_i \qquad \forall i, \tag{6.4d}$$

$$\sum_{l=0}^{10} y_{i,l} = 1 \qquad \forall i, \tag{6.4e}$$

$$y_{i,l} \in \{0,1\}, \ \ \theta \in \mathbb{R}^+, \ \ \alpha = 10^{-5}.$$

The parameters $y_{i,l}$ are binary variables to indicate whether duty cycle rank l is used for device i. Constraint (6.4c) limits the total offered traffic to be less than $x_t^{(j)}$, constraint (6.4d) excludes duty-cycle values higher than the default traffic of the device, and constraint (6.4e) allows one and only one duty-cycle value to be selected. Variable θ and constraint (6.4b) are introduced to linearize the max-min objective. Here, the term $y_{i,0}$ is multiplied by 1 instead of δ_i: some devices may have a very low offered traffic requirement δ_i to begin with (e.g. non periodical devices).

When maximizing the minimum duty-cycle, we may fulfill the requirements of devices early and halt the max-min optimization while the min duty-cycle among other devices could be increased further. This big constant (w.r.t. duty-cycle values, 1 is enough) is introduced to avoid considering the duty cycle of devices which have their offered traffic requirement already satisfied. This is more in line with what happens in the progressive-filling algorithm for max-min fairness. Finally, we use the small

constant α to define the secondary objective of maximizing the average offered traffic. This is done to assign as well as possible all available offered traffic, mitigating the loss in precision introduced by the max-min objective.

On the topic of computational complexity, this problem can be evaluated in two steps according to the two maximization objectives. First, the problem of maximizing the minimum duty-cycle value among devices can be solved by computing the resulting total offered traffic for each possible duty-cycle configuration in the set $\{\frac{1}{2^7}, \cdots, \frac{1}{2^{15}}, 0\}$. For each device i we need to check if the duty-cycle parameters currently under consideration is lower than its natural duty-cycle δ_i, and chose the minimum among the two. As a result, the computation is linear in $N^{(j)}$.

Once the max-min duty-cycle among devices is determined, we need to fill up the remaining offered traffic as in the secondary objective of the problem. For each device that has δ_i higher than the one found with previous max-min step, we can select one duty-cycle value equal or higher. Now, this becomes a multiple-choice subset sum problem (a special case of the better known multiple-choice knapsack problem) [78]. This class of problems is NP-hard, and it is known to have approximated solutions obtainable in pseudo-polynomial time in the number of variables, $N^{(j)}$, and the solution's precision [79].

6.4 Simulation Setup

Our proposal addresses large and dense networks; building a physical test-bed seems currently to be impossible. For evaluation purposes, we adopt the state-of-the-art simulation model further explained in Chapter 7 based on ns-3 [71]. Similarly to previous chapters, this tool implements the interference model proposed in [26], but also includes a detailed representation of the LoRaWAN MAC layer features. In our simulations, path loss follows the Okumura-Hata model for large urban environments with Rayleigh fading as in Section 5.4 on page 66. In interference calculations, we adopt the empirical Signal-to-Interference thresholds matrix shown in Table 4.4 on page 51.

Seven gateways are placed using hexagonal tiling as already illustrated in Figure 4.1 on page 50. Devices are uniformly placed in range of gateways, and they transmit with a periodical traffic pattern, possibly interfering with other devices in the same and other cells. The Okumura-Hata model takes into consideration the difference in path loss suffered by devices at different heights. To consider a

more realistic scenario, we set the gateway height equal to 30 meters, while that of devices is extracted from a uniform distribution between 1 and 10 meters.

Packet transmission time is computed by following the SX1272/73 transceiver datasheet [27]. The parameters considered for LoRa modulation are given in Table 4.2 on page 50. Devices are assigned a single frequency (868.1 MHz), with the exception of multi-cluster simulations where we give one frequency per cluster (868.3 MHz, 868.5 MHz are used in addition to 868.1 MHz). This is intended to reduce the computational load of interference; as each LoRaWAN node must transmit uniformly on assigned frequencies (frequency hopping), the offered traffic of a cluster can be scaled up proportionally with the number of frequencies to maintain the same collision probability. Therefore, our results can be generalized in terms of $\frac{\text{nodes}}{\text{km}^2 \cdot \text{frequency}}$.

Gateways are modeled by using Semtech's SX1301 chips for LoRaWAN outdoor macro-gateways [25]. As discussed in Sections 5.4.2.1 on page 68 and 6.3, the limited number of parallel reception paths (namely, 8) of these chips can cause heavy packet loss and affect full channel-based quality differentiation. This is a present-time hardware limitation; gateways with 16 demodulators already exist and can be co-located and listen to different frequencies. In our simulations, gateways have a high number of reception paths (namely, 32 corresponding to the double the current maximum) to prevent from gateway congestion.

The maximum distance from a gateway is set to 2426.85 m. As explained in Section 5.2.1 on page 59, this value is obtained from the path loss model such that the average coverage probability is higher than 98% on SF_{12} at average height from the ground. This results in a total area of 116.715 km^2, used to obtain network densities ($^{\text{nodes}}/_{\text{km}^2}$) in the following sections.

We assign the lowest possible SF to each device, in line with the default LoRaWAN ADR algorithm [39], while still granting a coverage probability of at least 98% (see, Section 5.2.1 on page 59). Figure 6.1 on the next page shows an example of the SF allocation around a gateway. Figure 6.2 on the facing page shows, for each SF, the density over distance from a gateway, and distribution of SFs among devices. These figures illustrate the fact that devices near the radio gateway have a SF equal to 7 while the assigned SF of other devices increases with the distance.

Inter-transmission time and payload of each device are the same used in Section 4.3.1 on page 50. Devices store the time and duration of their last transmission to be able to postpone the following one

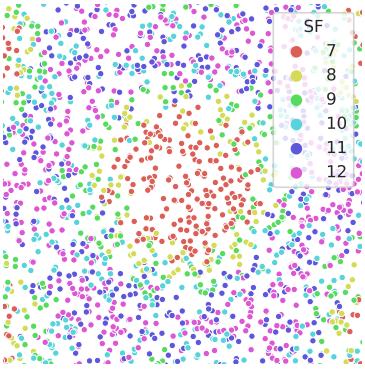


Figure 6.1: Example SF allocation for 60 $^{\text{nodes}}/_{\text{km}^2}$ to illustrate the relative positions of devices using different SFs. The square side is 5.4km, centered on the gateway G1 (see Figure 4.1 on page 50).

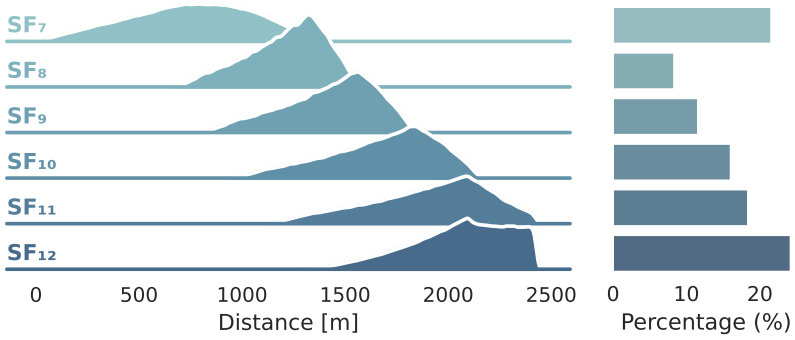


Figure 6.2: Qualitative representation of devices using a SF: where and how many? (left) Normalized density over distance from the nearest gateway. Curves are probability density functions obtained via kernel density estimation. (right) Distribution of SFs among devices. Data from 30 simulations at 200 $^{\text{nodes}}/_{\text{km}^2}$.

according to the aggregated duty-cycle setting received by the network.

Measurements (PDR, carried traffic, energy) are taken every 30 simulated minutes. Energy measurements are produced with the energy model integrated in the simulator, in its default configuration [51]. Solutions to the MILP problem are obtained using the CBC solver in the OR-Tools suite [76] with a time limit of 30s. We randomize each simulation 30 times to be able to draw figures with 95% confidence intervals.

6.5 Results

The performance of the technique proposed in Section 6.3 is evaluated for network densities ranging from 45 $^{\text{nodes}}/_{\text{km}^2}$ (medium) to 90 $^{\text{nodes}}/_{\text{km}^2}$ (high) on a single frequency [43]. In reference to ADR [39] (no collision control), for medium density, we obtain an overall network PDR of 88%, and a minimum PDR among SFs of 64% for SF_{12}; for high density, we obtain an overall PDR of 76%, with a minimum of 35% PDR for SF_{12}. It is worth noting that with 8 frequencies this is equivalent to having 360 $^{\text{nodes}}/_{\text{km}^2}$ and 720 $^{\text{nodes}}/_{\text{km}^2}$ for medium and high density, respectively. While we commonly express PDR values in percentage (0-100%), they are left in decimal form (0.0-1.0) in the following figures.

For the parameters of Algorithm 3 on page 81, we set $\varepsilon = 10^{-2}$ to accept values up to one percent

from the target PDR. The bisection tolerance is set to $\texttt{TOL} = 10^{-3}$, such that the algorithm stops when the optimization interval reaches the magnitude of a single device offered traffic. In the X_0 definition given in Section 6.3.1, we use $\beta = 16$ (obtained from testing), to avoid overshooting the target PDR with the first configuration.

The previous works discussed in Section 3 develop resource allocation techniques based on quality requirements. Among them, only a few recent ones address the problem of strict requirement compliance by means of traffic control [62] as we do in this thesis. The technique in Chapter 5, a direct improvement of the one of Chapter 4, is modular and can be compared to the approach detailed in this chapter. On the contrary, the traffic control proposal in [62] is non-modular, embedding access control in the inter-cluster resource allocation process. Therefore, a fair comparison would require pairing our technique with the same inter-cluster resource allocation strategy. Unfortunately, this is not feasible without substantial changes to our proposal; for example, we would need to give up the isolation provided by different frequencies, the ground hypothesis of our online estimation algorithm.

In the following, we first study the algorithm convergence capabilities. Then, we analyse its overall accuracy and trade-offs compared with standard LoRaWAN operation and the static traffic control model used in Chapter 5. Subsequently, we show that it can be used to differentiate traffic quality by means of multiple frequency channels. Finally, we test its ability to bring the PDR back to the target value when devices are added or removed from the network.

6.5.1 Algorithm Convergence

We first illustrate how the technique presented in Section 6.3 proceeds for an example scenario with high discrepancy between the initial state and the target PDR. In Figure 6.3 on the next page we show the evolution of the PDR in time (expressed on hour) for a device density of 90 $^{\text{nodes}}/_{\text{km}^2}$ and a collision control target of 95% PDR. SFs are shown separately because our proposal targets each one of them according to their individual performance. Here, we sample the PDR for 2 hours between each duty-cycle reconfiguration phase.

Initially, we see that SF_{11} and SF_{12} are highly congested and that SF_{10} is missing the target by 3 percents. The other SFs are not congested at the beginning. This trend reflects the lower collision probability of shorter transmissions at higher data-rates. After a 2 hour monitoring phase, the algorithm starts and the first batch of duty-cycle configurations is dispatched, as denoted by the

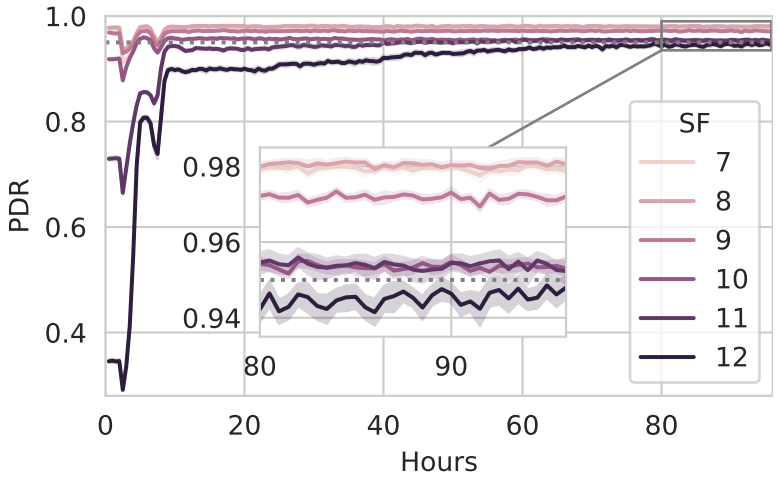


Figure 6.3: Evolution of the PDR of SFs with collision control. Example scenario with 90 $^{\text{nodes}}/_{\text{km}^2}$ and a target of 95% PDR (dotted line).

downward spike in PDR quality affecting all SFs. This temporary degradation is caused by gateways not being able to receive uplink packets while transmitting downlinks.

The PDR smoothly converges to an acceptable value after 80 hours. At the end of the third monitoring phase (10 hours), the PDR approaches the target at a slower rate. As a matter fact, downlink transmissions only happen in response to uplink packets, and the total time needed to dispatch a set of configurations increases because we are lowering their duty-cycle, that is, increasing the time between consecutive uplink transmissions. This phenomenon happens more significantly on SF_{12}, due to the higher number of devices to re-configure, and the longer time-on-air on higher SFs amplifying the effect of duty-cycle restrictions.

In Figure 6.4 on the following page, we explore how changing the duration of the PDR monitoring phase affects the convergence process of the algorithm. We plot the absolute PDR error $|L(x)|$, that is, the absolute value of the difference between measured PDR and target PDR (see Equation (6.1)). We consider that an SF has fully 'converged' whenever its absolute PDR error reaches the 0.01 threshold (1% point from the target PDR). We set the PDR target equal to 95%, and we consider two scenarios: high density (90 $^{\text{nodes}}/_{\text{km}^2}$) and medium density (45 $^{\text{nodes}}/_{\text{km}^2}$). We replicate the experiments with different lengths of the monitoring phase, that is, the period between re-configurations during which we sample the current PDR: 6 minutes (0.1 hours), 30 minutes (0.5 hours), 2 hours, and 6 hours. We show only SF_{11} and SF_{12}, as they are the ones most affected by collision control.

