Università degli Studi di Padova

Università degli Studi di Padova

Department of Information Engineering

Ph.D. in Information Engineering

Curriculum "Science and Information Technology"

XXXIV cycle

Mathematical and Experimental Analysis of Low Power Wide Area Network Technologies in Advanced IoT Scenarios

Coordinator
Prof. A. Neviani
Università di Padova

Supervisor
Prof. A. Zanella
Università di Padova

Ph.D. Candidate
Martina Capuzzo

Academic Year 2021/2022

To those
who believe in me.

Abstract

The Internet of Things (IoT) paradigm is nowadays applied to multiple domains, including Smart Cities, Smart Industry and Smart Agriculture. To support the specific requirements of these scenarios, Low Power Wide Area Network (LPWAN) technologies have been developed, among which Long Range Wide Area Network (LoRaWAN) and Narrowband IoT (NB-IoT) play a dominant role.

This thesis aims at evaluating the performance of these technologies by considering both traditional IoT scenarios and more challenging use cases, such as industrial monitoring or remote drone tracking, which have strict communication requirements in terms of reliability and delay. To estimate the network performance in all these domains, mathematical modeling and network simulations have been used, leveraging the ns-3 `lorawan` module. From these evaluations, it is appearent that a proper technology configuration has a significant impact on the system's performance, and that multiple elements should be taken into account when implementing an IoT system.

Another aspect considered in this thesis regards the energy efficiency: indeed, although LPWAN technologies are designed to be low power, the evaluation on real testbeds can help in assessing the correctness of the node's behavior and the impact of the network settings on the device lifetime. Furthermore, most of the IoT devices are currently battery-powered, an approach that is not economically sustainalble, nor environmental friendly. A possible alternative is to implement *Green IoT* systems by providing IoT nodes with a mechanism that allows them to harvest power from renewable sources. The thesis applies this concept to LoRaWAN devices, and discusses its feasibility by leveraging ns-3 simulations and experiments on real testbeds. Finally, in the last chapter of the thesis, we present experimental results considering the energy consumption of a NB-IoT device connected to different operators' networks, evaluating several configurations.

Sommario

Attualmente il concetto di *Internet of Things (IoT)* è applicato a diversi ambiti, che spaziano da applicazioni di Città Intelligenti (Smart City) a quelle in ambito industriale e agricolo (Smart Industry e Smart Agriculture). Per rispondere agli specifici requisiti di questi scenari sono state progettate le tecnologie Low Power Wide Area Network (LPWAN), tra le quali Long Range Wide Area Network (LoRaWAN) e Narrowband IoT (NB-IoT) hanno un ruolo predominante.

Obiettivo di questa tesi è valutare le prestazioni di queste tecnologie, considerando sia scenari IoT "tradizionali", sia casi d'uso più impegnativi, come il monitoraggio di sistemi industriali e la localizzazione di droni da remoto, dove i requisiti di comunicazione in termini di affidabilità e latenza sono più stringenti. Per stimare le prestazioni di rete in questi ambiti sono stati impiegati modelli matematici e simulazioni di rete che utilizzano il modulo `lorawan` di ns-3. Da queste valutazioni emerge che un'appropriata configurazione della tecnologia di comunicazione ha un impatto significativo sulle prestazioni del sistema, e che vari fattori devono essere considerati quando si implementa un sistema IoT.

Un altro aspetto considerato in questa tesi è quello del consumo energetico: infatti, nonostante le tecnologie LPWAN siano progettate per avere basso consumo, la loro valutazione in sistemi reali può contribuire a verificare il corretto comportamento del nodo e l'impatto dei settaggi di rete sul ciclo di vita del dispositivo. Inoltre, molti dispositivi IoT sono attualmente alimentati a batterie, un approccio poco sostenibile economicamente e con grande impatto ecologico. Pertanto, una possibile alternativa è implementare sistemi di *Green IoT*, equipaggiando i nodi IoT con meccanismi che permettono di assorbire energia da sorgenti rinnovabili. La tesi applica questo concetto a dispositivi LoRaWAN e ne discute la fattibilità utilizzando simulazioni ns-3 e esperimenti con dispositivi reali. Infine, nell'ultimo capitolo della tesi, sono presentati

risultati sperimentali che considerano il consumo energetico di un dispositivo NB-IoT connesso a reti di diversi operatori, e ne valutano varie configurazioni.

Contents

List of Figures

List of Tables

Acronyms

Symbols

3D-CTRA 3-Dimensional CTRA

3GPP Third Generation Partnership Project

A

ABP Activation By Personalization

ACK Acknowledgment

ADR Adaptive Data Rate

AFA Adaptive Frequency Agility

AoI Age of Information

B

BER Bit Error Rate

BF Bayesian Filtering

BLE Bluetooth Low Energy

BS Base Station

C

CD Confirmed Downlink PDR

CDF Cumulative Distribution Function

cDRX Connected state Discontinuous Reception

CFO Carrier Frequency Offset

CoAP Constrained Application Protocol

CPSR Confirmed Packet Success Rate

CR Code Rate

CRC Cyclic Redundancy Check

CS Constant Speed

CSS Chirp Spread Spectrum

CTRA Constant Turn Rate and Acceleration

CU Confirmed Uplink PDR

D

DC Duty Cycle

DCI Downlink Control Indicator

DL Downlink

DR Data Rate

DRX Discontinuous Reception

DSSS Direct-Sequence Spread Spectrum

E

EC-GSM Extended Coverage GSM

ECL Extended Coverage Level

ED End Device

EDGE Enhanced Data rates for GSM Evolution

eDRX Extended Discontinuous Reception

eMTC Enhanced Machine-Type Communication

ETSI European Telecommunications Standard Institute

F

FDGW Full Duplex Gateway

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FFT Fast Fourier Transform

G

GPRS General Packet Radio Service

GPS Global Positioning System

GSM Global System fro Mobile communications

GW Gateway

I

iDRX Idle state DRX

IIoT Industrial Internet of Things

IoT Internet of Things

ISM Industrial, Scientific, and Medical

ITU International Telecommunication Union

K

KF Kalman Filter

KPIs Key Performance Indicators

L

LBT Listen Before Talk

LoRa Long Range

LoRaWAN Long Range Wide Area Network

LOS line-of-sight

LPWAN Low Power Wide Area Network

LR-WPAN Low-Rate Wireless Personal Area Network

LTE Long Term Evolution

LTE-M LTE-Machine Type Communication

M

M2M Machine-to-Machine

MAC Medium Access Control

MCL Maximum Coupling Loss

MCU Micro Controller Unit

MFSK Multiple Frequency Shift Keying

MHDR MAC Header

MIC Message Integrity Code

mMTC Massive Machine Type Communication

MQTT Message Queue Telemetry Transport

MTC Machine Type Communication

MType Message Type

N

NB-IoT Narrowband IoT

NLOS non line-of-sight

NRSRP Narrowband Reference Signal Received Power

NS Network Server

O

OOB Out-Of-Band

OTAA Over-The-Air Activation

P

PAN Personal Area Network

PAoI Peak Age of Information

PCI Physical Cell Identity

PDR Packet Delivery Rate

PF Particle Filter

PHY Physical

PSM Power Saving Mode

PTW Paging Time Window

Q

QoS Quality of Service

R

RACH Random Access Procedure

RAI Release Assistance Indicator

RF radio-frequency

RPMA Random Phase Multiple Access

RRC Radio Resource Control

RSRP Reference Signal Received Power

RSRQ Reference Signal Received Quality

RSSI Received Signal Strength Indicator

RTT Round Trip Time

RX1 first receive window

RX2 second receive window

S

SB1 Sub-Band 1

SB2 Sub-Band 2

SF Spreading Factor

SINR Signal to Interference plus Noise Ratio

SIR Signal to Interference Ratio

SNR Signal to Noise Ratio

T

TAU Tracking Area Update

TDD Time Division Duplexing

TDMA Time Division Multiple Access

TDMA Time Division Multiple Access

ToA time on air

TTN The Things Network

TX Power Transmission Power

U

UAV Unmanned Aerial Vehicle

UE User Equipment

UKF Unscented Kalman Filter

UL Uplink

UL-PDR Uplink Packet Delivery Ratio

UNB Ultra Narrow Band

UT Unscented Transformation

UU Unconfirmed Uplink PDR

W

WPAN Wireless Personal Area Network

1 Introduction

The Internet of Things (IoT) is a new paradigm according to which every object can potentially be connected to the Internet. This disruptive concept enables the remote monitor and control of a wide set of heterogeneous objects through an Internet connection, a capability that enables various services in a wide array of scenarios and that can impact our lifestyle and everyday activities.

For example, cities could benefit from smart lighting control, a more efficient waste management, and continuous infrastructure monitoring [1, 2], which are examples of the so-called *Smart City* applications, that can improve citizens security, offer better public services and reduce the city's costs. In industrial scenarios, connected sensors can be adopted to implement the *Industrial Internet of Things (IIoT)*, helping in the continuous monitoring of the production process, and making it possible to quickly detect or even predict failures. In the agricultural sector the widespread collection of environmental data, such as temperature and soil moisture, can improve quantity and quality of the soil production, while reducing costs (*Smart agriculture* applications) [3]. Health monitoring, home security and home automation are yet other examples of possible application scenarios [4, 5]. Therefore, what in the last years was just a *vision* is currently being deployed into different domains, thanks to the strong push from the industrial community and city administrators. Indeed, minimizing maintenance costs and waste through a sharper monitoring processes' efficiency and resource consumption makes it possible to increase both the production and the quality of the services offered to the final users.