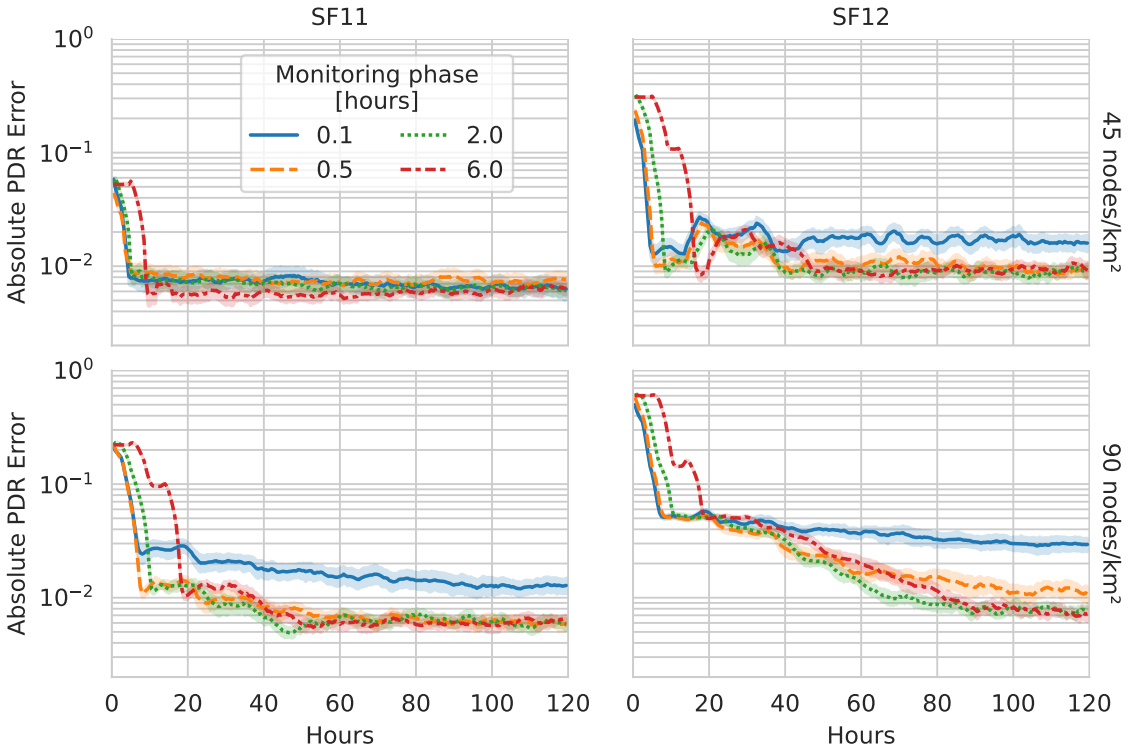


Figure 6.4: Evolution of the absolute PDR error $|L(x)|$ for different time-lengths of the sampling period during convergence of the algorithm. Vertical axis is on logarithmic scale to highlight small value ranges. PDR target is 95%, and only SFs that are strongly affected by collision control are shown (i.e., SF_{11} and SF_{12}). Tested for 45 $^{nodes}/_{km^2}$ (medium density) and 90 $^{nodes}/_{km^2}$ (high density). To increase readability, noise is smoothed in each independent set of measurements (i.e., data from one simulation) applying a gliding average of 4 hours around each point (i.e., every 30 minutes).

With a 6 minute PDR sampling window, results stabilize between 1 and 4 percents from the target, with the exception of the less congested SF_{11} for a density of 45 $^{nodes}/_{km^2}$. Similarly, on SF_{12} for 90 $^{nodes}/_{km^2}$, a monitoring phase of 30 minutes presents a higher average error than longer monitoring phases. In all other cases the procedure meets the convergence threshold, with a maximum error equal to 1.3% on SF_{12} for a density of 45 $^{nodes}/_{km^2}$. We determine that a short monitoring phase can bias the PDR value reached by the procedure.

The results corroborate the fact that higher SFs and node densities yield a longer time to convergence. For SF_{11} at 45 $^{nodes}/_{km^2}$, we can expect to reach an acceptable PDR value in 10 hours, while this time increases to 80 hours for SF_{12} with a density of 90 $^{nodes}/_{km^2}$. It must be noted that after this time, the bisection algorithm becomes 'trained' to the PDR target and is able to re-use the optimized value for offered traffic when facing changes in the pool of connected devices. Among tested values, the best trade-off is reached after a 2 hour long sampling phase, as a longer monitoring phase does not improve the precision of the result, and a shorter one introduces biases without significantly improving the time needed to converge. All following results are obtained with this setting.

6.5.2 Accuracy and Trade-offs

To assess the accuracy and trade-offs of our proposal, we explore a large parameter space: four possible network densities going from medium to high (45, 60, 75 and 90 $^{nodes}/_{km^2}$), and several values of the PDR target (1%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 75%, 80%, 85%, 90%, 95% and 97%). We measure the PDR error, the total amount of carried traffic, and the total energy consumption on each SF. In each case, we wait for the algorithm to converge, and we compute metrics on the following 24 hours of traffic.

As a baseline, we consider a scenario without any form of traffic control (NoCtrl). Furthermore, we take into account the state-of-the-art by considering the traffic control technique for PDR differentiation (TraffCtrl) proposed in Chapter 5, which, instead of the bisection method, exploits a static stochastic-geometry model [43] to produce an offered traffic value to limit packet loss. Our proposal is identified in figures as CollCtrl. Only SFs from 10 to 12 are shown, as SFs 7-9 are not impacted by collision control.

In Figure 6.5 on the next page, we show the PDR error $L(x)$ (Equation (6.1)). In some cases, no changes of duty-cycle are made and the measured PDR is higher than the targeted value. In these

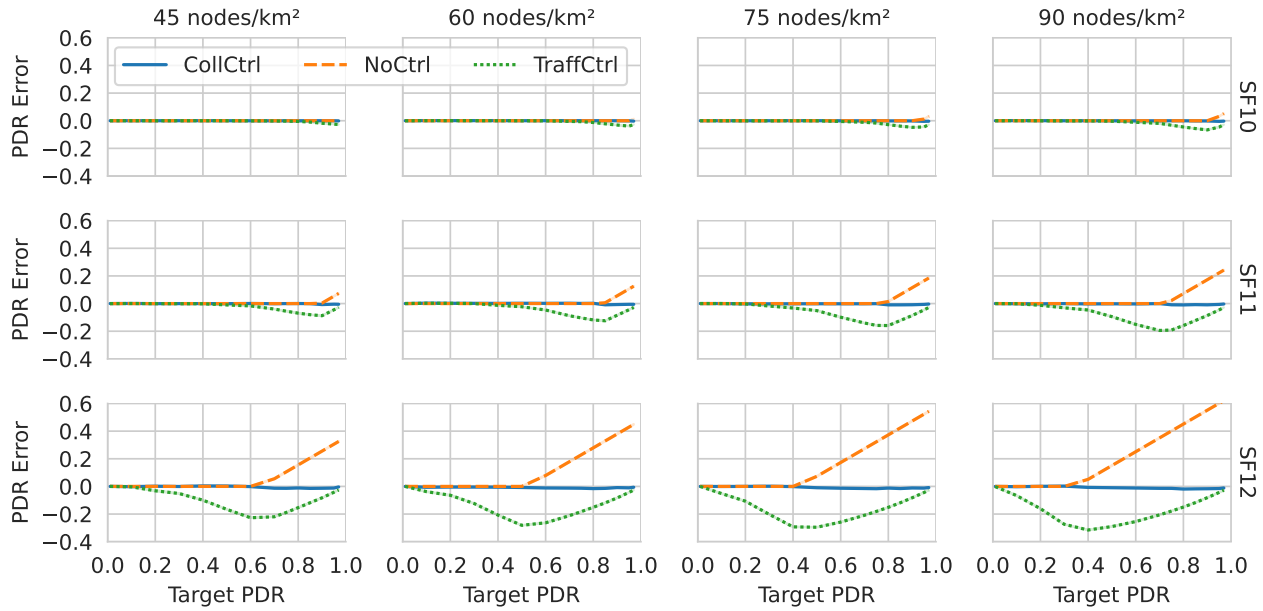


Figure 6.5: Accuracy of collision control. The PDR error $L(x)$ (Equation (6.1)) is shown against the possible PDR targets in input. The error is manually set to 0 when both (i) no changes to duty-cycle were made, and (ii) the PDR is higher than the target. Positive error values denote a PDR lower than target, negative values indicate a PDR higher than target. Relevant SFs are shown separately as rows of the grid. Tested for different input device densities on columns. The proposed technique is shown as CollCtrl. Baseline results without traffic control are shown for reference (NoCtrl). Results obtained with the technique in Chapter 5 are also shown for comparison (TraffCtrl).

situations, collision control is not necessary, therefore the PDR error is manually set to 0. From the results of CollCtrl, we see that our proposal is capable of respecting the target within a close margin (the PDR error is always between -2% and 0% with 95% confidence), thus being able to counter with high accuracy the degradation in packet loss that would normally happen with NoCtrl. When compared with our proposal, the criteria used in Chapter 5 (TraffCtrl) tends to largely exceed the target (up to 30%).

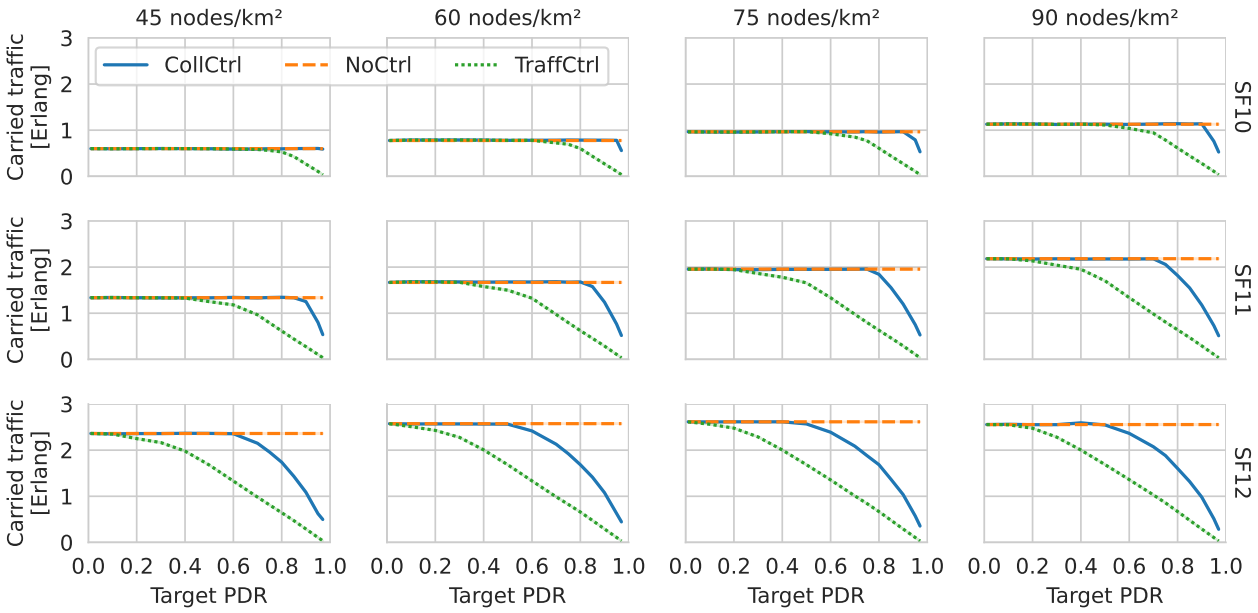


Figure 6.6: Carried traffic in Erlang shown against the possible PDR targets in input. Relevant SFs are shown separately as rows of the grid. Tested for different input device densities on columns. The proposed technique is shown as CollCtrl. Baseline results without traffic control are shown for reference (NoCtrl). Results obtained with the technique in Chapter 5 are also shown for comparison (TraffCtrl). Carried traffic is computed by multiplying the total offered traffic of devices by the PDR. In LoRaWAN literature it is also called throughput [26], or utilization [43].

In Figure 6.6, we plot the carried traffic for each group of devices. This metric is obtained by multiplying the total offered traffic of devices by the PDR [26, 43]. It measures the average number of packets carried by the radio link at any point in time. From the results, we see that the traffic carried with CollCtrl is always equal or lower than without (NoCtrl), while being always higher than the current state-of-the-art (TraffCtrl).

The only exception is on SF_{12} for a 90 $^{\text{nodes}}/_{\text{km}^2}$ density: with a target PDR of 40% the mean carried traffic is higher with CollCtrl than without (Figure 6.7 on the next page). This result is in

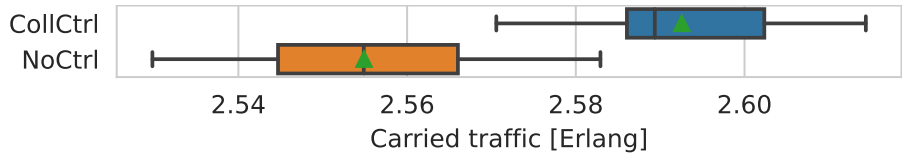


Figure 6.7: Density of 90 $^{\text{nodes}}/_{\text{km}^2}$, SF_{12}, 40% PDR target. Carried traffic of CollCtrl and NoCtrl. The mean is indicated by a triangle.

line with those in [26, 43], showing that there exists an ALOHA-like threshold (at around 40% PDR in a similar setting [43]) beyond which increasing the input traffic on the link does not improve the amount of carried traffic. After such threshold our proposal is able to improve carried traffic as well as PDR, although this is only possible in scenarios with very high packet loss.