In general, the communication needs of IoT scenarios differ significantly from the classic high-throughput and low-delay requirements that have so-far driven the design of traditional communication systems. For example, in many IoT scenarios, supporting sporadic transmissions of short packets from a massive number of devices is more important than providing stable high-throughput connections to a few users [6]. Furthermore, long-range communication technologies are required to minimize the infrastructure and provide connectivity in remote areas and with low power consumption, so that the IoT nodes (usually sensors) can work for a longer time. To address these requirements, new communication technologies have been recently proposed under the name of Low Power Wide Area Network (LPWAN). Depending on the specific application considered, one technology may fit better than the others, and the variety of the proposed solutions can therefore be suitable to fulfill the different use cases. For example, cellular IoT technologies can benefit from the already existing LTE networks and licensed frequency bands provided by telecommunication operators, are more reliable in terms of latency, but are associated with higher subscription costs. On the other hand, other Low Power Wide Area Network (LPWAN) technologies employ unlicensed frequency bands and allow the deployment of private networks, which can be under the full control of the user (with all the advantages and drawbacks). Also, hybrid and *ad hoc* solutions can be proposed, according to the needs of the market.

Among the proposed LPWAN solutions for the IoT applications, Long Range Wide Area Network (LoRaWAN) and Narrowband IoT (NB-IoT) play a major role. One of the main advantages of LoRaWAN is the flexibility of deployment and configuration, also motivated by the use of the unlicensed spectrum and the open standard defining the Medium Access Control (MAC) layer protocol. LoRaWAN leverages the patented Long Range (LoRa) modulation, whose performance can be adapted to the considered scenario, providing a reliable communication even in harsh environments and a coverage range up to few kilometers. This, together with its low power consumption and high network scalability, make it one of the best candidate for applications such as Smart Cities and Smart agriculture. On the other side, NB-IoT can be easily provided by telecommunication operators that, having complete control of the network, can offer more guaranteed communication performance, as well as support for a proper configuration of the network, to fit the users' needs. Therefore, this technology can be a more reliable candidate for IIoT applications, where the stronger channel impairments can be mitigated by a more dense infrastructure.

In this thesis, we focus on the evaluation of some LPWAN technologies considering several aspects. We evaluate the performance of LoRaWAN networks in terms of scalability, communication reliability, and delay performance in different scenarios, spanning from the most traditional IoT applications, to more "challenging" use cases, where the application needs to transmit as frequently as possible, and a reliable communication with short delays is required. We also analyze a Green IoT solution, where, instead of using batteries, the device harvests the required energy from renewable external sources.

Assessing the technology capabilities in such scenarios can help answer questions of uttermost importance when considering IoT networks, such as

(i) How should the network be configured to achieve given performance in terms of scalability, data successful delivery, minimum delay?

(ii) What is the trade-off between communication reliability and network scalability? Is it possible to improve it?

(iii) Can we leverage LoRaWAN extended coverage and low-power communication to support non-traditional IoT applications?

(iv) What is the impact of external regulations about the use of the unlicensed frequency spectrum on network performance?

(v) Is it feasible to apply Green IoT to LoRaWAN?

To answer these questions, we studied LoRaWAN technology with different approaches. First, we developed a mathematical model that captures the main features of the technology and its specificities. Second, we used the popular ns-3 network simulator to investigate the impact of several possible system settings on different applications. The advantage of the simulator is that it offers a rather complete and accurate representation of the full protocol stack, making it possible also to evaluate the effect of possible changes in the protocol design. However, the high accuracy comes at the price of long and computational costly simulations compared to the mathematical modeling. At the same time, the simulator makes it possible to study the behavior and performance of a complete network before the actual installation, easing the configuration of all the nodes. A third approach we followed was experimental, on real testbeds, to assess whether the empirical behavior was coherent with what was theorized.

Finally, we consider the energy performance of NB-IoT nodes, deeply evaluating the impact of each operational phase of the device, and the effect of timers values and communication settings. Since NB-IoT employs the infrastructure provided by private telecommunication operators, the node does not have full control of the communication and network settings, and the choice of the operator can actually impact on the device performance. Therefore, we also evaluated the impact of the operator, as well as the device choice, on the energy consumption. To this aim, we performed an extensive measurement campaign, collecting data in different scenarios and with multiple configurations of the technology.

This thesis addresses the aforementioned topics providing an in-depth analysis of the performance of LoRaWAN, under different perspectives, and an evaluation of NB-IoT with respect to the energy consumption. The presented results show that LoRaWAN can be an efficient technology both in traditional IoT scenarios, ensuring communication reliability and network scalability, and in more advanced use cases, supporting, for example, tracking applications and industrial monitoring. However, these results can be obtained only with a careful tuning of the system's setting and technology parameters. Furthermore, the considerations on the energy consumption demonstrate that devices can be prone to high consumption if the network and/or the technology are not carefully configured, and that more ecological solutions adopting harvesting techniques can be viable but require further investigations.

The rest of this thesis is structured as follows. Chapter 2 presents the main technologies that are in the field for IoT applications, describing the differences between the various categories and then focusing on the functioning and protocols of NB-IoT and LoRaWAN . Chapter 3 introduces the `lorawan` module for the ns-3 simulator. The module has been extensively employed to obtain most of the results presented in this dissertation. Chapters 4 and 5 investigate the LoRaWAN performance in traditional applications and more challenging scenarios, respectively. The energy consumption is the focus of the last part of the thesis. Chapter 6, discusses whether the Green IoT approach with battery-less devices can be applied to LoRaWAN, assessing the performance with simulation results and experimental evaluations. Then, Chapter 7 focuses on the current consumption performance of a NB-IoT node, showcasing that proper evaluation and careful settings need to be done before deploying the devices in a plug-and-play fashion. Finally, Chapter 8 summarizes the achieved results and discusses possible extensions.

LPWAN technologies 2

This section gives an overview of the most important technologies that are competing for the IoT market. These technologies are classified according to their coverage range, that determines their suitability for certain applications. In Sec. 2.1, the chapter first provides a general overview of three groups of IoT communication technologies, namely Low-Rate Wireless Personal Area Networks, cellular technologies for the IoT, and Low Power Wide Area Networks. Then, the chapter focuses on two technologies whose performance are discussed in the rest of the thesis, i.e., NB-IoT (Sec. 2.2), and the LoRa modulation and the LoRaWAN standard (Sec. 2.3, and Sec. 2.4).

2.1 Technologies' overview

IoT technologies are often classified in three categories based on their characteristics in terms of coverage range, throughput, cost and power consumption. Fig. 2.1, for example, depicts their different capabilities in terms of coverage range and power consumption. In the following, we give an overview of the main solution proposed in the market for the three categories.

2.1.1 Low-Rate Wireless Personal Area Network

Low-Rate Wireless Personal Area Network (LR-WPAN) technologies (indicated in Fig. 2.1 under the name of Short-range Wireless) includes technologies characterized by low bit rate, low power consumption and short coverage range, reaching at maximum a few hundred meters. This range can be extended using a dense deployments of gateways and devices connected in a multihop mesh network. Such a deployment becomes economically unfeasible and technically complex in very large scenarios such as cities or wide open-air areas. The main solutions belonging to this category are the following.

BLE Bluetooth Low Energy (BLE) was defined in 2010 by the Bluetooth Special Interest Group as a single-hop solution suitable for use cases as healthcare, consumer electronics and short-range tracking of personal devices. It operates in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band, defining 40 channels with different functions. The physical data rate is 1 Mbps and the coverage range is typically of few tens of meters. Similarly to Bluetooth, BLE specification defines two device

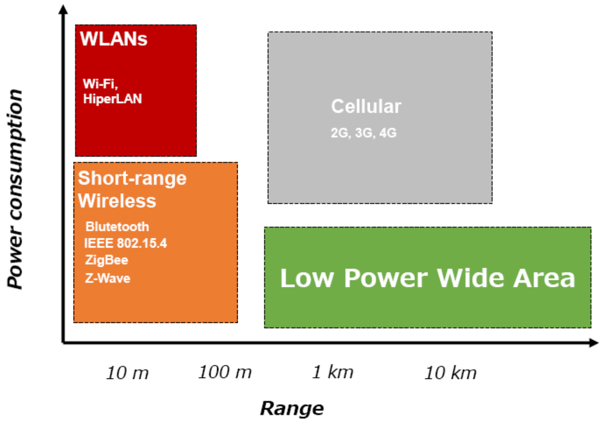


Figure 2.1.: Low Power Wide Area Network (LPWAN) vs. legacy wireless technologies [7].

roles: master and slave. To save energy, slaves can turn their radios off for some time intervals. However, the star topology does not allow for path diversity, which is important to overcome radio propagation impairments and node failures. For this reason, the implementation of a mesh network topology has been suggested by standard development organizations, academic community and industries [8].