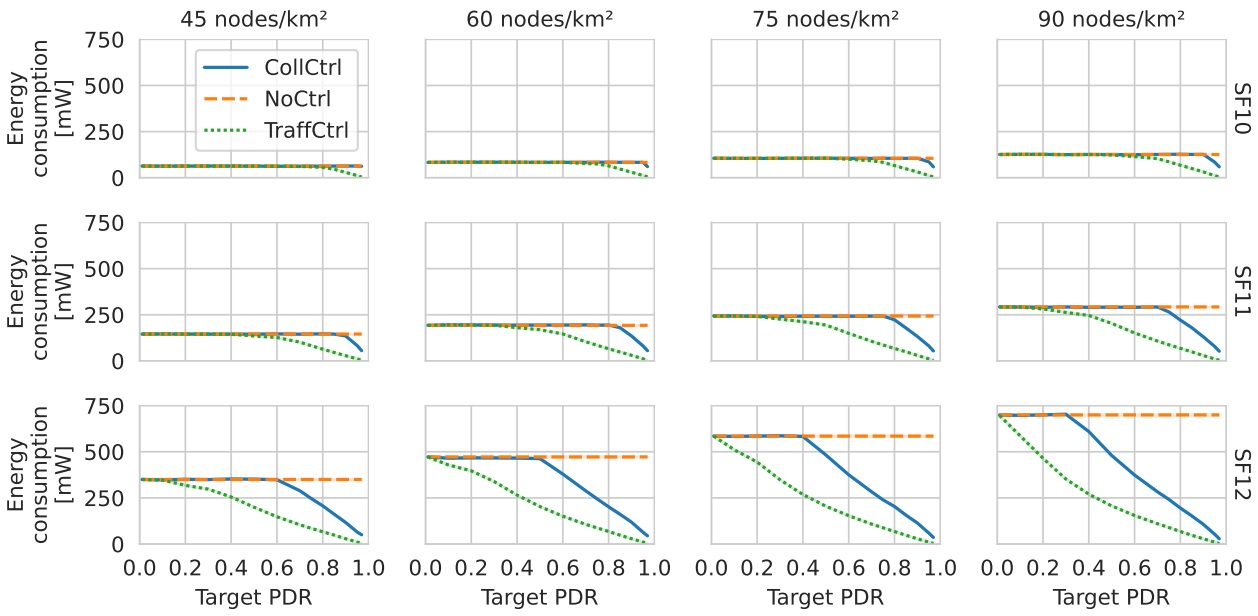
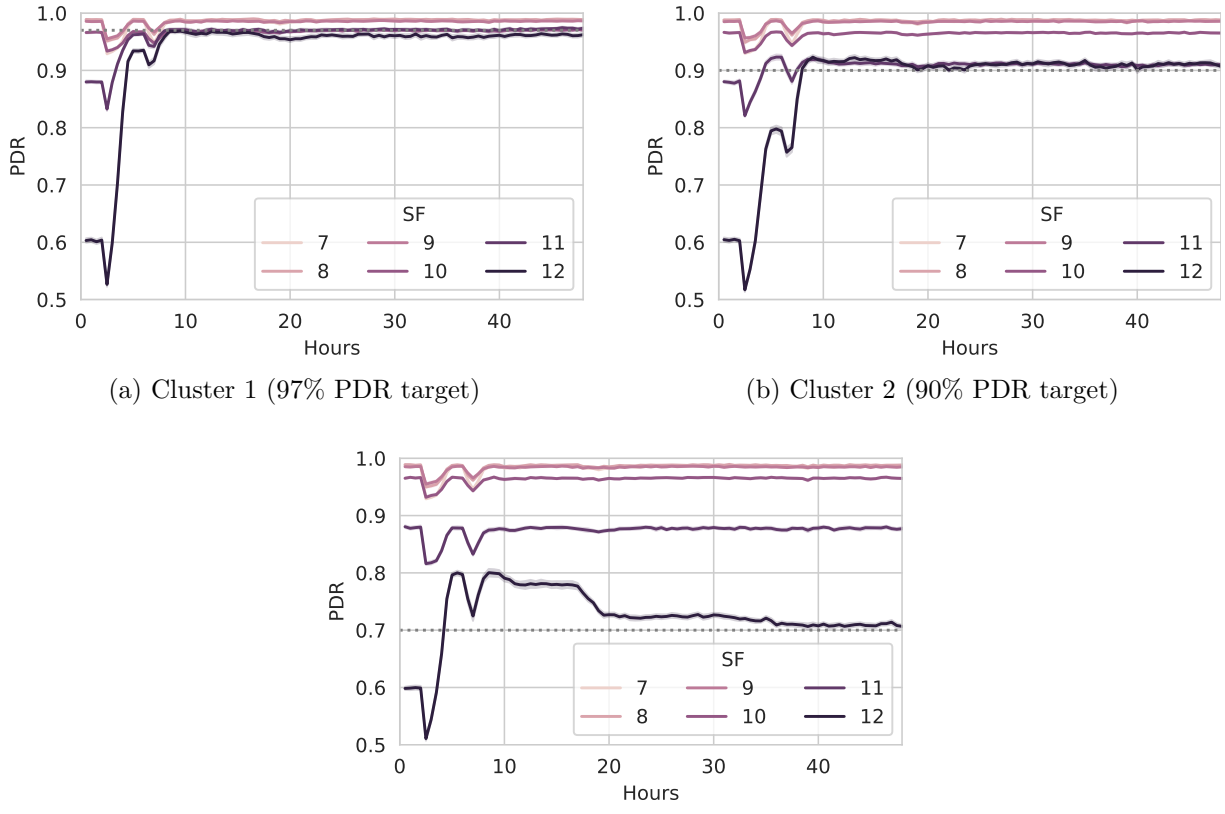


Figure 6.8: Total energy consumption of devices. Shown in mWatt against the possible PDR targets in input. Relevant SFs are shown separately as rows of the grid. Tested for different input device densities on columns. The proposed technique is shown as CollCtrl. Baseline results without traffic control are shown for reference (NoCtrl). Results obtained with the technique in Chapter 5 are also shown for comparison (TraffCtrl).

In Figure 6.8, we show the total energy consumption for each group of devices. As expected with our proposal, devices spend more time in the low-energy sleep state due to the reduced frequency of transmissions, and thus the total energy consumption is lower than with NoCtrl. The same rationale can be applied to TraffCtrl, having established with the results in Figure 6.5 on page 90 that it causes

an excessive reduction of offered traffic.

In summary, our proposal enables accurate (between -2% and 0% error) control of PDR in a wide range of scenarios and is more efficient by a significant margin (30% PDR in the worst case) with respect to the current state-of-the-art (our technique in Chapter 5). In fact, we are able to carry more traffic while meeting the same minimum PDR requirements. Nevertheless, a strong trade-off remains between carried traffic quality (i.e., the PDR) and quantity when applying collision control. Energy consumption decreases in a linear fashion when we use collision control to increase the PDR.



(a) Cluster 1 (97% PDR target)

(b) Cluster 2 (90% PDR target)

(c) Cluster 3 (70% PDR target)

Figure 6.9: Collision control applied to multi-cluster service differentiation. Each plots shows the PDR over time of a cluster comprised of different nodes in the same network. Each cluster has a device density of 50 $^{\text{nodes}}/_{\text{km}^2}$, operates on a different frequency, and has a different PDR requirement (dotted line). All devices share the same gateways.

6.5.3 Multi-cluster Service Differentiation

In this section we show how our proposal can be used to set differentiated levels of service quality among multiple clusters of devices in the same network. We place 150 $^{\text{nodes}}/_{\text{km}^2}$, and we subdivide them randomly into 3 clusters of 50 $^{\text{nodes}}/_{\text{km}^2}$ each. As commonly done in the literature [52, 53, 55, 56, 61], we set each cluster to operate on a different frequency (indicated in Section 6.4) to make them independent.

As discussed in Chapter 3, we set the minimum PDR level of the first cluster to 97% (very high), of the second cluster to 90% (high), and of the third cluster to 70% (best effort). The collision control mechanism is started simultaneously, and its effect on the PDR of each cluster is shown in Figure 6.9 on the preceding page.

From the results, we observe that gateway transmissions cause downward spikes in PDR affecting all clusters, but then only SFs that are being reconfigured have their PDR changed. Cluster 3 (Figure 6.9c on the previous page) is the slowest to converge, taking approximately 40 hours. For SF_{12} of Cluster 3, the reconfiguration slows down after 10 hours, showing that it is faster to increase the PDR rather than to decrease it. This result is in line with what we previously observed in Section 6.5.1: more infrequent uplink transmissions decrease the reconfiguration chances. Overall, we see that our proposal is able to reconfigure each cluster according to its minimum PDR requirement, confirming its effectiveness in creating services with differentiated traffic quality.

6.5.4 Robustness

In this section, we test the capability of our proposal to recover when faced with a sudden change in the pool of connected devices. For example, this could happen if a new batch of sensors is added to an existing network, or if some are removed. In the following scenarios, we consider again a single group of devices on a frequency, and the algorithm has been trained in advance to meet the PDR objectives.

In Figure 6.10 on the facing page, a network with 60 $^{\text{nodes}}/_{\text{km}^2}$ is set to meet a minimum PDR of 95%. At time 48 hours, 30 $^{\text{nodes}}/_{\text{km}^2}$ are added. This produces a degradation in PDR in the following hour (in the worst case, a 25% reduction for SF_{12}), bringing SFs 10 to 12 under the PDR objective. At 50 hours the algorithm monitoring phase ends, and a batch of configurations is sent to devices. It

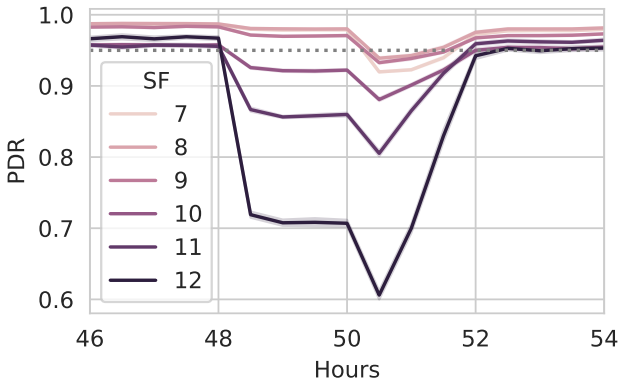


Figure 6.10: Collision control facing an increase of connected devices. The initial device density is 60 $^{\text{nodes}}/_{\text{km}^2}$ and collision control has already reached a target of 95% PDR; 30 additional $^{\text{nodes}}/_{\text{km}^2}$ are added at time 48h. A new set of configurations is dispatched after a sampling phase (2 hours).

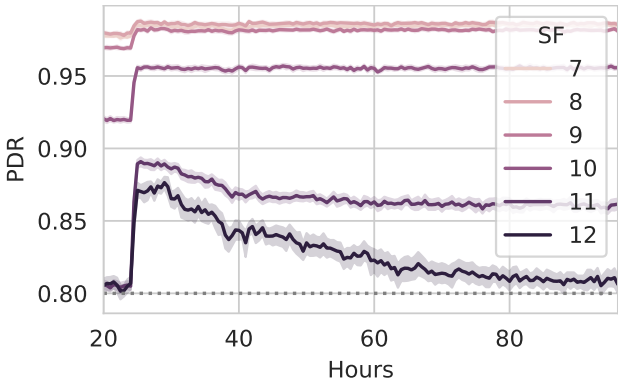


Figure 6.11: Collision control facing a decrease of connected devices. The initial device density is 90 $^{\text{nodes}}/_{\text{km}^2}$ and collision control has already reached a target of 80% PDR; 30 $^{\text{nodes}}/_{\text{km}^2}$ are removed from the network at time 24h. A new set of configurations is dispatched after a sampling phase (2 hours).

is important to recall that here the algorithm is reusing offered traffic values computed for the 95% objective during its training process. In 4 hours, the PDR is brought back up to its previous state of compliance with the objective.

In Figure 6.11, a network with 90 $^{\text{nodes}}/_{\text{km}^2}$ is set to meet a minimum PDR of 80%. At time 24 hours, 30 $^{\text{nodes}}/_{\text{km}^2}$ are removed causing an increase in PDR on all SFs. Going back to Figure 6.6 on page 91 for 90 $^{\text{nodes}}/_{\text{km}^2}$, we notice that only SF_{11} and SF_{12} had their traffic reduced by collision control to begin with. This is reflected here by the action of the algorithm increasing their traffic (thus, decreasing their PDR). This configuration is longer than in the previous scenario for the reasons discussed in the previous section (i.e., that it is faster to increase the PDR rather than to decrease it). SF_{12} becomes in range of the 80% PDR objective in approximately 60 hours. Instead, the PDR of SF_{11} does not decrease after a certain value (between 86-87%) because all its traffic is allocated back (see Figure 6.5 on page 90, where for SF_{11} at 60 $^{\text{nodes}}/_{\text{km}^2}$ collision control is not needed under 85% PDR targets).

We deduce that our proposal is able to bring back PDR compliance after a change in the pool of connected devices. In these situations, the algorithm executes a single configuration cycle and it does not need additional training steps. The time to apply these configurations varies: when PDR needs to be increased, it is a matter of few hours (for instance, 4 in the considered scenario of switching from

medium-large density to high density), or it can take multiple days when devices are removed and duty-cycles need to be increased once again.

6.6 Conclusion

In this chapter, we have addressed the problem of handling LoRaWAN IoT services with various levels of PDR. Leveraging on existing LoRaWAN primitives, we have proposed a collision control algorithm based on the bisection method. This technique enables online estimation with a minimal number of re-configurations, fitting the sparse nature of LoRaWAN traffic. Our contribution exploits a known trade-off between the amount of offered traffic in the network and the packet collision probability, specific to the ALOHA-like, random-access nature of LoRaWAN. After an initial training phase, this method fairly configures the duty-cycle values of low-power IoT devices to precisely meet a minimum level of PDR.

We have tested our proposal using a state-of-the-art model to simulate a large urban network of static devices. Numerical results show that our algorithm is able to steadily maintain the PDR of congested SFs within a close margin (always higher than target, maximum overshoot of 2%). As expected in ALOHA-like settings, increasing the PDR comes at the expense of a lower amount of traffic carried by the network. Nevertheless, our approach proves drastically more efficient (30% at best) than previous techniques. Also, we have shown that increasing the PDR brings the benefit of lower energy consumption and thus longer battery life time of devices. When correctly tuned after a training phase, the collision control algorithm converges to the target in a maximum of 80 hours in a high density scenario ($90\,^{\text{nodes}}/_{\text{km}^2}$ per frequency).

We have successfully applied our proposal to a multi-cluster scenario, and we have shown how it can be used to co-locate groups of devices having distinct minimum PDR targets. Moreover, we have shown that when the pool of connected devices experiences a substantial change (medium to high device density, and vice versa), our approach is able to bring the PDR back to compliance.

Up to this point in the thesis, we concentrated our efforts to integrate service quality differentiation in dense LoRaWAN radio access networks. This lead us to develop accurate simulation tools to test our techniques, to the extent of precisely serializing the contents of LoRaWAN packets. Now, an interesting perspective is to consider the impact of dense networks on the core LoRaWAN infrastructure. To

this end, in the following chapter we present a tool to transparently connect our radio access network model to a real infrastructure.

Chapter 7

Development of an Emulator for Massive IoT Services in Real Infrastructures

Contents

7.1 Introduction

In the growing IoT, LoRaWANs [22] have become a popular solution for environment sensing and monitoring thanks to its cheap and easy-to-operate nature. Due to the unavailability of large-scale real-life testbeds, many simulators have been developed to study LoRaWAN at scale [26, 39]. These tools aim at accurately modeling the radio performance of devices and gateways, but do not cover the core network functions, in particular LoRaWAN network servers and bridging functions. On the other hand, only basic traffic generators currently exist for network servers [80] without, however, the possibility of producing non-trivial management scenarios involving the radio access network, resource allocations and traffic anomalies.

In this chapter, we present *ELoRa*, an open source[1] software tool aimed at accurately emulating E2E LoRaWAN traffic, from device to server. *ELoRa* is built using the well known ns-3 network simulator [71], extending the LoRaWAN module presented in [26]. Among existing simulators, this module presents the most complete implementation of the LoRaWAN MAC protocol. In our proposal, we build a translation layer between the simulation, and real (UDP-encapsulated) LoRaWAN traffic, *de facto* enabling two-way real-time communications with the outside environment. The resulting UDP traffic is transparently accepted by most LoRaWAN servers, as we re-implement in ns-3 the gateway packet forwarder protocol [28] developed by Semtech, which patented the LoRa modulation technology.

In our proposal, we focus on the well established ChirpStack open-source LoRaWAN network server [30]. To enable the server's full device management capabilities, we include a component exploiting the REST API of ChirpStack to seamlessly register devices on the server. In addition, we significantly improve the original simulator to produce traffic that is compatible with a real server. Changes include, but are not limited to, resolving multiple MAC layer inaccuracies, adding cryptographic capabilities to devices, as well as introducing a tool to emulate smart-city traffic as per Table 3.1 on page 33, and implementing the generation of LoRaWAN `.pcap` packet captures that can be dissected in programs like Wireshark [81].

Thanks to the variety of modeling tools offered by ns-3, *ELoRa* makes it possible to test management techniques on LoRaWAN servers under countless realistic scenarios and loads. Furthermore, parameters

[1] `https://github.com/non-det-alle/elora-docker`

of simulated devices can be controlled live by the server to test existing and novel radio resource management algorithms in an comprehensive E2E controlled environment. Thanks to the cloud-native architecture of Chirpstack, ELoRa can be deployed locally to the server or in a distributed fashion, standalone, in multiple instances, or accompanying other traffic flows from real networks.