IEEE 802.15.4 and ZigBee In 1999, IEEE established the 802.15 working group to develop specific standards for Wireless Personal Area Networks (WPANs). In particular, the target group four (802.15.4) was given the responsibility of developing standards for Physical (PHY) and Medium Access Control (MAC) layers enabling the transmission of small data flows consuming a little energy. By using the standard developed by 802.15.4, ZigBee proposes a complete standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), adding the specifications for network and application layers [9]. This technology works in the ISM 2.4 GHz band and defines 16 channels, which can be selected by the operating device. Three type of devices are defined. The Personal Area Networks (PANs) coordinator and the coordinator are defined as Full Function Devices as they implement all the functionalities of the IEEE 802.15.4 protocol. They transmit beacon frames to provide global and local synchronization, respectively. Simple nodes are connected to the coordinators and have no coordinating functionalities. ZigBee proposes three types of network topologies: star, tree and mesh. While with star and tree topologies a single link failure compromises the communication, the mesh topology is more robust, since it provides multiple paths from simple nodes to the coordinator. The specifications require a maximum transmission range of a hundred meters, a throughput between 20 and 250 kbps and a capacity of 64000 nodes for each PAN coordinator. The main applications of Zigbee are found in home automation and healthcare [10].

Z-Wave Z-Wave is a proprietary protocol designed by Zensys mainly for home automation applications. Only partial information can be found in literature, since it is a closed standard owned by the Z-Wave Alliance, an organization that groups all the Z-Wave vendors. It is aimed to support the exchange of short messages in a reliable manner from a central control unit to one or more nodes in the network using half duplex communication. It works in the ISM band around 900 MHz to limit power consumption and interference with other technologies. Its coverage range goes from 30 meters indoor to about 100 meters in outdoor environments. Two basic kinds of nodes are specified: controlling devices and slave nodes that receive and execute controlling commands, and, potentially, answer to them. If required, slave nodes can also forward commands to other nodes, extending the communication range of the controller. Controllers also maintain routing tables and one of them (the primary controller) can include/exclude nodes from the network [11].

2.1.2 Cellular IoT

Cellular technologies for IoT leverage the existing cellular network infrastructure to offer new services. The network infrastructure is, for most, already installed and, usually, only a software upgrade is required. One of their main characteristic is that, differently from the other IoT technologies, they operate in the licensed spectrum. In the following, we give an overview of the three solutions proposed by 3GPP in Release 13 [12]. Notice that connectivity for IoT and Massive Machine Type Communication (mMTC) is also among the use cases of 5G networks.

EC-GSM Extended Coverage GSM (EC-GSM) has been proposed to improve already existing GSM/EDGE networks, providing services to areas where Long Term Evolution (LTE) is not available. Using repetitions and signal combining techniques, a coverage of 164 dB Maximum Coupling Loss (MCL) is reached when using the maximum transmission power (33 dBm). EC-GSM adds new control and data channels to legacy Global System fro Mobile communications (GSM) and has the possibility of multiplexing new EC-GSM devices with other cellular technologies, such as legacy Enhanced Data rates for GSM Evolution (EDGE) and General Packet Radio Service (GPRS) [13]. Its deployment only requires a software upgrade on existing GSM networks and provides a combined capacity of 50000 devices per cell. eDRX is used to improve power efficiency and battery life. The achievable rate can vary from 350 bps to 240 kbps depending on the coverage and modulation used [12].

LTE-M LTE-Machine Type Communication (LTE-M), also called Enhanced Machine-Type Communication (eMTC) aims to enhance LTE technology for machine-type communications. By reducing bandwidth and maximum transmit power, it enables new power-saving functionality, lower device complexity and cost. The coverage is extended up to 155.7 dB MCL and a low deployment cost for network operator is possible: control and data are multiplexed in the frequency domain and therefore it is possible to schedule IoT devices within any legacy LTE system, sharing the carrier capacity, antenna, radio, hardware and frequency spectrum. The achieved rate varies between 10 kbps to 1 Mbps, both for Uplink (UL) and DL communication [12].

NB-IoT Similarly to the previous technology, Narrowband IoT (NB-IoT) reuses the already existing LTE infrastructure. The main difference is the narrow bandwidth used: 200 kHz [13]. To adapt it to IoT use cases, LTE has been simplified, reducing data rate, bandwidth, mobiliy support and optimizing the protocol. With these adaptations, the number of devices supported by each cell is about 50000 with an extended coverage of 164 dB MCL. NB-IoT supports three modes of operation according to the position of its carrier with respect to the LTE spectrum: standalone, guard band and in-band. Power saving mode described in 3GPP Release 12 and eDRX are used to increase battery life. The bit rate is about 250 kbps both in UL and DL. Also in this case, the low costs are achieved by lowering the complexity of the devices and by the fact that a software upgrade is sufficient to deploy it in LTE networks. Further details about this technology and its power saving mechanisms are provided in Sec. 2.2.

2.1.3 Low Power Wide Area Networks (LPWANs)

The Low Power Wide Area Network (LPWAN)s are halfway between LR-WPANs and cellular networks. In fact, they have a larger coverage than LR-WPANs, but lower cost and less energy consumption than cellular technologies. For these reasons they are spreading in the market and have already been adopted in different parts of the world. They are characterized by a star topology where peripheral nodes are directly connected to a concentrator, that acts as gateway towards the IP network. Robust modulation allows for the use of these technologies also in challenging environments, where cellular communication may fail [14].

Most of the LPWAN technologies use sub-GHz ISM bands, that are unlicensed and less crowded then 2.4 GHz band used by Wi-Fi and Bluetooth. We present now an overview of three LPWAN solutions that compete with the LoRaWAN solution, which will be thoroughly described in the Sec. 2.3, and Sec. 2.4.

Sigfox Sigfox has been the first LPWAN technology proposed in the IoT market by the homonymous French start-up founded in 2009. The architecture consists of transmitting devices, gateways and Sigfox back-end. The coverage is deployed in each country by a Sigfox Network Operator that owns the whole infrastructure and generally requires a yearly fee for each device to provide access to the network. By using Ultra Narrow Band (UNB) communications, Sigfox can transmit a signal occupying a channel of only 100 Hz, thus benefiting from flat fading and very low noise contribution. Therefore, it is possible to have a simpler receiver and to successfully demodulate signals received with extremely low power. The channel access is random both in time and frequency. The usage of randomness in the frequency domain makes it possible to limit the effects of the imprecision due to electrical components deterioration and oscillator jitter; gateways are configured to continuously scan the spectrum, listening in parallel on all frequency channels. Since no acknowledgment is used, the reliability of communication is increased by making the device transmit three times the same message, on randomly selected channel. In this way, the transmission benefits from both time and frequency diversity. Since Sigfox is also subject to regional duty cycle limitations, each device can send a maximum of 140 messages per day in UL and receive no more then 4 messages per day in DL. The data rate is 100 bps in UL and 600 bps in DL [7]; the claimed range is 30-50 km in rural areas and 3-10 km in urban environments. Although Sigfox technology is one of the most prominent solution in the IoT market, the limited number of messages per day is a major drawback, as well as the fact that the network is owned by Sigfox, which may limit the flexibility of the deployment.

Weightless Weightless is a set of three standards proposed by the British company Neul, acquired by Huawei in 2014. Each standard targets different use-cases, but each of them complies with IoT requirements of low-power, low-cost and large coverage. *Wightless-N* is based on narrow band technology with a star architecture. Transmissions are performed in the 868 MHz ISM band and there is only support for one-way communication that achieves a data rate of 30 - 100 kbps. The coverage range is 5 km also in urban environment, while the device cost is maintained very low [15]. *Wightless-P* improves the previous standard by adding DL communication, that permits the use of acknowledgments to increase the reliability. The channel band is 12.5 kHz and the access is performed using Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) schemes. The range is reduced to 2 km in urban environment. *Weightless-W* takes advantage of the available TV white spaces, so it uses the frequencies at 470 and 790 MHz. It also enables coverage and data rate adaptation to better fit the actual requirements. Time Division Duplexing (TDD) is used to provide UL and DL pairing, as spectrum is not guaranteed in the TV white space [15]. The range varies from 5 km (indoor) to 10 km (outdoor) and the rate from 1 kbps to 1 Mbps.

Ingenu Ingenu technology was proposed by the American company On-Ramp Wireless. The solution is based on the Random Phase Multiple Access (RPMA) scheme, working in the 2.4 GHz ISM band with a typical channel bandwidth of 1 MHz. Data are encoded, interleaved, and spread by a Golden Code, then the signal is randomly delayed before transmission. The spreading provides a processing gain and makes it possible to adapt the data rate to the propagation conditions. UL and DL transmissions are performed in an half-duplex way and their rate varies from 60 bps to 30 kbps [15]. The communication range has been estimated to be up to 10 km in urban environment.

LoRa LoRa is a widespread IoT technology, which is considered more flexible than others (such as Sigfox) and where only the PHY modulation is patented. Therefore, each user or company can buy LoRa devices and build its own network. The LoRa PHY layer makes it possible to adapt range and data rate to the different scenarios. LoRa actually refers to the modulation scheme at PHY layer, while LoRaWAN includes the specifications at MAC layer, which are promoted by the LoRa Alliance to set up the network and exploit the properties of the LoRa modulation at the best way possible. The coverage range varies from 2 to 5 km in urban areas and from 15 to 30 km in open space, while the raw data rate ranges between 250 bps to about 5.5 kbps. Since one of the objectives of this thesis is to simulate and analyze LoRaWAN's performance, the features of this technology will be further explored in Sec. 2.3 (LoRa PHY) and Sec. 2.4 (LoRaWAN specifications).