The contribution presented in this chapter was published in the following demo conference paper:

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "ELoRa: End-to-end emulation of massive IoT LoRaWAN infrastructures," in *Proc. 2023 IEEE/IFIP Netw. Operations and Manage. Symp. (NOMS)*, 2023, pp. 1–3, doi: 10.1109/NOMS56928.2023.10154373.

The chapter is organized as follows. The system architecture and contribution are illustrated in Section 7.2. We detail an example scenario emulated with *ELoRa* in Section 7.3. Concluding remarks are presented in Section 7.4.

7.2 System Architecture

ELoRa is a discrete-event simulation tool running in real-time and interacting with other processes in the operating system. It takes benefit of the typical ns-3 object-oriented C++ workflow, requiring the specification of all simulated elements in a main file that is then compiled and executed by command line. As usual for ns-3 simulations, it runs under GNU/Linux as a single-threaded process. The architecture of a Chirpstack deployment with *ELoRa* is shown in Figure 7.1. In the following, we

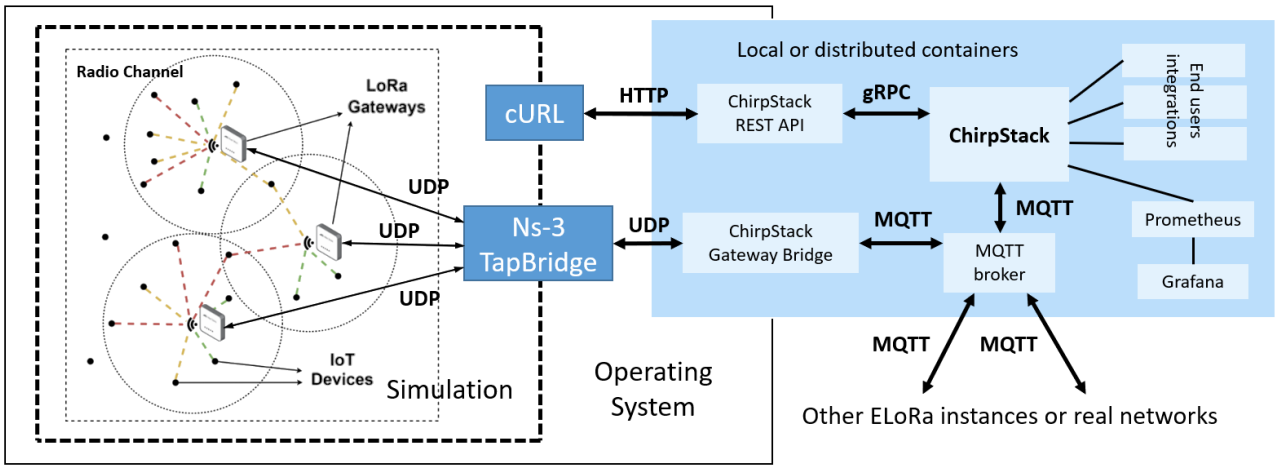


Figure 7.1: Software architecture of an *ELoRa* deployment (*left*) on Chirpstack (*right*).

detail the different elements, from left to right.

The simulated radio channel is defined in terms of path loss and delay, picking up one out of the numerous models offered by ns-3 [82]. Simulated IoT devices and gateways are placed in a three dimensional space. Position and mobility can also be set with several models, such as for instance, hexagonal tiling for gateways, and uniformly in range for devices as shown in Sections 4.3.1 on page 50 and 6.4 on page 83. In our implementation, devices can be set to send periodical or Poisson traffic, with different payload sizes. We include an allocation helper for these parameters to emulate smart-city traffic as proposed in Table 3.1 on page 33.

Currently the module supports Class A devices, the baseline LoRaWAN type, in the EU868 region. Devices comply with the frame format defined in the specifications [34, 22]. When compared with the original simulator, devices now implement all Class A mandatory MAC primitives. Using the cryptographic libraries from [83], we add message integrity code computation on uplink frames and the possibility to decrypt payloads from the server. Also, we introduce the option to export `.pcap` files of bidirectional traffic from the perspective of any device or gateway. In addition to the metrics produced by the ns-3 tracing system (e.g., packet delivery ratio, packet loss causes, channel utilization, energy consumption), such files can be used to examine live and in detail the serialized content of headers to show which MAC commands are exchanged.

All devices and gateways send frames through the same radio channel using multiple frequencies and modulation parameters. All interference computations happen according to the model in [26], with the added possibility of using the SIR matrix from [70]. The different transmission parameters used are taken into account by the interference model. In the original simulator, interference between uplink and downlink frames is computed as if it was uplink on uplink. It has been shown that downlink transmissions retain an average 90% PDR (between 85% and 99% PDR with 95% confidence) in the scenario of equal-power, concurrent uplink traffic, whereas uplink on uplink yields a lower 50% to 75% PDR [31]. For more accurate results, and in lack of a specific interference model, we consider downlink and uplink transmissions to be independent in terms of interference. Still, we maintain the assumption that gateways cannot receive uplink frames while busy transmitting downlink.

Uplink frames can be received by one or multiple gateways, which forward received traffic to core functions. The original module had a high level model of both the forwarding and the server. In our proposal, we implement in ns-3 the UDP packet forwarder protocol [28], integrating most libraries used

in real gateways to communicate with servers. This lightweight application encapsulates LoRaWAN frames in UDP, and manages the synchronization of downlink transmissions from the server with reception windows of devices.

The UDP packets are sent to a final node using one of the connection models offered by ns-3, for instance 'IP over CSMA', and then they exit the simulation. This is made possible by the ns-3 `TapBridge` class on the final node. Ns-3 already serializes packets as if they were real ones, so the `TapBridge` node creates a tap interface in the underlying operative system and takes care of translating bidirectional traffic between the simulation and the operative system. Here, a ChirpStack Gateway Bridge can be deployed locally, or traffic can be forwarded to another machine hosting one. From this point, traffic enters the ChirpStack server infrastructure.

ChirpStack architecture is composed of different components that are generally deployed as containers. Components communicate with secure protocols (MQTT [29], gRPC [84]), and can be easily distributed to achieve MEC goals. There can be multiple (MQTT) sources of LoRaWAN traffic, and gRPC is used to manage the server API. A separate translation component is given to provide a REST endpoint for the API. Most components expose metrics that can be exported with Prometheus [85] and observed in Grafana [86]. Finally, numerous end user integration schemes are available to exploit the data carried by frames or to elaborate additional metrics on the traffic.

To take full advantage of the server's network management features, i.e., to see metrics and enable parameter reconfiguration, devices and gateways need to be registered. For this reason, we develop an helper component using libcurl [87] to interact with the server's REST API. An API key can be generated directly from the server graphical interface and copied in the helper. All simulated objects are registered at the beginning of the simulation and then teared down on any interruption of the *ELoRa* process.

7.3 Example Scenario

We emulate 7 gateways positioned using hexagonal tiling at a distance of 5 km from each other; 1000 static devices are placed uniformly at a maximal distance of 2.5 km from any gateway, at an height between 1 and 10 meters. Their behaviour and payload are selected by following the distribution indicated in Table 3.1 on page 33 for commercial devices in the city of New York. Path loss is computed

Figure 7.2: The dashboard of ChirpStack.

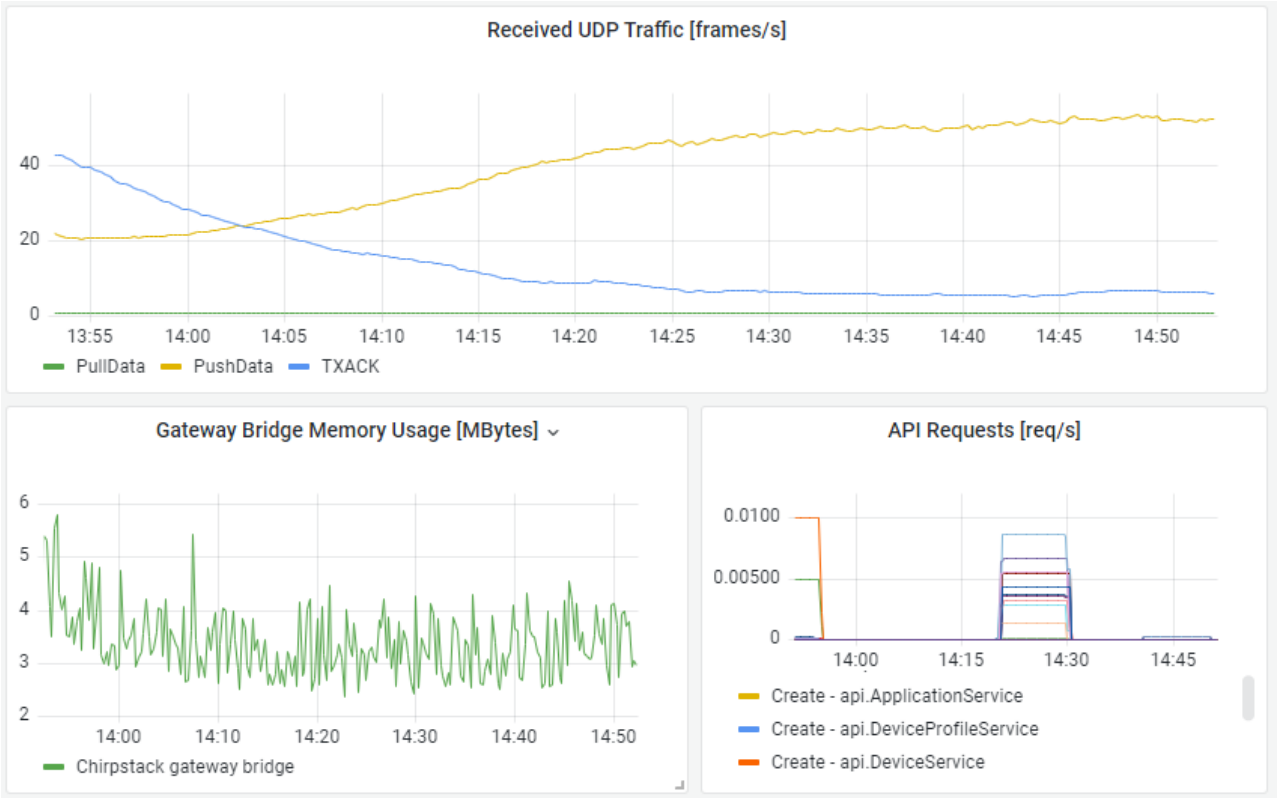


Figure 7.3: Metrics monitoring example with Grafana.

using the Okumura-Hata model for large urban areas with Rayleigh fading as in Section 5.4 on page 66.

In Figure 7.2 on the preceding page, we show the dashboard of ChirpStack when executing *ELoRa* with the described configuration. After 1 hour the server has detected more than an half of the devices, while devices with longer periodicity have only been registered. The server has increased the transmission data-rate of devices with the default ADR algorithm [39]. This helps reduce both interference and the impact of limitation on duty-cycle imposed on the EU868 band.

This is confirmed by Grafana metrics shown in Figure 7.3 on the facing page, where uplink UDP traffic increases, and the amount of acknowledgments to downlink MAC primitives decreases. The overall amount of traffic declines so the Chirpstack Gateway Bridge uses less memory. Finally, we can see the initial API calls of *ELoRa* to register simulated objects, and a second step representing internal API calls performed by the server.

In our testing, we were able to repeat this scenario with up to 50000 devices before being CPU bounded on an Intel Core i7-6600U @ 2.60 GHz processor. During this limit case, the process was exhibiting a stable 950 MB of RAM usage.

7.4 Conclusion

We have developed *ELoRa*, a tool to closely emulate LoRaWAN traffic that is capable of interacting with real server instances. *ELoRa* aims at helping research and industry actors to better understand the capabilities and limitations of LoRaWAN service infrastructures.

The flexibility of ns-3 enables emulation of real scenarios as well as the introduction of anomalies affecting the radio link and the server load. Radio parameters of simulated devices can be changed using the LoRaWAN protocol directly from the server API, allowing real-time experimentation with radio resource allocation algorithms. To our knowledge, *ELoRa* is the first tool for massive LoRaWAN simulation that presents these features, difficult to replicate with physical testbeds.

Consequently, orchestration platforms can be plugged in to test resource allocation techniques and improve the system's management automation capabilites under realistic loads. The overall framework can be easily scaled and distributed thanks to the cloud-native nature of ChirpStack.

Chapter 8

Conclusion and Research Perspectives

Contents

8.1 Summary of Contributions

Motivated by the beyond 5G evolution of network management paradigms towards network slicing and full automation, in this thesis we studied the introduction of differentiated traffic quality and reliability requirements in LoRaWAN, a prominent low-power IoT technology. To that effect, we concentrated on the RAN resource allocation problem, due to the challenges presented by its best effort operation. Moreover, packet loss quickly increases with the number of connected devices. Given the expected growth of the IoT in the years to come, our end goal is to separate classes of devices with different reliability requirements in such a way to be able to guarantee a certain minimum PDR. From the analysis carried out on the state of the art we produced three main research questions, then addressed in our contributions: (i) "Can LoRaWAN traffic differentiation be achieved with strict quality requirements?" (ii), "Is it economically viable for LoRaWAN operators and users to adopt strict quality requirements?" and (iii) "Which approaches to adopt to go towards automated quality management in LoRaWAN?"

Our first contribution is a radio resource management scheme integrating access control elements. We adopt a mathematical model from the literature to obtain the amount of offered traffic to admit on each SF such that a certain PDR is guaranteed on a single frequency channel. After, we use this quantity to integrate the PDR requirement in the channel allocation problem among clusters of devices. We propose two criteria to assign the discrete number of channels proportionally to the demands of clusters: the former has full isolation, while the latter add the possibility to upgrade devices to an higher reliability class to better distribute resources. Finally, SFs are progressively assigned to devices ordered by SNR. For each SF the allocation stops once the offered traffic capacity coming from the desired PDR is filled up. The technique is tested in a simulated multi-gateway scenario with devices assigned to three classes of PDR requirements (97%, 90%, and 70%) adapted to the categorization used for 5G MTC use cases. Results show that our method outperforms known ones in terms of PDR-satisfaction via improved parameter allocation and achieves high level of intra-cluster fairness, at the expense however of decreasing the maximum cell range.

After establishing the technical feasibility of introducing strict PDR requirements, we move onto evaluating the economical viability of this class of approaches. To do so, we design metrics for user satisfaction and operator gain, where the latter can be also seen as the resource allocation efficiency.

We consider a more realistic and dense urban scenario, and we introduce three channel allocation techniques to span over the fairness spectrum (priority to high requirements, proportional-fairness and overall throughput maximization). Moreover, aside access control, we present a first approach employing the native duty-cycle tuning primitive of the LoRaWAN protocol. Numerical results show that in high density settings we can reach a 20% better PDR with one of the proposed policies, improving mean device servicing rate by 10% and the operator gain by 7.5%. In the other cases, however, the usage of resources of the network can be lower with quality differentiation. This can be compensated by the network only via ad-hoc pricing. This last point can raise attractiveness issues for potential customers.

Our previous results mark the improvement of resource utilization as a crucial perspective towards the adoption of strict PDR-based requirements. We identify the main limitation of our previous approaches in the adoption of a static model for capacity estimation. Consequently, in our third contribution we set out to propose an adaptive collision control technique that is able to autonomously reach the offered traffic needed to achieve a desired PDR level. This approach makes use of online estimation based on the bisection method to keep the amount of needed reconfigurations for convergence at a minimum. Moreover, it improves on the duty-cycle control previously proposed, keeping the technique compatible with existing LoRaWAN primitives. Results show a drastic improvement in resource efficiency (always higher than target PDR, maximum overshoot of 2%) compared to our previous model-based approach. This comes at the expense of having to train the algorithm once, a process that can take up to 80 hours. However, once converged the approach is able to self-heal to changes in the pool of connected devices and it can be paired with a channel allocation algorithm to establish quality differentiation.

Finally, we published *ELoRa*, an open-source tool to accurately emulate LoRaWAN traffic in an E2E manner. In *ELoRa*, we build a translation layer between a well known ns-3 simulation model and real (UDP-encapsulated) LoRaWAN traffic, *de facto* enabling two-way real-time communications with the outside environment. The MAC layer and gateway packet forwarder implementation in the original LoRaWAN simulation model is greatly expanded to be transparently accepted by most LoRaWAN servers. We also introduce a tool to emulate smart-city traffic as shown in Table 3.1, we implement the generation of LoRaWAN `.pcap` packet captures aside logs and metrics to reach full RAN observability, and package the tool in a container to facilitate its deployment for automated

multi-scenario testing. Thanks to the variety of modeling tools offered by ns-3, *ELoRa* makes it possible to test management techniques on LoRaWAN servers under countless realistic scenarios and loads. Furthermore, parameters of simulated devices can be controlled live by the server to test existing and novel radio resource management algorithms in an comprehensive E2E controlled environment. Thanks to the cloud-native architecture of the LoRaWAN server stack, *ELoRa* can be deployed locally to the server or in a distributed fashion, standalone, in multiple instances, or accompanying other traffic flows from real networks.

8.2 Future Perspectives

A short term perspective for the work presented in this thesis would be the introduction of further types of quality requirements. As discussed in Chapter 5, there is a direct trade-off between overall throughput and packet reliability, where LoRaWAN in its default operation always maximizes overall throughput and minimizes energy consumption. However, a threefold PDR/throughput/energy trade-off could be modeled considering additional configuration primitives offered by the protocol. For instance, re-transmissions could be considered to increase both PDR and throughput at the expense of energy consumption [68]. If confirmed transmissions are introduced for collision detection, one-way latency could also become an interesting metric based on the number of trials needed to correctly deliver a message. Moreover, the definition of metrics could be extended to the whole infrastructure as in network slicing, for instance by multiplying reliability coefficients of nodes along the chain of RAN and core network functions.

Other short term perspectives concerning our proposed techniques are as follows. Online execution of channel allocation policies could be evaluated and then improved to adapt to changes, and always integrate soft isolation (presented in Chapter 4) in the optimization process for more fine grained partitioning of resources (channels). The collision control algorithms of Chapter 6 could be extended with ability to rollback steps of the bisection procedure if it is deployed on a new scenario and it needs some amount of re-training. Further perspectives may include the integration of mobile devices in the traffic quality differentiation framework, possibly making use of predictive techniques and preemptive allocation [41]. In the future we also plan to use *ELoRa* to reproduce other massive IoT scenarios, including anomaly detection techniques, in order to test LoRaWAN network function placement and automation solutions.

In this thesis we considered a resource allocation framework involving multiple modular sub-problems: channel allocation to clusters of devices, optimization of modulation parameters (i.e., ADR), and traffic control for reaching cluster-specific PDR targets. This distinction between different problems helps shed light on the dynamics of the system, clarifying limitations and reducing the complexity of the mathematical modeling involved in the resource allocation. The drawback of this approach is that it may reduce the space of possible solutions that would be considered in a more global (but monolithic) resource allocation technique. For instance, some intertwined approaches are presented in Chapter 4 (a combination of ADR and traffic control), in Chapter 5 and in [62] (a combination of channel allocation and traffic control), and they result in an improved 'partitionability' of radio resources. Thus, a medium term future perspective may involve the design of an all-encompassing technique for parameter allocation which is nowadays missing.

In this direction, Deep Reinforcement Learning (DRL) has proven a good tool to automatically manage the mathematical complexity that could be otherwise involved. Still, the adoption of DRL in this context remains a very open ended endeavour: the large dimensionality of the action space, combined with the LoRaWAN's sparse configuration opportunities and extremely partial observability of the radio environment (very different from the closed systems with low-latency between actions where DRL has shown relevant results [88]), could result in very long convergence time during training. A possible solution could involve partial model training in a simulated environment directed towards high generalization capabilities, with a large amount of different scenarios and randomization in all parameters. This of course presumes further improvements of current simulation models coming from the radio research community to achieve near-perfect replication of interference and noise effects. Then, the model could be deployed in a real environment for a second training step aimed at learning the specificities of that system. However, the lack of scenario repeatability in this second step could hinder the learning process. Current ongoing advancements in machine learning could help solve these problems in the future.

Important long term perspectives involve the ongoing evaluation of LoRaWAN capabilities. In this thesis we assume that gateways are duty-cycle limited as in the rest of the literature, but as discussed in Section 2.5 this may not be completely true, as by law they could use LBT + AFA (enabled by default in existing implementations) and specifications are unclear on the subject. Thus, the low viability of confirmed traffic could still be up for debate and advancements could be made

towards collision resolution techniques. For instance, the adoption of class B devices to enable forms of slotting and multi-cast configuration dispatching. At the detriment of energy consumption, class B devices could largely improve re-configuration times in our collision control proposal of Chapter 6. Unfortunately, to the best of our knowledge, no evaluation tool currently exists that integrates the mentioned features of class B devices. In the future, it could be interesting to consider adapting the algorithms proposed in this thesis to LoRaWAN networks with class B devices, or an heterogeneous mix of class A and B, and to integrate class B devices in *ELoRa* in order to study the performances that could be achieved in comparison with class A devices.

Bibliography

[1] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE access*, vol. 8, pp. 116 974–117 017, June 2020, doi: 10.1109/ACCESS.2020.3001277.

[2] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies," *IEEE Access*, vol. 9, pp. 67 512–67 547, Oct. 2020, doi: 10.1109/ACCESS.2020.3031234.

[3] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. of Commun. Soc.*, vol. 2, pp. 334–366, Feb. 2021, doi: 10.1109/OJCOMS.2021.3057679.

[4] A. Abuzaid, "Navigating 5G-era network architecture," EXFO, July 2020, Accessed: June 25, 2023. [Online]. Available: https://www.exfo.com/en/resources/blog/5g-network-architecture

[5] ETSI, "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Release description; Release 15 (3GPP TR 21.915 version 15.0.0 Release 15)," ETSI, Sophia Antipolis, France, Tech. Rep. ETSI TR 121 915 V15.0.0, Oct. 2019, Accessed: June 25, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/121900_121999/121915/15.00.00_60/tr_121915v150000p.pdf

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017, doi: 10.1109/MCOM.2017.1600951.

[7] C. Benzaid and T. Taleb, "AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions," *IEEE Netw.*, vol. 34, no. 2, pp. 186–194, Feb. 2020, doi: 10.1109/MNET.001.1900252.

[8] P. Wegner, "Global IoT market size to grow 19% in 2023 — IoT shows resilience despite economic downturn," IoT Analytics, Feb. 2023, Accessed: June 24, 2023. [Online]. Available: https://iot-analytics.com/iot-market-size

[9] T.-h. Kim, C. Ramos, and S. Mohammed, "Smart city and IoT," *Future Gener. Comput. Syst.*, vol. 76, pp. 159–162, Nov. 2017, doi: 10.1016/j.future.2017.03.034.

[10] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," in *Proc. 4th World Conf. Smart Trends in Syst., Secur. and Sustainability (WorldS4)*, 2020, pp. 449–453, doi: 10.1109/WorldS450073.2020.9210375.

[11] P. Bellini, P. Nesi, and G. Pantaleo, "IoT-enabled smart cities: A review of concepts, frameworks and key technologies," *Appl. Sci.*, vol. 12, no. 3, p. 1607, Feb. 2022, doi: 10.3390/app12031607.

[12] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016, doi: 10.1109/MWC.2016.7721743.

[13] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017, doi: 10.1109/MCOM.2017.1600613.

[14] S. Hoffenberg and C. Rommel, "The Global Market for LPWANs," VDC Research, July 2021, Accessed: June 25, 2023. [Online]. Available: https://www.vdcresearch.com/Coverage/IoT-Tech/reports/21-LPWANs.html

[15] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Jan. 2017, doi: 10.1109/LWC.2016.2647247.

[16] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proc. 19th ACM Int. Conf. Model. Anal. and Simul. of Wireless and Mobile Syst. (MSWiM)*, 2016, p. 59–67, doi: 10.1145/2988287.2989163.

[17] R. Huang, H. Li, B. Hamzeh, Y. Choi, S. Mohanty, and C. Hsu, "Proposal for evaluation methodology for 802.16p," IEEE 802.16 Broadband Wireless Access Working Group, Tech. rep. IEEE C802.16p-11/0102r2, May 2011, Accessed: June 20, 2023. [Online]. Available: https://www.ieee802.org/16/m2m/contrib/C80216p-11_0102r2.doc

[18] I. M. Martinez Bolivar, "Jamming on LoRaWAN Networks: from modelling to detection," Ph.D. dissertation, IETR, INSA Rennes, Rennes, France, Jan. 2021, Accessed: June 24, 2023. [Online]. Available: https://theses.hal.science/tel-03196484

[19] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, Dec. 2017, doi: 10.1109/ACCESS.2017.2779844.

[20] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surv. & Tut.*, vol. 19, no. 2, pp. 855–873, Jan. 2017, doi: 10.1109/COMST.2017.2652320.

[21] R. Marini, K. Mikhaylov, G. Pasolini, and C. Buratti, "Low-power wide-area networks: Comparison of LoRaWAN and NB-IoT performance," *IEEE Internet of Things J.*, vol. 9, no. 21, pp. 21 051–21 063, May 2022, doi: 10.1109/JIOT.2022.3176394.

[22] *LoRaWAN L2 1.0.4 Specification*, LoRa Alliance Spec. TS001-1.0.4, Oct. 2020, Accessed: June 24, 2023. [Online]. Available: https://lora-alliance.org/resource_hub/ts001-1-0-4-lorawan-l2-1-0-4-specification

[23] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondón, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Trans. Ind. Inform.*, vol. 15, no. 3, pp. 1425–1436, Aug. 2018, doi: 10.1109/TII.2018.2864681.

[24] N. Blenn and F. Kuipers, "LoRaWAN in the wild: Measurements from The Things Network," Jun. 2017, doi: 10.48550/arXiv.1706.03086.

[25] Semtech Corp., Wireless & Sensing Products Division, Camarillo, CA, USA. *SX1301 Datasheet*, Rev. 2.4 (2017). Accessed: Jan. 24, 2023. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-core/sx1301

[26] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *Proc. 2017 IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–7, doi: 10.1109/ICC.2017.7996384.

[27] Semtech Corp., Wireless & Sensing Products Division, Camarillo, CA, USA. *SX1272/73 Datasheet*, Rev. 4 (2019). Accessed: June 12, 2023. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1272

[28] *Lora network packet forwarder project v4.0.1.* (2017). Semtech. Accessed: Jan. 13, 2023. [Online]. Available: https://github.com/Lora-net/packet_forwarder

[29] *MQTT.* (2019). OASIS. Accessed: Jan. 13, 2023. [Online]. Available: https://mqtt.org

[30] *ChirpStack v4.* (2022). ChirpStack. [Online]. Accessed: Jan. 13, 2023. [Online]. Available: https://www.chirpstack.io

[31] R. Saroui, A. Guitton, O. Iova, and F. Valois, "Uplink and downlink are not orthogonal in LoRaWAN!" in *Proc. 96th IEEE Veh. Technol. Conf. (VTC2022-Fall)*, 2022, pp. 1–4, doi: 10.1109/VTC2022-Fall57202.2022.10012754.

[32] G. Callebaut and L. Van der Perre, "Characterization of LoRa point-to-point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet of Things J.*, vol. 7, no. 3, pp. 1910–1918, Nov. 2019, doi: 10.1109/JIOT.2019.2953804.

[33] *Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment*, ETSI EN 300 220-2, Rev. 3.2.1, June 2018, Accessed: June 24, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf

[34] *LoRaWAN Regional Parameters*, LoRa Alliance Spec. RP002-1.0.4, Sep. 2022, Accessed: June 24, 2023. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters

[35] M. Aref and A. Sikora, "Free space range measurements with Semtech LoRa technology," in *2nd Int. Symp. on Wireless Syst. within the Conf. on Intell. Data Acquisition and Adv. Comput. Syst. (IDAACS-SWS)*, 2014, pp. 19–23, doi: 10.1109/IDAACS-SWS.2014.6954616.

[36] G. Boquet, P. Tuset-Peiró, F. Adelantado, T. Watteyne, and X. Vilajosana, "LR-FHSS: Overview and performance analysis," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 30–36, Mar. 2021, doi: 10.1109/MCOM.001.2000627.

[37] J. J. Metzner, "ALOHA Protocols," in *Wiley Encyclopedia of Telecommunications*, J. G. Proakis, Ed. Hoboken, NJ, USA: Wiley, 2003, doi: 10.1002/0471219282.eot275.

[38] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on the viability of confirmed traffic in a LoRaWAN," *IEEE Access*, vol. 8, pp. 9296–9311, Jan. 2020, doi: 10.1109/ACCESS.2020.2964909.

[39] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of LoRa networks for dense IoT deployments," in *Proc. 2018 IEEE/IFIP Netw. Operations and Manage. Symp. (NOMS)*, 2018, pp. 1–9, doi: 10.1109/NOMS.2018.8406255.

[40] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," in *Proc. 13th IEEE Int. Conf. Wireless and Mobile Comput., Netw. and Commun. (WiMob)*, 2017, pp. 1–8, doi: 10.1109/WiMOB.2017.8115779.

[41] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," in *Proc. 15th Int. Wireless Commun. and Mobile Comput. Conf. (IWCMC)*, 2019, pp. 1–6, doi: 10.1109/IWCMC.2019.8766738.

[42] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, Sep. 2020, doi: 10.3390/s20185044.

[43] M. Heusse, T. Attia, C. Caillouet, F. Rousseau, and A. Duda, "Capacity of a LoRaWAN cell," in *Proc. 23rd Int. Conf. Model. Anal. and Simul. of Wireless and Mobile Syst. (MSWiM)*, 2020, p. 131–140, doi: 10.1145/3416010.3423228.