2.2 Narrowband IoT (NB-IoT)

NB-IoT was proposed by Third Generation Partnership Project (3GPP) in Release 13 as a standard focused on IoT and Machine Type Communication (MTC), with the aim of minimizing battery consumption at the price of a lower data rate and a higher latency. The standardization of NB-IoT was driven by 3GPP targeting the following main objectives [16]: 1) *Improved indoor coverage*: NB-IoT is meant to retain connectivity up to 20 dB beyond the limit of older technologies aimed at similar use cases, such as GPRS; 2) *Support for massive access of low-throughput devices*: the target is 50 thousand devices per cell, much larger than the typical number of broadband User Equipment (UE) per cell considered in the classic LTE access; 3) *Reduced complexity*: NB-IoT is meant for very cheap and possibly disposable devices, with reduced throughput requirements, for which the radio has to be very simple; 4) *Improved power efficiency*: the target devices are expected to be mostly battery powered, so that the energy efficiency of the communication is critical to provide a lifetime in the order of 10 years with batteries of 5 Wh (Watt-hours) capacity; 5) *Relaxed latency requirements*: most of NB-IoT use-cases are delay tolerant, and even in specific scenarios that require some delay guarantees (e.g., event notification systems), a delay of 10 seconds between action trigger and uplink transmission is considered reasonable by the specifications.

NB-IoT occupies a bandwidth of 180 kHz within the LTE spectrum, according to three possible options: (i) Standalone, where NB-IoT is placed in existing idle spectrum resources, (ii) Guardband, where the LTE guard bands are used for NB-IoT, and (iii) In-band, where in-band LTE resource blocks are assigned to NB-IoT [17] (Fig. 2.2).

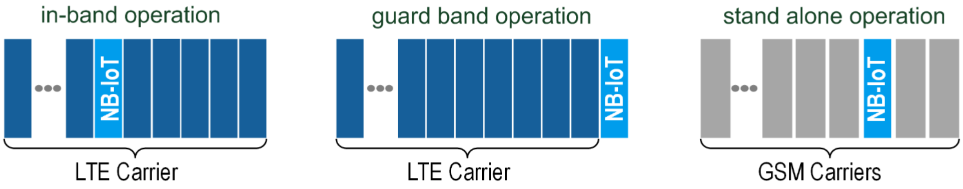


Figure 2.2.: NB-IoT operational modes [18]

Table 2.1.: Main features of NB-IoT [17, 19, 20].

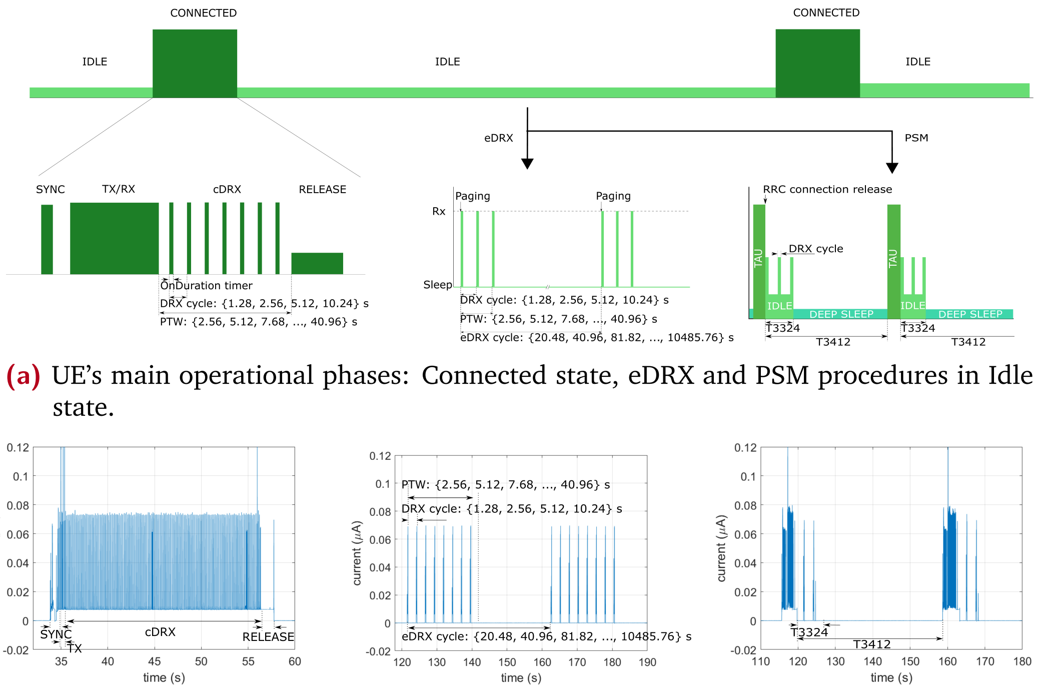
Feature	NB-IoT
Frequency	Licensed LTE frequency bands
Bandwidth	180 kHz
Theoretical peak data rate at the PHY layer	226.7 kbps (DL); 250 kbps (UL)
Range	∼1km (urban); ∼10km (rural)
Handover	Not available[1]
Latency	$\leq$ 10 s
Low power mechanisms	eDRX, PSM

The main features of the technology are listed in Table 2.1. Next, we briefly present the main operational modes and power saving mechanisms of NB-IoT.

2.2.1 Operational phases

This section describes the operational phases of a UE at different time scales, as schematically illustrated in Fig. 2.3a. At a macro timescale, the UE alternates between two main states: *Connected* and *Idle*. In Connected state, the UE maintains a control link with the network. When such link is released, the UE enters the Idle state. In both states, the UE periodically should check for the availability of DL messages at the Base Station (BS). To reduce power consumption, the UE can employ the Discontinuous Reception (DRX) mechanism (see Fig. 2.4), which consists in listening/sleep cycles whose time duration is specified by the *DRXCycle* parameter. The duration of the listening period inside a *DRXCycle* is specified by the *OnDurationTimer* and is expressed in multiples of ∼1 ms, corresponding to the duration of a Paging Occasion, i.e., a time interval during which the UE can receive notifications of pending packets from the BS. The values of *DRXCycle* and *OnDurationTimer* are set by the BS. The plots in Figs. 2.3b, 2.3c, 2.3d and Fig. 2.5 show some experimental current traces with periodic UL traffic. The periods of high current consumption

[1] The handover functionality is not considered in the standard: therefore, a new connection procedure is required for mobile devices entering in a cell covered by a different Base Station (BS). Further, NB-IoT does not officially support mobility. Cell reselection is intended only for attaching to a cell with better coverage.



(a) UE's main operational phases: Connected state, eDRX and PSM procedures in Idle state.

(b) Current trace showing UE's Connected state.

(c) Current trace showing eDRX procedure.

(d) Current trace showing PSM procedure.

Figure 2.3.: UE's operational phases: illustrative scheme and empirical traces of current consumption.

correspond to intervals in which the device is active transmitting, receiving or sensing the channel for possible DL messages. In the following, we examine in more detail the operations in Connected and Idle states.

Connected state

This actually consists of a combination of the following operations: synchronization, transmission/reception, listening and release, which are described below.

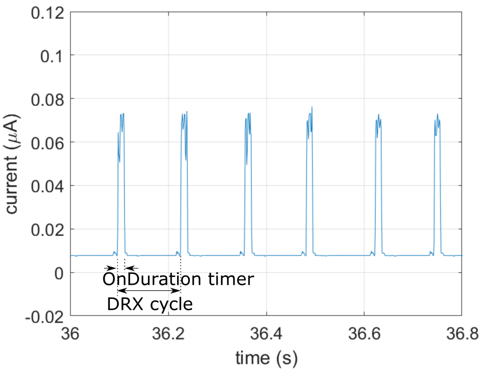
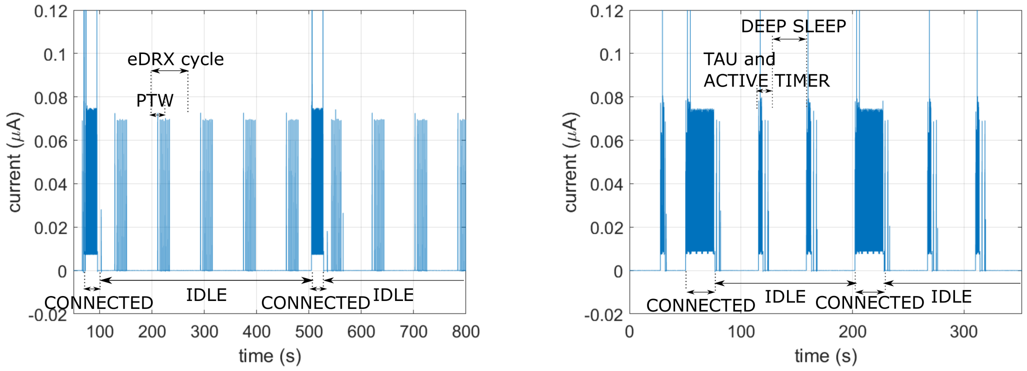


Figure 2.4.: Example of the DRX procedure in Connected state.



(a) Experiment showing Connected state and eDRX procedure.

(b) Experiment showing Connected state and PSM procedure.

Figure 2.5.: Current traces of two experiments where the UE alternates between Connected and Idle states.

Synchronization (SYNC) This phase is performed by the UE to re-synchronize with the network whenever it exits from the Idle state. If the UE does not have any allocated resources, it performs a random access procedure to initiate the communication with the BS. In our experiments, we observe that this phase can have a variable duration.