[44] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," *IEEE Commun. Surv. & Tut.*, vol. 23, no. 2, pp. 957–994, Mar. 2021, doi: 10.1109/COMST.2021.3067807.

[45] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 3, p. 1550147717699412, Mar. 2017, doi: 10.1177/1550147717699412.

[46] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges," *IEEE Commun. Surv. & Tut.*, May 2023, doi: 10.1109/COMST.2023.3274934.

[47] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surv. & Tut.*, vol. 20, no. 3, pp. 2429–2453, Mar. 2018, doi: 10.1109/COMST.2018.2815638.

[48] *Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 1: Enablers*, ETSI GS ZSM 009-1, Rev. 1.1.1, June 2021, Accessed: June 24, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/00901/01.01.01_60/gs_ZSM00901v010101p.pdf

[49] *Zero-touch network and Service Management (ZSM); Requirements based on documented scenarios*, ETSI GS ZSM 001, Rev. 1.1.1, Oct. 2019, Accessed: June 24, 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/001/01.01.01_60/gs_ZSM001v010101p.pdf

[50] Y. Ren, L. Liu, C. Li, Z. Cao, and S. Chen, "Is LoRaWAN really wide? Fine-grained LoRa link-level measurement in an urban environment," in *Proc. 30th IEEE Int. Conf. on Netw. Protocols (ICNP)*, 2022, pp. 1–12, doi: 10.1109/ICNP55882.2022.9940375.

[51] C. Delgado, J. M. Sanz, C. Blondia, and J. Famaey, "Batteryless LoRaWAN communications using energy harvesting: Modeling and characterization," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2694–2711, Feb. 2020, doi: 10.1109/JIOT.2020.3019140.

[52] S. Dawaliby, A. Bradai, Y. Pousset, and R. Riggio, "Dynamic network slicing for LoRaWAN," in *Proc. 14th Int. Conf. Netw. and Service Manage. (CNSM)*, 2018, pp. 134–142, Accessed: June 14, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8584980

[53] S. Dawaliby, A. Bradai, and Y. Pousset, "Adaptive dynamic network slicing in LoRa networks," *Future Gener. Comput. Syst.*, vol. 98, pp. 697–707, Sep. 2019, doi: 10.1016/j.future.2019.01.042.

[54] S. Dawaliby, A. Bradai, and Y. Pousset, "Network slicing optimization in large scale LoRa wide area networks," in *Proc. 2019 IEEE Conf. on Netw. Softwarization (NetSoft)*, 2019, pp. 72–77, doi: 10.1109/NETSOFT.2019.8806711.

[55] S. Dawaliby, A. Bradai, and Y. Pousset, "Distributed network slicing in large scale IoT based on coalitional multi-game theory," *IEEE Trans. Netw. and Service Manage.*, vol. 16, no. 4, pp. 1567–1580, Oct. 2019, doi: 10.1109/TNSM.2019.2945254.

[56] S. Messaoud, A. Bradai, and E. Moulay, "Online GMM clustering and mini-batch gradient descent based optimization for industrial IoT 4.0," *IEEE Trans. Ind. Inform.*, vol. 16, no. 2, pp. 1427–1435, Oct. 2019, doi: 10.1109/TII.2019.2945012.

[57] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep federated Q-learning-based network slicing for industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 17, no. 8, pp. 5572–5582, Aug. 2020, doi: 10.1109/TII.2020.3032165.

[58] S. Dawaliby, A. Bradai, and Y. Pousset, "Joint slice-based spreading factor and transmission power optimization in LoRa smart city networks," *Internet of Things*, vol. 14, p. 100121, June 2021, doi: 10.1016/j.iot.2019.100121.

[59] T. Mai, H. Yao, N. Zhang, W. He, D. Guo, and M. Guizani, "Transfer reinforcement learning aided distributed network slicing resource optimization in industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, pp. 4308–4316, Dec. 2021, doi: 10.1109/TII.2021.3132136.

[60] A. Tellache, A. Mekrache, A. Bradai, R. Boussaha, and Y. Pousset, "Deep reinforcement learning based resource allocation in dense sliced LoRaWAN networks," in *Proc. 40th IEEE Int. Conf. on Consum. Electron. (ICCE)*, Dec. 2022, pp. 1–6, doi: 10.1109/ICCE53296.2022.9730234.

[61] F. Z. Mardi, M. Bagaa, Y. Hadjadj-Aoul, and N. Benamar, "An efficient allocation system for centralized network slicing in LoRaWAN," in *Proc. 2022 Int. Wireless Commun. and Mobile Comput. Conf. (IWCMC)*, 2022, pp. 806–811, doi: 10.1109/IWCMC55113.2022.9825330.

[62] G. Dandachi and Y. Hadjadj-Aoul, "A frequency-based intelligent slicing in LoRaWAN with admission control aspects," in *Proc. 25th Int. Conf. Model. Anal. and Simul. of Wireless and Mobile Syst. (MSWiM)*, 2022, p. 189–196, doi: 10.1145/3551659.3559055.

[63] T. Bellanger, A. Guitton, R. Stanica, and F. Valois, "Everyone can slice LoRaWAN," in *Proc. 19th Int. Conf. Wireless and Mobile Comput., Netw. and Commun. (WiMob)*, 2023, pp. 375–380, doi: 10.1109/WiMob58348.2023.10187844.

[64] *Packet Broker*. (2018). The Things Industries. Accessed: June 16, 2023. [Online]. Available: https://www.thethingsindustries.com/docs/the-things-stack/packet-broker

[65] S. Delbruel, N. Small, E. Aras, J. Oostvogels, and D. Hughes, "Tackling contention through cooperation: A distributed federation in LoRaWAN space," in *Proc. 2020 Int. Conf. on Embedded Wireless Syst. and Netw. (EWSN)*, 2020, p. 13–24, doi: 10.48550/arXiv.1712.08221.

[66] B. K. Al-Shammari, N. Al-Aboody, and H. S. Al-Raweshidy, "IoT traffic management and integration in the QoS supported network," *IEEE Internet of Things J.*, vol. 5, no. 1, pp. 352–370, Dec. 2017, doi: 10.1109/JIOT.2017.2785219.

[67] L. Feltrin, C. Buratti, E. Vinciarelli, R. De Bonis, and R. Verdone, "LoRaWAN: Evaluation of link-and system-level performance," *IEEE Internet of Things J.*, vol. 5, no. 3, pp. 2249–2258, Apr. 2018, doi: 10.1109/JIOT.2018.2828867.

[68] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of LoRaWAN performance under different parameter settings," *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 116–127, Jan. 2019, doi: 10.1109/JIOT.2019.2946487.

[69] H. Fawaz, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "Joint spreading factor and channel assignment in multi-operator LoRaWAN deployments," *Sensors*, vol. 21, no. 1, p. 162, Dec. 2020, doi: 10.3390/s21010162.

[70] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 796–799, Jan. 2018, doi: 10.1109/LCOMM.2018.2797057.

[71] *Ns-3 Network Simulator v3.37*. (2008). nsnam. Accessed: Jan. 13, 2023. [Online]. Available: https://www.nsnam.org

[72] S. Messaoud, S. Dawaliby, A. Bradai, and M. Atri, "In-depth performance evaluation of network slicing strategies in large scale industry 4.0," in *Proc. 18th Int. Multi-Conf. on Syst., Signals and Devices (SSD)*, 2021, pp. 474–479, doi: 10.1109/SSD52085.2021.9429361.

[73] H. Kellerer, R. Mansini, U. Pferschy, and M. G. Speranza, "An efficient fully polynomial approximation scheme for the subset-sum problem," *J. Comput. and Syst. Sci.*, vol. 66, no. 2, pp. 349–370, Mar. 2003, doi: 10.1016/S0022-0000(03)00006-0.

[74] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, May 1998, doi: 10.1057/palgrave.jors.2600523.

[75] S. Das, "A brief note on estimates of binomial coefficients," 2016, Accessed: May 25, 2023. [Online]. Available: http://discretemath.imp.fu-berlin.de/DMI-2016/notes/binomials.pdf

[76] *OR-Tools v9.2*. (2021). Google. Accessed: July 2, 2023. [Online]. Available: https://developers.google.com/optimization

[77] G. Corliss, "Which root does the bisection algorithm find?" *SIAM Rev.*, vol. 19, no. 2, pp. 325–327, Apr. 1977, doi: 10.1137/1019044.

[78] H. Kellerer, U. Pferschy, and D. Pisinger, "The multiple-choice knapsack problem," in *Knapsack Problems*. Hidelberg, Germany: Springer Berlin, 2004, ch. 11, doi: 10.1007/978-3-540-24777-7_11.

[79] E. L. Lawler, "Fast approximation algorithms for knapsack problems," in *18th Annu. Symp. on Found. Comput. Sci. (SFCS)*, 1977, pp. 206–213, doi: 10.1109/SFCS.1977.11.

[80] *ChirpStack Simulator*. (2020). ChirpStack. Accessed: Jan. 13, 2023. [Online]. Available: https://github.com/brocaar/chirpstack-simulator

[81] *Wireshark v4*. (2022). The Wireshark Foundation. Accessed: Jan. 13, 2023. [Online]. Available: https://www.wireshark.org

[82] M. Stoffers and G. Riley, "Comparing the ns-3 propagation models," in *20th IEEE Int. Symp. Modeling, Anal., Simul. Comput. Telecommun. Syst. (MASCOTS)*, 2012, pp. 61–67, doi: 10.1109/MASCOTS.2012.17.

[83] *LoRaWAN end-device stack implementation and example projects v4.6*. (2015). Semtech. Accessed: Oct. 20, 2022. [Online]. Available: https://github.com/Lora-net/LoRaMac-node

[84] *gRPC*. (2016). Google. Accessed: Jan. 13, 2023. [Online]. Available: https://grpc.io

[85] *Prometheus*. (2016). Cloud Native Computing Foundation. Accessed: Jan. 13, 2023. [Online]. Available: https://prometheus.io

[86] *Grafana*. (2014). Grafana Labs. Accessed: Jan. 13, 2023. [Online]. Available: https://grafana.com

[87] *libcurl*. (1996). curl. Accessed: Jan. 13, 2023. [Online]. Available: https://curl.se/libcurl

[88] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

Acronyms

ADR Adaptive Data Rate. xiv, xvii, 25, 26, 38–41, 51–54, 58, 59, 67–73, 76–78, 84, 85, 105, 111, 137

ADS Adaptive Dynamic Slicing. xvii, 51–54, 67–73

AFA Adaptive Frequency Agility. xiii, 19, 24, 111, 138

API Application Programming Interface. xvi, 34, 100, 103, 105

CAGR Compound Annual Growth Rate. xv, xx, 3, 129

CSS Chirp Spread Spectrum. 15

DL down-link. xvi, 17

DRL Deep Reinforcement Learning. 111, 137

E2E end-to-end. 5, 8, 31, 34, 35, 41, 100, 101, 109, 110, 131, 135, 136

EIRP Effective Isotropic Radiated Power. 22

ERP Effective Radiated Power. 19, 22, 50, 59

FSK Frequency Shift Keying. 21

IdO Internet des Objets. vii

IoT Internet of Things. v, vi, xiii, xv, xix, xx, 2–5, 12, 13, 27, 30, 31, 36–38, 96, 100, 108, 110, 128–132, 135, 137

ISM industrial, scientific, and medical. 18

LBT Listen Before Talk. xiii, 19, 24, 111, 138

LNS LoRaWaN Network Server. 13, 14, 21, 34, 35

LoRa Long Range. xiii, xv, xvi, 12, 13, 15–19, 50, 100

LoRaWAN Long Range Wide Area Network. v–viii, xiii, xv, xvi, xviii, xx, 4–9, 12–14, 19–22, 24–27, 30, 32–42, 44, 46, 49–51, 54, 55, 58, 60–62, 67, 69, 72, 76–79, 82–84, 86, 91, 96, 100, 102, 103, 105, 108–112, 130–138

LPWAN Low Power Wide Area Network. xv, xx, 4, 12, 26, 30, 130

LR-FHSS Long-Range Frequency Hopping Spread Spectrum. 21

MAC Medium Access Control. 8, 12–14, 20–24, 26, 49, 83, 100, 102, 105, 109, 135

MEC Multi-access Edge Computing. xv, xix, 2, 103, 128

mMTC massive Machine Type Communications. 30

MTC Machine Type Communications. xiii, 5, 32, 33, 36–38, 42, 108, 131, 134

NB-IoT Narrowband IoT. xv, xx, 4, 12, 130

NFV Network Function Virtualization. xv, xix, 2, 128

PDB Packet Delay Budget. 36, 37

PDR Packet Delivery Ratio. vi–viii, xvi–xix, 5–8, 37, 38, 40–42, 44–49, 51–55, 58–65, 67–73, 76–81, 85–96, 102, 108–111, 131–137

QCI QoS Class Identifier. xiii, 36, 37

QoS Quality of Service. xiii, 3, 6, 26, 27, 31, 36–38, 42, 129, 132

RAN Radio Access Network. 5, 6, 8, 27, 32, 35, 36, 41, 108–110, 131–133, 136

RSSI Received Signal Strength Indicator. xiii, 18

SBI SouthBound Interface. xvi, 34

SDN Software Defined Networking. xv, xvi, xix, 2, 31, 32, 34, 35, 128

SF Spreading Factor. xiii, xiv, xvi–xviii, 15, 16, 18–21, 25, 26, 36, 38–40, 44–47, 49–51, 59–62, 64–67, 77–81, 84–92, 94–96, 108, 133, 134

SIR Signal to Interference Ratio. xiv, 51, 102

SLA Service Level Agreement. v–viii, 2, 5, 6, 31, 32, 35–37, 41, 42, 62, 77, 129, 131, 132

SNR Signal to Noise Ratio. xiii, 18, 25, 46, 47, 49, 59, 63, 108, 134

TP Transmission Power. 25, 38–40, 44, 46, 49, 59, 67, 77

TRL Technology Readiness Level. 2

UL up-link. xvi, 17

vLNS virtual LoRaWaN Network Server. 34, 35

VNF Virtualized Network Function. 31, 32

ZSM Zero-touch network and Service Management. 32

Annexe A

Résumé du manuscrit en français

Contenu

A.1 Introduction

De nos jours, de nouveaux paradigmes et technologies tels que la softwarisation et la virtualisation, le calcul de périphérie multi-accès (MEC) et l'Internet des objets (IoT) massif, la gestion automatisée et l'orchestration, permettent une évolution dans la conception des infrastructures de réseau de communication vers des architectures "au-delà de la 5G" et 6G [1, 2, 3]. Ces transformations envisagées dans les réseaux sont conçues pour répondre aux changements et aux demandes identifiés dans la plupart des cas d'utilisation sociaux et industriels tels que : la disponibilité de services à la demande, la connectivité et les capacités de calcul à long terme, les boucles intelligentes pour toutes les opérations de support et de maintenance, etc. Cela nécessite une évaluation minutieuse des technologies mentionnées et de leur intégration, étant donné l'impact profond qu'elles auront très probablement sur l'ensemble du système 6G. En fait, certaines d'entre elles sont encore à un niveau de maturité technologique relativement faible et auront des interactions complexes avec les autres.

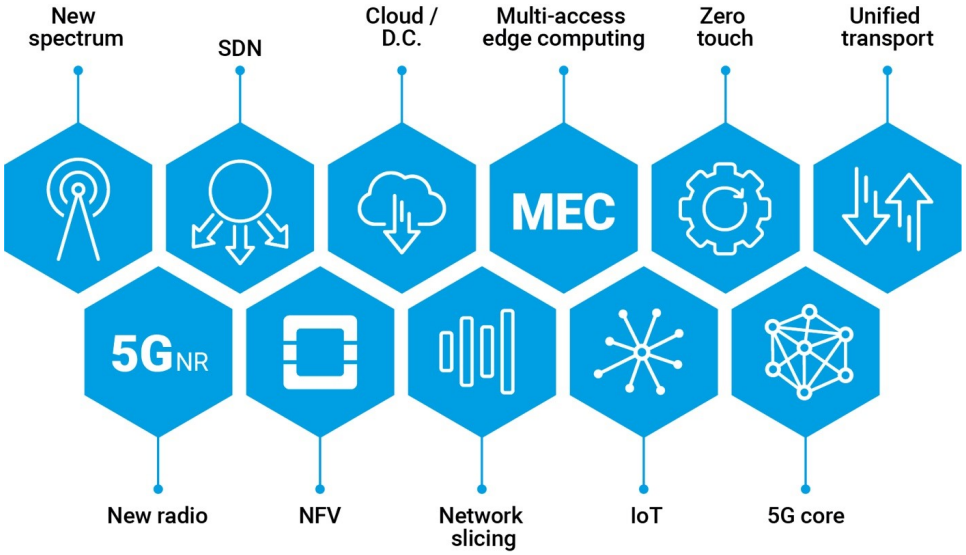


FIGURE A.1 : Éléments évolutifs dans l'architecture du réseau 5G [4, 5]. Acronymes et abréviations : réseau défini par logiciel (SDN), virtualisation des fonctions réseau (NFV), Cloud Distribué (D.C.), MEC, IoT.

Comme le montre la Figure A.1, parmi les principes et technologies de la 5G, on retrouve les paradigmes plus larges du découpage de réseau (*network slicing*) et de la gestion sans intervention (*zero-touch management*). D'une part, le network slicing postule la partition des réseaux en tranches horizontales plus ou moins isolées pour offrir à la demande des services spécifiques avec des accords

sur les niveaux de service (SLA) garantis [6]. D'autre part, le zero-touch management vise à atteindre une automatisation complète avec l'introduction de boucles intelligentes pour toutes les opérations de support et de maintenance [7]. Ensemble, ils sont conçus pour aider les opérateurs de réseau, les fournisseurs de services et les utilisateurs finaux à atteindre des niveaux améliorés d'efficacité, de flexibilité, d'automatisation, de sécurité et de qualité de service (QoS) globale.

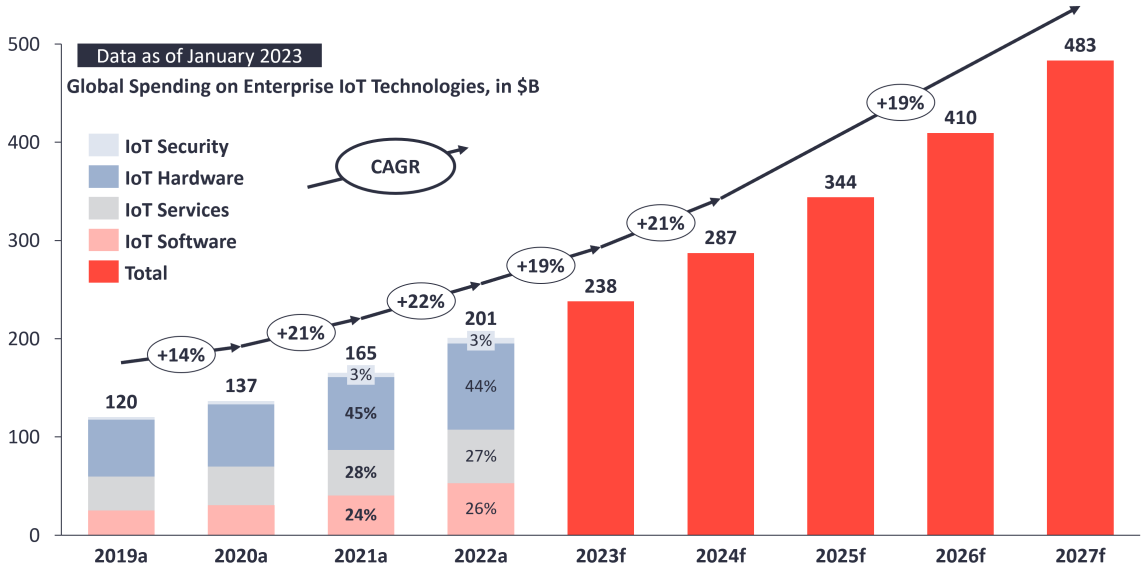


FIGURE A.2 : Marché de l'IoT d'entreprise de 2019 à 2027 [8]. Les données annuelles sont soit réelles (a), soit des prévisions (f). L'évolution du marché au fil des années est mesurée par le CAGR (*taux de croissance annuel composé*). Dans le cadre de l'étude de marché représentée, l'IoT est défini comme un réseau d'objets physiques connectés à Internet. Les objets qui deviennent connectés à Internet (appareils IoT) interagissent généralement via des systèmes embarqués, une forme de communication réseau ou une combinaison de calcul en périphérie (*edge computing*) et de l'informatique en cloud (*cloud computing*). Les données provenant des appareils connectés à l'IoT sont souvent utilisées pour créer de nouvelles applications pour les utilisateurs finaux. Les ordinateurs personnels, les tablettes et les smartphones connectés ne sont pas considérés comme faisant partie de l'IoT, bien qu'ils puissent faire partie de la configuration de la solution. Les appareils connectés via des méthodes de connectivité extrêmement simples, telles que l'identification par radiofréquence ou les codes QR, ne sont pas considérés comme des appareils IoT.

L'un des segments de réseau dont l'évolution a connu une croissance rapide en nombre est celui des appareils IoT, ce qui a conduit à la création du terme "Massive IoT" pour mettre en évidence l'impact qu'ils auront sur la peripherie du réseau. Comme le montre la Figure A.2, la part de marché de l'IoT d'entreprise a connu une croissance régulière ces dernières années, et plusieurs sources d'études de marché prévoient une tendance exponentielle dans les années à venir [9, 10, 11]. La présence émergente

de l'IoT se caractérise par une pléthore de technologies indépendantes au niveau du réseau Edge, dont certaines ont atteint le marché sans nécessairement respecter la vision globale des réseaux 5G et au-delà proposée par 3GPP, ETSI et d'autres organismes de normalisation. Ainsi, une évolution cohérente des réseaux de communication nécessite une intégration transparente de l'IoT dans la direction des transformations au-delà de la 5G.

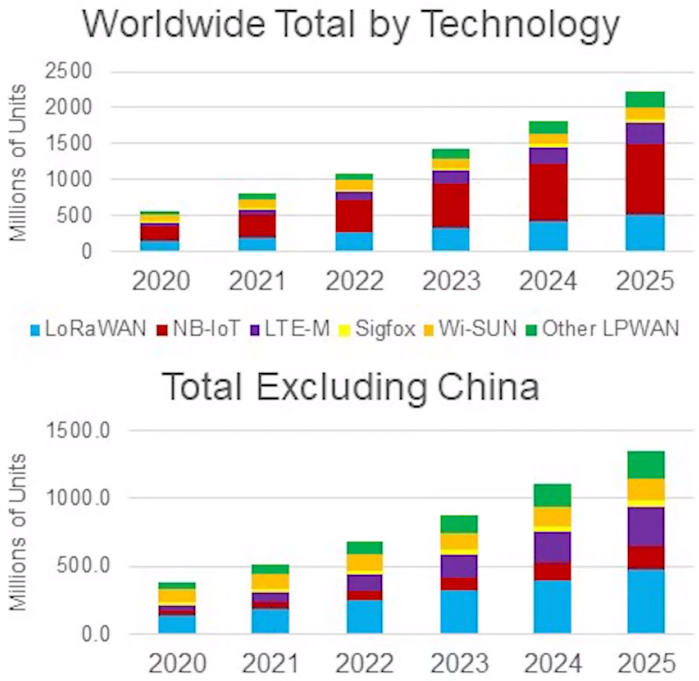


FIGURE A.3 : Évolution annuelle du nombre d'unités de base LPWAN installées par technologie [14]. Données réelles pour 2020, prévisions jusqu'en 2025. Une distorsion du marché IoT à bande étroite (NB-IoT) est présente en raison des initiatives du gouvernement chinois. En dehors de la Chine, LoRaWAN représente la plus grande part d'unités installées.

Parmi les technologies IoT, les réseaux étendus à longue portée (LoRaWAN) ont gagné en popularité grâce à leur caractère bon marché et facile à utiliser [12]. LoRaWAN est une technologie de réseau étendu à basse consommation d'énergie (LPWAN) *non-3GPP* qui vise à la détection et la surveillance à longue portée. Les cas d'utilisation se situent généralement dans le domaine de l'agriculture (par exemple, le suivi du bétail, la surveillance des propriétés du sol, etc.) et des villes intelligentes (par exemple, la gestion des vélos en libre-service, la surveillance des déchets, la surveillance de la qualité de l'air, etc.) [13]. Comme le montre la Figure A.3, LoRaWAN représente actuellement une part importante des déploiements LPWAN existants.

L'un des choix de conception dans LoRaWAN est l'accès non coordonné au support radio. Par exemple, en Europe, seule une limitation de cycle de service simple est présente, établissant le pourcentage moyen de temps que chaque appareil peut passer à occuper le canal. Cela vise à minimiser la consommation d'énergie des appareils intermittents en simplifiant la pile de protocoles de communication. En conséquence, les collisions deviennent plus fréquentes dans les déploiements denses et entraînent un fonctionnement au mieux et une utilisation intrinsèquement inefficace des ressources [15]. Pour cette raison, une métrique de performance clé pour une application utilisant LoRaWAN est le taux de livraison des paquets (PDR) [16], mesurant la fiabilité de la liaison en tant que nombre de paquets reçus divisé par le nombre de paquets envoyés sur une période de temps.

Étant donné la popularité actuelle de LoRaWAN et la croissance attendue de l'IoT dans les années à venir, il est pertinent d'étudier comment la technologie pourrait être intégrée dans l'évolution de l'écosystème des réseaux 5G et au-delà. En particulier, les cas d'utilisation des communications de type machine (MTC) 5G actuellement couverts par LoRaWAN ont souvent des exigences de qualité différentes, notamment en termes de fiabilité des paquets (pensez, par exemple, à une alarme incendie par rapport à un lave-vaisselle intelligent par rapport à un capteur d'humidité dans un contexte de ville intelligente [17, 18]). Alors que la différenciation de qualité en 5G est abordée par le biais du network slicing et de l'allocation automatisée des ressources, LoRaWAN gère toujours le trafic de manière au mieux, sans garantie en termes de fiabilité. À ce jour, l'intégration de LoRaWAN dans le cadre d'automatisation de gestion envisagé pour les technologies MTC 5G reste une question de recherche ouverte.

A.2 Énoncé du problème et défis

Le network slicing tel qu'envisagé dans la 5G nécessite la partition des ressources pour créer des flux de trafic de bout en bout (E2E) plus ou moins isolés. Un défi notable dans le network slicing est l'idée d'introduire une automatisation complète dans le processus d'allocation des ressources : les opérateurs concluent avec chaque client du slicing un SLA contenant les métriques de qualité de trafic souhaitées, de sorte que les composants d'orchestration du réseau puissent adapter automatiquement les ressources pour répondre aux exigences définies dans l'accord. Bien que la recherche sur le sujet ait progressé pour le réseau central 5G grâce à l'avènement de fonctions réseau virtuelles et programmables, il s'agit toujours d'un problème largement non résolu en ce qui concerne le réseau d'accès radio (RAN) [3].

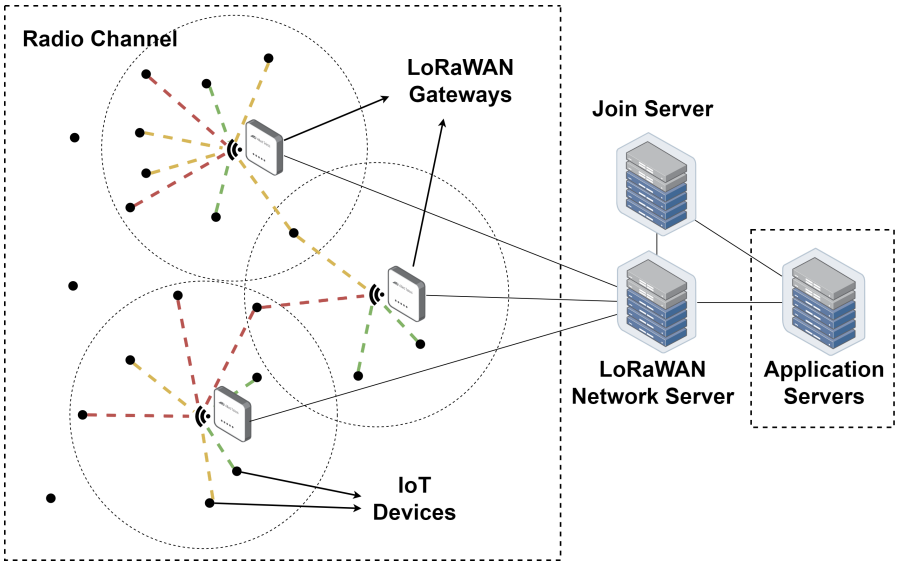


FIGURE A.4 : Aperçu de l'architecture LoRaWAN.

Dans cette thèse, nous examinons les problèmes non résolus concernant l'intégration de LoRaWAN (Figure A.4) dans un cadre de network slicing et de gestion automatisée. Une attention particulière est accordée au problème de l'isolation des groupes de dispositifs IoT et de l'application de différentes exigences de fiabilité grâce à des stratégies d'allocation des ressources. Les recherches existantes sur le slicing de LoRaWAN se concentrent sur le RAN. En effet, pour la plupart, les fonctions principales de LoRaWAN intègrent nativement la virtualisation, la programmabilité et l'observabilité requises par le network slicing. En revanche, plusieurs défis apparaissent lorsque le RAN est pris en compte : aucune spécification réaliste des exigences pouvant être intégrée dans les SLA, une action limitée dans la partition des ressources radio et des opportunités de contrôle des dispositifs dispersées.

Les travaux connexes sur la différenciation de la qualité du trafic RAN introduisent des exigences de QoS uniquement en tant que lignes directrices d'optimisation dans le processus d'allocation des ressources radio ; il manque encore un moyen d'imposer des exigences strictes en termes de qualité, et en particulier de fiabilité sous la forme de PDR, dans la littérature sur la différenciation du trafic. Étant donné le caractère contraignant des SLA dans le network slicing, ce point est un aspect clé et constitue un défi central pris en compte dans nos contributions.