Transmission and Reception (TX/RX) This phase corresponds to the transmission of (one or more) UL messages, each followed by a reception interval where the UE waits for possible DL data or acknowledgement packets. In the current traces, the UL transmissions are preceded and followed by peaks of current consumption, as illustrated in Fig. 2.6. Such peaks correspond to control signaling traffic. The actual data transmission causes a lower peak, which lasts longer. In the sequel, we consider as transmission phase the time between the highest peaks, thus including signaling associated to the actual packet transmission. As previously discussed, when the UE exits the Idle state, the first TX/RX phase is preceded by a SYNC phase to establish a control channel with the BS. The UE then requests the allocation of transmission resources by performing a Service Request operation, which in our analysis is considered part of the TX/RX phase. Instead, if the connection was only suspended rather than being released, the Service Request is replaced by a Connection Resume procedure, which is lighter in terms of control signaling.

Listening Period In the Connected state the UE maintains the so-called inactivity timer, which is restarted at any RX/TX event. If the timer expires the UE performs the release operation and enters the Idle state. The

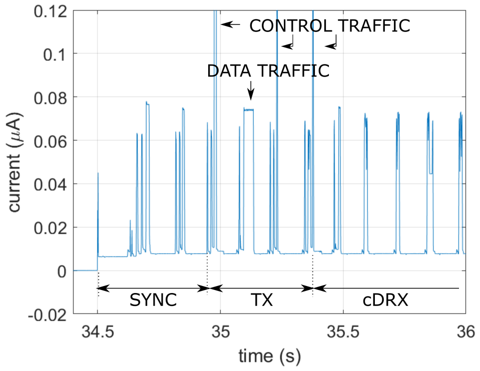


Figure 2.6.: First part of the Connected state.

value of the timer is set by the base station, typically in the range between 10 and 20 seconds. The UE can ask the network to set the value of this timer to zero using the Release Assistance Indicator (RAI) flag, explained in Sec. 2.2.2. In this case, the UE leaves the Connected state immediately after a TX/RX event.

While the inactivity timer is counting down, the UE might keep the radio on, always listening, or perform Connected state Discontinuous Reception (cDRX). During cDRX, the UE alternates between high energy periods of listening for scheduling information and low energy periods of sleeping. During sleeping periods, the radio consumes ~90% less energy. In case of available DL messages, the UE can directly perform a TX/RX without any SYNC operation. An example of the current consumption in cDRX is given in Fig. 2.4.

Release The UE releases the connection with the BS and leaves the Connected state entering the Idle state.

Idle state

In the Idle state, the UE may utilize two power saving mechanisms, in addition to normal DRX: eDRX or Power Saving Mode (PSM). These mechanisms, better described below, are based on timers that are negotiated with the network (see Table 2.2).

eDRX This mechanism is similar to cDRX, but with more sporadic listening periods. An eDRX cycle does indeed correspond to a sequence of DRX listening/sleep cycles, called Paging Time Window (PTW), followed by a long sleep period (see Figs. 2.3a, 2.3c, 2.5a for reference). The overall duration of an eDRX cycle is determined by the *eDRXcycle* parameter,

Table 2.2.: Main timers for DRX, eDRX and PSM. Note that timers used in DRX also apply to cDRX and eDRX, and that timers configurable by the UE are actually subject to the network approval, which could also set different values.

Mode	Timer	Description	Min Value	Max Value	Configured by
DRX	OnDurationTimer	Time spent in active listening	1 ms	200 ms	Network
	DRXcycle	Time interval between the beginning of two active listening periods	2 ms	2.56 s	Network
eDRX	PTW	Duration of Paging Occasions monitoring, composed of multiple DRX cycles	2.56 s	40.96 s	UE
	eDRXcycle	Time interval between the beginning of two PTWs	20.48 s	10485.76 s	UE
PSM	T3324	Active Timer: duration of DRX/eDRX within Idle state (listening for paging)	2 s	410 h	UE
	T3412	TAU timer: interval between two TAUs	2 s	410 h	UE
	Inactivity timer	Time spent in Connected state, after the end of the last TX/RX	0 s	65.536 s	Network

while the period in which the UE performs the DRX cycles is determined by the *PTW* parameter [21]. Therefore, the time difference between *eDRXCycle* and *PTW* gives the duration of the sleep period in an eDRX cycle. When not listening, the radio is off.

PSM This is the most effective power saving technique supported by NB-IoT. During PSM, the UE switches off its radio for a long period (deep sleep), but keeps its registration to the network: therefore, when exiting PSM, the UE just needs to perform a Connection Resume operation. Moreover, in PSM the UE periodically performs a Tracking Area Update (TAU) operation to communicate its location to the network. The PSM is characterized by two timers, namely T3412 and T3324 (see Figs. 2.3a, 2.3d, 2.5b for reference). The T3412 timer (or *TAUTimer*) defines the time interval between two TAU operations. Each TAU is followed by a period, whose duration is defined by the timer T3324 (*ActiveTimer*), during which the UE listens for paging, similarly to what happens during the PTW. After this time, the UE enters into a deep sleep state and is no longer reachable by the network. The UE exits the sleep state when T3412 expires or when a new UL data becomes available. Fig. 2.3d shows the current consumption for the TAU operation, followed by the listening for paging interval, whose duration is determined by the T3324 timer.

We observe that the TAU timer in PSM can be almost 17 days long. Therefore, a device entering in PSM will consume a minimum amount of energy, but may be unreachable from the network for several days if no UL transmission is required. On the contrary, when adopting the eDRX power saving mechanism, the UE can be contacted by the network within a limited time interval, but at the cost of higher energy consumption.

2.2.2 ECL and RAI

One of the objectives of NB-IoT is providing reliable communication to devices in harsh conditions, such as parking garages and ground pits. Therefore, the *Extended Coverage Level (ECL)* feature is introduced to tune the robustness of the communication. Robustness is primarily achieved by repeating the messages up to thousands of times, at the cost of a reduced data rate and an increased delay and energy consumption. The BS can set the ECL parameter based on the received Narrowband Reference Signal Received Power (NRSRP), a metric indicating the power of the LTE reference signals. The 3GPP identifies three different coverage levels, namely *Normal* (`ECL 0`), *Robust* (`ECL 1`) and *Extreme* (`ECL 2`), which are defined in terms of the target MCL, which is set to 144, 154, and 164 dB for the three levels, respectively.[2] Each level is associated to a certain setting of some transmission parameters, including the transmit power, the subset of subcarriers, the number of repetitions of random channel access, and the maximum number of transmission attempts. These result in prolonged transmission and reception under challenging conditions. In the worst case, in `ECL 2`, the number of repetitions may reach 2048 for the DL and 128 for the UL [22]. Consequently, the transmission delay may reach 10 seconds. The thresholds for each ECL class and the associated transmission parameters are determined by the operators. The BS monitors the signal strength of a target device on both the uplink and what the device reports for the downlink and decides its ECL level. The device does not have any control on the ECL parameter, but in our experiments, it was possible to retrieve its current value by using appropriate diagnostic commands.

[2] MCL is the largest attenuation between the transmitter and the receiver that can be supported by the system with a defined level of service.

Table 2.3.: Possible values of Release Assistance Indicator (RAI) field [23].

Flag value	Meaning
0x000	No flags set: remain in the Connected state for the duration of the inactivity timer
0x200	RAI: release after next UL message
0x400	RAI: release after next UL message has been replied to

Conversely, the UE can control the *Release Assistance Indicator (RAI)* flag that is carried into signaling messages before any UL transmissions. This flag is used to notify the BS that, after the upcoming UL transmission, the UE is expecting: (i) another UL transmission; (ii) a DL message; (iii) none of the previous. Based on this signaling, the BS can release the connection beforehand (see Table 2.3), so that the UE can reduce the time spent in cDRX phase, awaiting incoming DL transmissions. The effects of this parameter will be explored in the following sections.

2.3 LoRa modulation

LoRa (**Lo**ng **Ra**nge) is a proprietary physical layer modulation used to create long-range wireless links, developed by Semtech and based on the Chirp Spread Spectrum (CSS) modulation technique. Being covered by a patent, most of the information is found in semi-official documents published by Semtech, like [24, 25, 26].

2.3.1 LoRa's Chirp Spread Spectrum

The principle of CSS is to occupy with the transmission a bandwidth much larger than the one actually needed for the considered data rate. As explained in [15], CSS is a subcategory of Direct-Sequence Spread Spectrum (DSSS) and the receiver can benefit from controlled frequency diversity to recover weak signals and, thus, achieve a higher sensitivity. This way, the covered range is increased at the cost of a lower data rate. In CSS data are spread using *chirps*, sinusoidal signals whose frequency linearly increases in time, spanning all the available bandwidth. For example, if we assume that the frequency band that can be used for the transmission is $B = [f_0, f_1]$, a chirp can start at frequency $f_s \in [f_0, f_1]$, increase its frequency linearly till f_1 and then wrap around to f_0 to increment f_s. In particular, LoRa symbols are obtained as different circular shifts of the basic upchirp signal. These temporal shifts are slotted into multiples of the time $T_{chip} = 1/B$, called *chip* where $B = f_1 - f_0$,

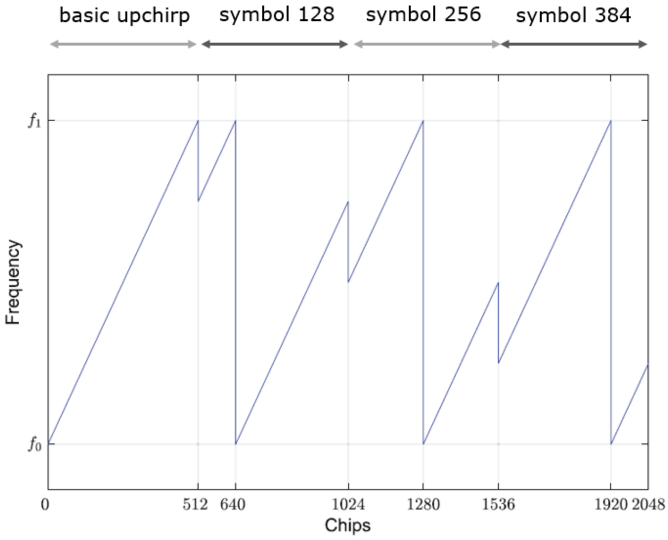


Figure 2.7.: Modulating signal with SF = 9 for one basic upchirp and three symbols: 128, 256, 384 [27].

and they characterize each symbol. The modulating signal for a generic n-th LoRa symbol can be expressed by the following equation

$$f(t) = \begin{cases} f_1 - n \cdot k \cdot T_{chip} + k \cdot t, & \text{for } 0 \leq t \leq n \cdot T_{chip} \\ f_0 + k \cdot t, & \text{for } n \cdot T_{chip} < t \leq T_s \end{cases} \tag{2.1}$$

where T_s is the LoRa symbol time and $k = (f_1 - f_0)/T_S$ is the slope of the frequency variations [27]. For the sake of clarity, Fig. 2.7 depicts the basic upchirp and three modulated signals.

2.3.2 Configurable parameters

When transmitting a signal using LoRa modulation, different parameters can be set to tune its data rate and robustness to channel impairments.

Spreading Factor (SF) The total number of symbols (each one coding SF information bits) is $M = 2^{SF}$, where SF is a tunable parameter called spreading factor and goes from 7 to 12. The symbol time can be obtained as

$$T_s = 2^{SF} \cdot T_{chip} = \frac{2^{SF}}{B}, \tag{2.2}$$

and it can be noted that, for an increase of 1 in the SF, the symbol time doubles. When the symbol duration increases, more energy is put into each symbol, and this makes the signal more robust to noise and

Table 2.4.: SNR values at LoRa demodulator for different SFs [28, 29].

SF	LoRa Demodulator SNR (dB)
7	-7.5
8	-10.0
9	-12.5
10	-15.0
11	-17.5
12	-20.0

interference, reaching longer distances. However, SF also influences raw data rate R_b at PHY layer, which is equal to

$$R_b = SF \cdot \frac{B}{2^{SF}}, \tag{2.3}$$

and, therefore, the data rate is lower when a higher SF is used. Moreover, transmission lasts longer for higher SFs (Eq. 2.2) and then it is more subject to interference and collisions.

In the datasheets [28, 29] we find the values of Tab. 2.4 showing that a higher SF leads to a better Signal to Noise Ratio (SNR) at the receiver.

Bandwidth (B) A second parameter that can be set in LoRa is the bandwidth (125, 250, 500 kHz). In general, a larger bandwidth yields higher data rate and a worse receiver sensitivity, and is used together with the SF to adapt to channel conditions. The availability of different bandwidths depends on the region the network is operated in, since regional regulations apply different constraints (see Sec. 2.4.6).

Carrier frequency The third configurable parameter is the carrier frequency. LoRa uses sub-GHz ISM bands, but the center frequency and channel division depend on regional regulations. The carrier frequency in use can be changed by the users, which can choose channels with more lenient duty cycle limitations to exchange more data [30].

Code Rate (CR) LoRa provides support for Forward Error Correction (FEC) with configurable Code Rates (CRs). The code rate is defined as $CR = n/k$ where n is the number of information bits in the word and k is the total length of the word. Having $k > n$ means that redundant bits have been added to correct errors. In LoRa, the code rate is given by $CR = 4/(4 + r)$ where $r \in 1, 2, 3, 4$. Therefore, the set of possible code rates is {4/5, 4/6, 4/7, 4/8}. Reducing the CR provides more protection against error bursts, but also increases the packet transmission time.

2.3.3 Additional LoRa modulation properties

Besides to the use of CSS modulation, that allows the recovering of signals having very weak power, LoRa modulation presents two particular advantages that make it very competitive in the IoT market: spreading factor orthogonality and capture effect.

Spreading Factor Orthogonality LoRa spreading factors are pseudo-orthogonal, even when the same channel and the same bandwidth are used. This means that if an ED transmits using spreading factor i, it can be correctly received by the GW even if it is overlapping with another transmission using spreading factor j, as long as $i \neq j$. Even if in [31] the LoRa spreading factor orthogonality is mathematically demonstrated, different works ([27], [32]) state that the orthogonality is not perfect, but, depending on the signal power, some spreading factors can mutually interfere. It is also observed that lower SFs are more susceptible to interference than higher SFs.

Capture effect When multiple packets transmitted with the same SF overlap in time and frequency, instead, they may generate destructive mutual interference, disrupting each other's reception and resulting in what is called a packet collision event. However, if one signal is significantly stronger than the others, with a power margin greater than the so-called "co-channel rejection parameter" ρ, then it can be correctly received despite the interference, giving rise to a *capture* phenomenon. The co-channel rejection threshold ρ is specified in LoRa chip datasheets [25, 28], and can be as large as 25 dB for SF 12, as estimated in [32]. Literature works, as [33], show the importance of this feature for the total performance of the network and underline that it should be taken into account when planning or simulating a LoRa network. Capture effects and spreading factors pseudo-orthogonality make LoRa's performance better than that of a classic Aloha network: even though the channel access is still Aloha-based, overlapping packet in LoRa can survive if their power at the receiver is sufficiently high, and this margin is even lower for packets transmitted with different SFs.

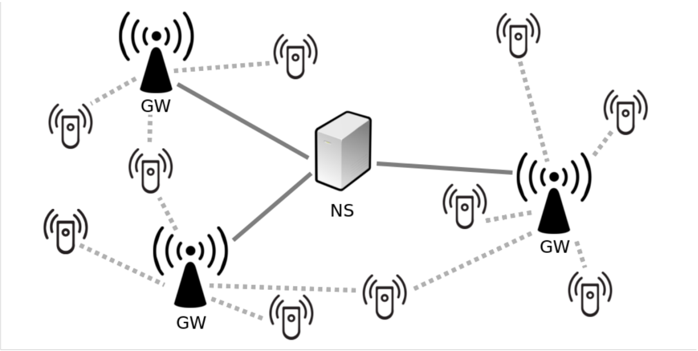


Figure 2.8.: LoRaWAN Infrastructure.

2.4 LoRaWAN standard

While the PHY layer specifications are proprietary of Semtech, the documentation describing the protocol at MAC layer, called LoRaWAN, is public and developed by LoRa Alliance, an association initiated by industry leaders, now extended to vendors and researchers, having the objective of spreading LoRa technology and guaranteeing interoperability among different operators.
The standard defines the use of a star topology, with Gateways (GWs) acting as transparent message forwarders between End Devices (EDs) and the Network Server (NS). EDs use LoRa to connect to one or more GWs, that, in turn use IP links to connect to the NS.
When the EDs transmit, the LoRa packets are collected by all the GWs in their coverage range and forwarded to the NS, which discards duplicates and chooses the best GWs for the return transmissions to the EDs. It is worth noting that the GWs do not support full-duplex transmission: to send a DL packet (from the NS to an ED), they have to interrupt any ongoing UL reception.

2.4.1 Classes of devices

LoRaWAN distinguishes three classes of EDs. They all provide bi-directional communication, but differ in functionalities and power constraints. Each device must implement at least the Class A features and optionally the others.

Class A (All) Devices belonging to this class access the channel to transmit the packet received from the application layer in a totally asynchronous fashion, implementing an Aloha Medium Channel Access Control. Each ED's UL communication is followed by at most two short DL receive windows, and the second one is opened in a different sub-band previously agreed with the network server in order to increase resilience against channel fluctuations. These are the most power-saving devices as receiving windows are opened only after a transmission starting from the EDs. Downlink communication could, thus, suffer a high latency. These nodes are meant to be used for monitoring applications.

Class B (Beacon) In addition to the functionalities of Class A's devices, Class B devices periodically open reception windows, following the scheduling information provided in periodic beacon messages sent by the GWs. This allows the NS to know when the device is listening and makes these nodes suitable for remote control applications.

Class C (Continuously listening) Class C devices usually are powered by the main electric grid and consequently can keep the reception window always open to get DL messages, when not transmitting.

In the reminder of this thesis, if not explicitly stated, we will refer to the most common Class A devices, which are expected to be battery powered, in accordance with a typical IoT scenario. In the following, we will better describe the functioning and operational phases of a LoRaWAN Class A device. After the transmission of each UL message the Class A ED opens two short receive windows, which are the only possibility for the communication from NS to ED. The start times of the receive windows are defined using the end of the UL transmission as reference (see Fig. 2.9).

The frequency and data rate of the first receive window (RX1) are function of the corresponding values used for the UL. By default, they keep the same data rate of the last UL message. The first receive window is opened $RECEIVE\_DELAY1$ seconds after the end of the transmission. The length of the receive windows can be variable, but must be at least the time required by the ED's radio to detect the DL message's preamble. If there is such a detection, the radio receiver stays on until the message is demodulated. If, during the reception, the ED notices, from the address field, that the message is intended for another user, it stops the reception discarding the packet. In this case or when RX1 goes empty, the ED must open the second receive window (RX2) $RECEIVE\_DELAY2$ seconds after the end of the UL transmission. Frequency and data rate of the second receive window have fixed values that can be configured by MAC commands

A transmission from the NS to the ED must initiate exactly at the beginning of one receive window and an ED shall not begin to transmit a new UL message before it either has received a DL message or the second receive window has expired.