LoRaWAN fonctionne sur des bandes sans licence, et les dispositifs partagent généralement toute la bande passante disponible en choisissant aléatoirement l'une des fréquences disponibles à chaque transmission. Le nombre limité de canaux discrets indépendants réduit la granularité de partitionnement

des ressources radio. Par conséquent, l'isolation des groupes de dispositifs n'est pas aussi simple que cela pourrait l'être dans d'autres technologies radio, en particulier lors de l'introduction de considérations d'équité, comme c'est couramment le cas dans le network slicing.

Enfin, dans les cas d'utilisation de LoRaWAN, les dispositifs émettent des paquets montants de manière périodique ou sporadique. Dans leur mode de fonctionnement le plus courant, les dispositifs n'admettent qu'une seule opportunité de configuration descendante par transmission montante. Par conséquent, l'introduction de boucles de rétroaction intelligentes pour l'automatisation du RAN, comme envisagé par le zero-touch management, nécessite un dimensionnement minutieux pour tenir compte des limitations techniques et physiques présentes dans les LoRaWAN.

A.3 Questions de recherche et contributions

À partir de l'analyse de l'état de l'art réalisée dans cette thèse et résumée dans la section précédente, les questions de recherche suivantes ont été formulées :

1. Est-il possible d'obtenir une différenciation du trafic LoRaWAN avec des exigences de qualité strictes ?

2. Est-il économiquement viable pour les opérateurs et les utilisateurs de LoRaWAN d'adopter des exigences de qualité strictes ?

3. Quelles approches adopter pour aller vers une gestion automatisée de la qualité dans LoRaWAN ?

La question 1 est abordée en proposant un premier schéma de gestion des ressources radio intégrant des éléments de contrôle d'accès. Nous adoptons un modèle mathématique issu de la littérature pour obtenir la quantité de trafic offert à admettre sur chaque facteur de propagation (SF, paramètres de modulation radio quasi-orthogonaux) de manière à garantir un certain PDR sur un seul canal de fréquence. Ensuite, nous utilisons cette quantité pour intégrer l'exigence de PDR dans le problème d'allocation des canaux entre les groupes de dispositifs. Nous proposons deux critères pour attribuer le nombre discret de canaux de manière proportionnelle aux demandes des groupes : le premier offre une isolation totale, tandis que le second ajoute la possibilité de mettre à niveau les dispositifs vers une classe de fiabilité supérieure pour mieux répartir les ressources. Enfin, les SF sont progressivement

attribués aux dispositifs classés par rapport signal sur bruit (SNR). Pour chaque SF, l'allocation s'arrête une fois que la capacité de trafic offerte provenant du PDR souhaité est remplie.

La technique est testée dans un scénario multi-gateway simulé avec des dispositifs assignés à trois classes d'exigences de PDR (97%, 90% et 70%) adaptées à la catégorisation utilisée pour les cas d'utilisation 5G MTC. Les résultats montrent que notre méthode surpasse les méthodes connues en termes de satisfaction du PDR grâce à une allocation améliorée des paramètres et atteint un niveau élevé d'équité intra-cluster, au détriment cependant de la diminution de la portée cellulaire maximale. Ces résultats sont présentés dans le Chapitre 4 et ont également été publiés dans l'article complet de conférence suivant :

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Packet delivery ratio guarantees for differentiated LoRaWAN services," in *Proc. 2022 IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 2014–2019, doi: 10.1109/GLOBECOM48099.2022.10001145.

Après avoir établi la faisabilité technique d'introduire des exigences strictes de PDR, nous passons à l'évaluation de la viabilité économique de cette classe d'approches. La question 2 est étudiée en testant un éventail plus large de stratégies d'allocation et en introduisant des métriques pour l'utilité de l'utilisateur et le gain de l'opérateur mesurés en termes d'efficacité des ressources. Nous considérons un scénario urbain plus réaliste et dense, et nous introduisons trois techniques d'allocation de canaux pour couvrir l'ensemble du spectre de l'équité (priorité aux exigences élevées, équité proportionnelle et maximisation du débit global). De plus, en plus du contrôle d'accès, nous présentons une première approche utilisant la primitive de réglage natif du cycle de service du protocole LoRaWAN.

Les résultats numériques montrent que dans des environnements à haute densité, nous pouvons atteindre un PDR supérieur de 20% avec l'une des politiques proposées, améliorant le taux de service moyen des dispositifs de 10% et le gain de l'opérateur de 7,5%. Dans les autres cas, cependant, l'utilisation des ressources du réseau peut être inférieure avec une différenciation de qualité basée sur un PDR strict. Cela peut être compensé par l'opérateur uniquement via une tarification ad hoc, ce qui pose des problèmes d'attractivité pour les clients potentiels. Les résultats indiquent que l'inefficacité peut être attribuée à deux facteurs principaux : (i) la quantité limitée de ressources matérielles dans les points d'accès LoRaWAN affectant l'isolation normalement fournie par différents canaux, et (ii) l'adoption d'un modèle mathématique conservateur pour produire des valeurs de capacité utilisées dans

l'allocation, entraînant souvent les techniques à dépasser le PDR requis. Ces résultats sont présentés dans le Chapitre 5 et ont également été publiés dans l'article complet de conférence suivant :

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Traffic control and channel assignment for quality differentiation in dense urban LoRaWANs," in *Proc. 20th Int. Symp. Model. and Optim. in Mobile, Ad hoc, and Wireless Netw. (WiOpt)*, 2022, pp. 153–160, doi: 10.23919/WiOpt56218.2022.9930551.

Notre troisième contribution vise à résoudre les inefficacités identifiées tout au long de l'étude de la question 2, tout en proposant une première approche à l'automatisation de la gestion de la qualité mentionnée dans la question 3. Une boucle de rétroaction est introduite pour configurer dynamiquement le cycle de service des dispositifs en fonction du PDR mesuré. Cela permet à l'algorithme d'apprendre la capacité du canal au lieu d'utiliser un modèle mathématique statique. Cette approche utilise une estimation en ligne basée sur la méthode de la bissection pour minimiser le nombre de reconfigurations nécessaires pour la convergence. De plus, elle améliore le contrôle du cycle de service précédemment proposé, en maintenant la compatibilité de la technique avec les primitives LoRaWAN existantes.

Les résultats montrent une amélioration drastique de l'efficacité des ressources (toujours supérieure à la PDR cible, dépassement maximal de 2%) par rapport à notre approche précédente basée sur un modèle. Cela se fait au prix de devoir entraîner l'algorithme une fois, un processus qui peut prendre jusqu'à 80 heures. Cependant, une fois convergée, l'approche est capable de s'auto-rétablir face aux changements dans le groupe de dispositifs connectés et peut être associée à un algorithme d'allocation des canaux pour établir une différenciation de qualité.

Enfin, pour capitaliser sur les considérables efforts de validation, d'implémentation et d'optimisation du code qui ont été déployés pour simuler LoRaWAN, *ELoRa* est proposé, un outil open source développé pour émuler et étudier les infrastructures massives de l'IoT E2E. Dans *ELoRa*, nous construisons une couche de traduction entre un modèle de simulation bien connu ns-3 et le trafic LoRaWAN réel (encapsulé en UDP), permettant ainsi les communications bidirectionnelles en temps réel avec l'environnement extérieur. La couche MAC et l'implémentation du transfert de paquets de passerelle dans le modèle de simulation LoRaWAN d'origine sont considérablement étendues pour être acceptées de manière transparente par la plupart des serveurs LoRaWAN. Nous introduisons également un outil pour émuler le trafic des villes intelligentes, comme indiqué dans le tableau 3.1,

nous mettons en œuvre la génération de captures de paquets LoRaWAN au format `.pcap` en plus des journaux et des métriques pour atteindre une observabilité complète du RAN, et nous empaquetons l'outil dans un conteneur pour faciliter son déploiement pour des tests automatisés multi-scénarios.

Grâce à la variété d'outils de modélisation offerts par ns-3, *ELoRa* permet de tester des techniques de gestion sur les serveurs LoRaWAN dans d'innombrables scénarios et charges réalistes. De plus, les paramètres des dispositifs simulés peuvent être contrôlés en direct par le serveur pour tester des algorithmes de gestion des ressources radio existants et novateurs dans un environnement contrôlé E2E complet. Grâce à l'architecture cloud-native de la pile de serveurs LoRaWAN, *ELoRa* peut être déployé localement sur le serveur ou de manière distribuée, autonome, dans plusieurs instances, ou accompagner d'autres flux de trafic provenant de réseaux réels. *ELoRa* est présenté dans le Chapitre 7 et a été publié dans l'article de démonstration de la conférence suivante :

> A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "ELoRa : End-to-end emulation of massive IoT LoRaWAN infrastructures," in *Proc. 2023 IEEE/IFIP Netw. Operations and Manage. Symp. (NOMS)*, 2023, pp. 1–3, doi: 10.1109/NOMS56928.2023.10154373.

A.4 Perspectives futures

Une perspective à court terme pour le travail présenté dans cette thèse serait l'introduction de nouveaux types d'exigences en matière de qualité. Comme discuté dans le Chapitre 5, il y a un compromis direct entre le débit global et la fiabilité des paquets, où LoRaWAN dans son fonctionnement par défaut maximise toujours le débit global et minimise la consommation d'énergie. Cependant, un compromis tripartite PDR/débit/consommation d'énergie pourrait être modélisé en considérant des primitives de configuration supplémentaires offertes par le protocole. Par exemple, les retransmissions pourraient être considérées pour augmenter à la fois le PDR et le débit au détriment de la consommation d'énergie [68]. Si des transmissions confirmées sont introduites pour la détection de collision, la latence unidirectionnelle pourrait également devenir une mesure intéressante basée sur le nombre d'essais nécessaires pour livrer correctement un message. De plus, la définition des métriques pourrait être étendue à l'ensemble de l'infrastructure, comme dans le network slicing, par exemple en multipliant les coefficients de fiabilité des nœuds le long de la chaîne de RAN et des fonctions du réseau central.

D'autres perspectives à court terme concernant nos techniques proposées sont les suivantes.

L'exécution en ligne des politiques d'allocation de canaux pourrait être évaluée, puis améliorée pour s'adapter aux changements, et intégrer toujours l'isolation souple (présentée au Chapitre 4) dans le processus d'optimisation pour une partition plus fine des ressources (canaux). Les algorithmes de contrôle de collision du Chapitre 6 pourraient être étendus avec la capacité de revenir en arrière dans les étapes de la procédure de bisection s'ils sont déployés sur un nouveau scénario et nécessitent une certaine quantité de ré-entraînement. D'autres perspectives pourraient inclure l'intégration des appareils mobiles dans le cadre de différenciation de la qualité du trafic, en utilisant éventuellement des techniques prédictives et une allocation préemptive [41]. À l'avenir, nous prévoyons également d'utiliser *ELoRa* pour reproduire d'autres scénarios massifs de l'IoT, y compris des techniques de détection d'anomalies, afin de tester le placement des fonctions du réseau LoRaWAN et les solutions d'automatisation.

Dans cette thèse, nous avons considéré un cadre d'allocation de ressources impliquant plusieurs sous-problèmes modulaires : l'allocation de canaux aux groupes d'appareils, l'optimisation des paramètres de modulation (c'est-à-dire l'ADR, les agorithmes de débit de données adaptatifs), et le contrôle du trafic pour atteindre des objectifs spécifiques de PDR par groupe. Cette distinction entre différents problèmes permet d'éclairer la dynamique du système, de clarifier les limitations et de réduire la complexité de la modélisation mathématique impliquée dans l'allocation des ressources. L'inconvénient de cette approche est qu'elle peut réduire l'espace des solutions possibles qui seraient considérées dans une technique d'allocation de ressources plus globale (mais monolithique). Par exemple, des approches entrelacées sont présentées au Chapitre 4 (une combinaison de ADR et de contrôle du trafic), au Chapitre 5 et dans [62] (une combinaison d'allocation de canaux et de contrôle du trafic), et elles aboutissent à une "partitionabilité" améliorée des ressources radio. Ainsi, une perspective future à moyen terme pourrait impliquer la conception d'une technique globale d'allocation de paramètres qui fait actuellement défaut.

Dans cette optique, l'apprentissage par renforcement profond (DRL) s'est avéré être un bon outil pour gérer automatiquement la complexité mathématique qui pourrait autrement être impliquée. Cependant, l'adoption du DRL dans ce contexte reste une entreprise très ouverte : la grande dimensionnalité de l'espace d'action, combinée aux opportunités de configuration limitées de LoRaWAN et à l'observabilité extrêmement partielle de l'environnement radio (très différente des systèmes clos avec une faible latence entre les actions où le DRL a montré des résultats pertinents [88]), pourrait

entraîner un temps de convergence très long pendant l'entraînement. Une solution possible pourrait consister en un entraînement partiel du modèle dans un environnement simulé axé sur des capacités de généralisation élevées, avec une grande variété de scénarios différents et une randomisation de tous les paramètres. Cela suppose bien sûr des améliorations supplémentaires des modèles de simulation actuels provenant de la communauté de recherche en radio pour parvenir à une réplication quasi-parfaite des effets d'interférence et de bruit. Ensuite, le modèle pourrait être déployé dans un environnement réel pour une deuxième étape d'entraînement visant à apprendre les spécificités de ce système. Cependant, le manque de reproductibilité des scénarios dans cette deuxième étape pourrait entraver le processus d'apprentissage. Les avancées actuelles en matière d'apprentissage automatique pourraient contribuer à résoudre ces problèmes à l'avenir.

Des perspectives importantes à long terme impliquent l'évaluation continue des capacités de LoRaWAN. Dans cette thèse, nous supposons que les passerelles sont limitées par le cycle de service, comme dans le reste de la littérature, mais comme discuté dans la section 2.5, cela peut ne pas être complètement vrai, car selon la loi, elles pourraient utiliser LBT + AFA (*écoute avant de parler + agilité de fréquence adaptative*, activé par défaut dans les implémentations existantes) et les spécifications ne sont pas claires à ce sujet. Ainsi, la faible viabilité du trafic acquitté pourrait encore faire l'objet de débats et des avancées pourraient être réalisées en matière de techniques de résolution de collision. Par exemple, l'adoption de dispositifs de classe B pour permettre des formes de découpage temporel et de configuration multi-diffusion. Au détriment de la consommation d'énergie, les dispositifs de classe B pourraient largement améliorer les temps de reconfiguration dans notre proposition de contrôle de collision du Chapitre 6. Malheureusement, à notre connaissance, aucun outil d'évaluation n'existe actuellement qui intègre les fonctionnalités mentionnées des dispositifs de classe B. À l'avenir, il pourrait être intéressant de considérer l'adaptation des algorithmes proposés dans cette thèse aux réseaux LoRaWAN avec des dispositifs de classe B, ou un mélange hétérogène de classe A et B, et d'intégrer des dispositifs de classe B dans *ELoRa* afin d'étudier les performances qui pourraient être atteintes par rapport aux dispositifs de classe A.