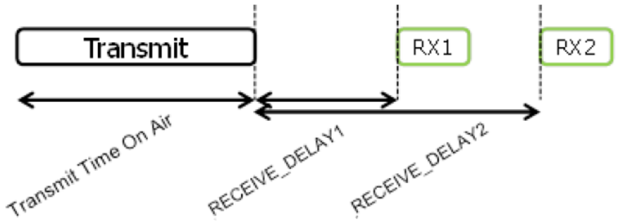


Figure 2.9.: End device receive slot timing [26].

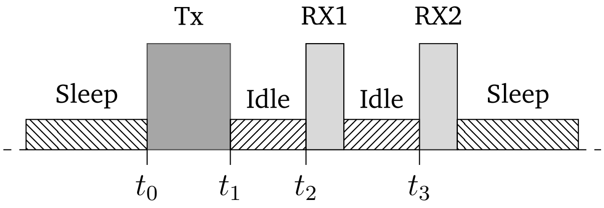


Figure 2.10.: Example of operational phases for a Class-A ED.

Fig. 2.10 represents the sequence of operations performed by a Class A device, using the default values described in Sec. 2.4.6 as time intervals between the opening of the reception windows. The ED can wake up at anytime (time t_0) to transmit its data. Then, it enters into idle state and 1 s after the end of the transmission, at time t_2, it opens RX1. During RX1 the ED listens for possible incoming packets for the time needed to detect a preamble. If no reception is started, the ED switches to idle state again. At time $t_3 = t_1 + 2s$, RX2 is opened, providing a second opportunity to receive DL packets. Then, the ED enters in sleep state until the next packet transmission. In the following, we will indicate as transmission *cycle* the period between two consecutive packet transmissions, including all the intermediate phases.

Note that these operational phases also correspond to different values of the current consumption, according to the operation that is performed by the device. More details about this aspect can be found in Sec. 6, where the current consumption is considered.

2.4.2 Packet frame

At the PHY layer, a LoRa frame consists in three parts: preamble, an header and a payload. Also, the header and the payload can be followed by two Cyclic Redundancy Check (CRC) fields for error correction. The preamble has a synchronization function and defines the packet modulation scheme, since it is modulated at the same SF as the rest of the packet. The header can be implicit or explicit, depending on the chosen mode of operation.

It is possible to compute the packet time-on-air with the formula given in [28]:

$$T_{packet} = T_{preamble} + T_{payload}, \tag{2.4}$$

where

$$T_{preamble} = (n_{preamble} + 4.25) \cdot T_s, \tag{2.5}$$

with $n_{preamble}$ the programmable number of symbols that constitute the preamble and T_s the symbol period.
The payload duration depends on the header mode that is enabled, and is computed as

$$T_{payload} = n_{payload} \cdot T_s \tag{2.6}$$

with the number of payload symbols $n_{payload}$ given by the following formula:

$$n_{payload} = 8 + max\left(\left\lceil \frac{(8PL - 4SF + 16CRC - 20IH)}{4(SF - 2DE)} \right\rceil (CR + 4), 0\right). \tag{2.7}$$

Different parameters are defined as follows:

- PL is the number of bytes of payload;
- SF is the spreading factor;
- IH indicates which header mode is used: $IH = 1$ for implicit header mode, $IH = 0$ for explicit header mode;
- DE set to 1 indicates the use of low data rate optimization, otherwise it is set to 0;
- CRC is equal to 1 when the CRC field is present, 0 otherwise;
- CR is the programmed coding rate (from 1 to 4).

Fig. 2.11 shows the spectrum of the transmission of one LoRa packet: at the beginning, the preamble sequence of constant upchirps spanning the entire available bandwidth is visible. Fig. 2.12 represent the results of the decoding process used in [34] to analyze a LoRa modulated signal. To obtain it, the authors first "de-chirp" the signal, then take its Fast Fourier Transform (FFT) with a number of bins equal to the number of symbols M that can be represented with the spreading factor in use. Secondly, since the signal can now be interpreted as if it was modulated using Multiple Frequency Shift Keying (MFSK), they take multiple overlapping FFTs and, to detect the symbol at each time frame, select the bin with the highest power content. Again,

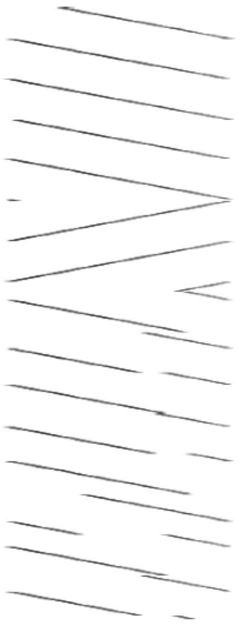


Figure 2.11.: Spectrogram excerpt of a LoRa chirp. *x-axis*: Frequency, *y-axis*: Time [34].

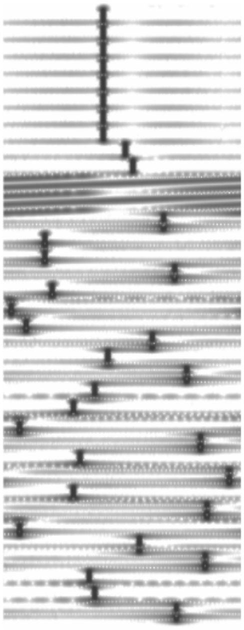


Figure 2.12.: Dechirped signal (preamble and body) [34].

the constant part at the beginning indicates the constant symbols used in the preamble.

To make data more robust, LoRa perfors a set of operations on packets at the PHY layer before sending them:

Interleaving This procedure scrambles data bits throughout the packet and is often used with Forward Error Correction (FEC): in this way, if interference affects data, corrupted bits will be spread across the whole packet after de-interleaving, avoiding bursts of errors and making the correction operation more effective. In [34] it has been found that the interleaver is a diagonal interleaver with the two most significant bits reversed, slightly different from the one described in Semtech's patent.

Forward Error Corrections (FECs) LoRa uses Hamming FEC with a variable codeword size in the range from 5 to 8 bits and fixed data size of 4 bits per codeword, as previously described.

Data Whitening This technique is applied to transform the sequence of symbols, providing more features for clock recovery. It is particularly useful when there are long runs of equal bits. To recover the signal, data are XORed against the same whitening sequence used by the transmitter.

Gray map This encoding procedure maps a block of SF bits into one of the M symbols in the costellation and, by definition, makes sure that two adjacent symbols differs by at most one bit. In this way, when applying FEC it is easier to successfully correct symbols that differ by one bit.

For the principle of encapsulation, the packet at MAC layer is the payload of the PHY packet , which consists of a 1-byte-long MAC Header (MHDR), a MAC Payload (MACPayload) with variable size, and a Message Integrity Code (MIC) of 4 bytes, checking the integrity of the whole message. In particular, the MHDR specifies the Message Type (MType) of the packet and according to which version of the LoRaWAN specification the frame has been encoded. In Tab. 2.5 we have the codes used to indicate the type of the message and the corresponding description. We focus now on the four types used for data messages: they are employed for UL or DL communication (*Data Up* and *Data Down*) and can require an acknowledgment from the receiver, or not (*Confirmed* - *Unconfirmed* data). We remark that for Class A EDs the only possibility that the NS has of acknowledging UL traffic is during the receive windows, while if an ED is asked for an acknowledgment, it should answer whenever it is possible. MAC commands often require the ED to confirm the message reception and command execution.

Table 2.5.: MAC message types [26]

MType	Description
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	Reserved for Future Use
111	Proprietary

MAC commands are used for network administration and can be exchanged in both UL and in DL directions, enabling the following functionalities:

- Resetting;
- Link checking: commands sent by EDs to validate their connectivity to the network and answered by network server with information about the received signal power;
- ADR setting (see Sec. 2.4.4);
- Duty cycle limitations, set by the network server;
- Setting of different parameters (reception slots, transmission power, DL channel...);
- Exchange of information about the status of the ED (battery level and demodulation margin);
- Creation or modification of radio channels.

2.4.3 Retransmission procedure

As previously discussed, transmissions can either be unconfirmed or confirmed. In the first case, a message is transmitted only once and is not expected to be acknowledged by the NS, while in the latter case, messages are retransmitted until an ACK packet is returned by the receiver.

When a device (both ED and NS) receive a confirmed data message, it shall answer with a data frame having the ACK bit set: the server will answer in one of the two receive windows while the ED will reply at its own discretion, by sending an empty frame with ACK immediately after the reception or piggybacking it to a following data message. Uplink messages are transmitted $NbTrans$, unless a valid DL is received following one of the transmissions. It is used to control the redundancy and obtain a given Quality of Service (QoS) that can depend on the application in use. It is suggested to use frequency hopping also in the retransmission procedure and to delay retransmissions until the receive windows have expired. Moreover it is recommended to stop further retransmissions if the corresponding DL acknowledgment frame is received.

Consequent retransmissions should be done at least $ACK\_TIMEOUT$ seconds after RX2, with the value of $ACK\_TIMEOUT$ is randomly chosen in the interval [1, 3] s. The parameter $NbTrans$ can be set by appropriate MAC commands and its value ranges from 0 to 15. In the following chapters, we will indicate with m and h the value of $NbTrans$ for confirmed and unconfirmed traffic, respectively.

In the previous version of the LoRaWAN standard [35] it was also remarked that in addition to selecting a different frequency channel, retransmissions could also use a different (lower) data rate: the recommended scheme was to decrement the data rate every couple of failed transmissions, until the lowest possible data rate is reached. Any further transmission will employ the last data rate used, and the recommended maximum number of transmissions is 8. For example, suppose that an ED sends a confirmed frame using DR5. If it does not receive any ACK, after the eight re-transmission, it will be using DR2 and the packet will be dropped. A following application packet will initiate a new transmission cycle, where the first two attempts will be done at DR2, then, in case of failure, the end device will switch to DR1, then to DR0. As shown in Sec. 4.3, this behavior is generally damaging for the network performance, and it is not longer recommended in the later versions of the LoRaWAN standard. [26]. Nonetheless, it is currently adopted in some public implementations, such as that used by [36].

2.4.4 Adaptive Data Rate (ADR)

The LoRaWAN standard provides the Adaptive Data Rate (ADR) mechanism that allows the NS to control the transmission parameters of the EDs in order to optimize the performance of either the device itself or the network as a whole. Since each ED is allowed to use any data rate and transmission power, the network, via the NS, has the possibility of controlling them and this is referred to as ADR. This is possible only when the UL ADR bit is set, informing the NS that it can send appropriate MAC commands to modify ED's transmission parameters. The NS will then choose to decrease SF and transmission power if the ED transmissions are received above sensitivity or increase them if the SNR margin is too low. If the ADR bit is unset, the ED controls its own parameters. It is worth noting that the LoRaWAN specifications do not describe any ADR algorithm: its implementation is left to network owners (which control the NS) that should find a compromise between computational efficiency and accuracy. Some papers have been recently published about this topic, proposing and/or evaluating different algorithm, which will be described in Sec. 4.1.

2.4.5 Device activation

The specifications [26] define two ways to activate an ED and allow its participation in a LoRaWAN network: Over-The-Air Activation (OTAA) and Activation By Personalization (ABP).

In the first one the ED sends a request to join the network; the network server checks if that specific device is allowed to join the network (if, for example, the device has a valid subscription or has not been banned from the network for security problems) and, if so, answers with a join accept message. During this procedure, the ED is also given keys (an application key and a network key) for message encryption and decryption. Higher security is given by the fact that these keys are computed based on root keys specific to the ED and assigned to it during fabrication: if a key is extracted from an ED, it only compromises that specific ED and not the whole network. The over-the-air procedure can be repeated multiple times (for example, every day) and each time new keys are computed and provided to the device, further increasing the security of the network. With the activation, the ED is also given a device address (DevAddr) that uniquely identifies it in the network.

With the ABP procedure, the ED can participate in the network as soon as it is started, without requiring the joining procedure, since the device address and all the keys are already stored into the device.

2.4.6 Regional parameters

In addition to the LoRaWAN specifications, the LoRa Alliance also publishes a document, that describes how LoRaWAN should be used and configured in the different regions. As already mentioned, this is because the available ISM band is not the same everywhere, and the local regulations can be more or less strict. In particular, the document provides information about preamble format, channel frequencies, data rate, maximum payload size, maximum transmission power, MAC commands, Class B implementation and default settings. In the following of this section, we better describe the elements that mostly affect the protocol performance. Tab. 2.6 reports the frequencies of the ISM band in the different regions. In the following, we will focus our discussion on the European configurations.

Frequency plan and transmission power

For the European region the protocol defines the use of the 868 MHz frequency, where network operators can freely assign channels. However, the

Table 2.6.: Regions and LoRaWAN frequency bands [37].

Region	Frequency band (MHz)
Europe (EU)	863 - 870 and 433
United States (US)	902 - 928
China (CN)	779 - 787 and 470 - 510
Australia (AU)	915 - 928
Asia (AS)	470 - 510
South Korea (KR)	920 - 923
India (INDIA)	865 - 867

specifications define a minimum set of channels (see Tab. 2.7) that must be implemented and cannot be modified, so as to guarantee in any occasion a minimal common channel set between ED and GWs. In fact, each GW should always listen to at least these three bi-directional channels, which are located at 868.1, 868.3, 868.5 MHz, and have a channel bandwidth of 125 kHz. Additionally, other channels could be defined to improve the network capacitance: for example, in [36] five additional channels are configured in the 867 MHz band.

Even if the ISM band is unlicensed, users have to respect some regulations dictated by different entities: the National Administrators, organisms at the European level and the International Telecommunication Union (ITU) at the worldwide level. In particular, in European countries, LoRa is required to respect the European Telecommunications Standard Institute (ETSI) regulations. The restrictions concern the physical medium access, such as maximum transmission time. Radios are required to either adopt a Listen Before Talk (LBT) policy or duty cycled transmissions, to limit the rate at which messages are generated. The Duty Cycle (DC) is defined as the ratio of the maximum transmitter “on” time over one hour, expressed as a percentage. For example, a device with a 1% DC, can perform 10 transmissions of 3.6 seconds within one hour, while if the same device had a 10% of DC, the transmissions could last 36 seconds. To keep ED simple and save energy, the DC option is adopted in LoRa. This can have an important influence when the packets are generated frequently or when the transmitter uses high SFs (and thus longer transmission times) since it limits the activity of the device. The 868.1, 868.3, 868.5 MHz channels belong to the same regulatory sub-band (Sub-Band 1 (SB1)), and have to share a DC limitation of 1%, while the channel reserved for DL, located in the 869 MHz sub-band (Sub-Band 2 (SB2)), can benefit from a more lenient DC of 10% and a higher transmission power.

Table 2.7.: Mandatory channels in EU863-870 ISM band [37, 38].

Sub-band	Frequency [MHz]	Bandwidth	Use	Duty Cycle	Maximum ERP
SB1	868.1	125 kHz	UL/DL	1%, shared	14 dBm
	868.3		UL/DL	1%, shared	14 dBm
	868.5		UL/DL	1%, shared	14 dBm
SB2	869.525	125 kHz	DL	10%, dedicated	27 dBm

Table 2.8.: Minimum transmission interval allowed by a duty cycle of 1% for different packet sizes, SF 7 and additional 9 B due to the MAC header.

Payload size [B]	Packet duration [ms]	Min interval [s]
0	41.22	4.12
5	46.34	4.63
50	112.90	11.29
100	184.58	18.46

The LoRaWAN specifications also allow for the use of a single 250 kHz channel with SF 7 for the transmission of UL packets in the European region. The wider bandwidth makes it possible to transmit the same message in half the time, reducing the collision probability. Once the bandwidth is defined, there is a one-to-one correspondence between SFs and Data Rates (DRs), as defined in Tab. 2.9. Here, we also give the values of the indicative bit rate at the PHY layer. According to the LoRaWAN specifications, the rates indicated with DR0 to DR5 correspond to SF 12 to 7 (respectively) with 125 kHz channel, while DR6 corresponds to SF 7 with a 250 kHz channel.

ETSI also limits the effective power radiated by the devices. In [37] the possible values for the maximum transmission power are given, with the maximum value being equal to 16 dBm.

Default configurations

For Class A devices, the standard requires the RX1 to be opened in the same frequency channel and with the same SF used for the UL communication. The RX2, instead, is always opened on the dedicated 869.525 MHz channel and with SF 12, in order to maximize the robustness of the communication. These standard settings can be changed by using appropriate MAC commands sent by the NS. However, for RX1, only SFs higher than that used in UL can be set, since the ED is assumed to use the lowest SF that allows the GW to receive its signal. If a lower SF were used in DL, the message could arrive at the ED with a power level below the sensitivity, preventing its correct reception.

Table 2.9.: Data rate table for EU863-870 band with bandwidth of 125 kHz [37].

DataRate	Configuration	Indicative physical bit rate [bit/s]
DR0	LoRa SF12/125 kHz	250
DR1	LoRa SF11/125 kHz	440
DR2	LoRa SF10/125 kHz	980
DR3	LoRa SF9/125 kHz	1760
DR4	LoRa SF8/125 kHz	3125
DR5	LoRa SF7/125 kHz	5470
DR6	LoRa SF7/250 kHz	11000

Table 2.10.: Default values of some parameters in the EU863-870 band [37].

Parameter	Value
RX1 frequency	same as UL transmission
RX1 data rate	same as UL transmission
RX2 frequency	869.525 MHz
RX2 data rate	DR0 (SF12, 125 kHz)
ACK_TIMEOUT	2 ± 1 s
RECEIVE_DELAY1	1 s
RECEIVE_DELAY2	2 s = RECEIVE_DELAY1 + 1 s

Class C devices, instead, when not transmitting or receiving in RX1, are always listening in RX2 using the default parameters. These and other default parameters, such as the time interval between the opening of the reception windows, are reported in Tab. 2.10.

2.4.7 Additional considerations on LoRaWAN devices

In order to take advantage of the powerful feature of orthogonality between different SFs, the SX1301 LoRa PHY chipset, typically employed in GWs [25], provides 8 parallel demodulation chains, which allow the chip to demodulate up to 8 different signals simultaneously, irrespective of their SFs and frequency. We also remark that the GWs do not support full-duplex transmission and reception: in order to send a DL packet they have to interrupt any ongoing reception, regardless of the frequency channels in which transmission and reception occur.

We further notice that, as described in the corresponding datasheets [25, 28, 29] GWs and EDs have different sensitivities, with the GW chipset having higher capability of receiving weaker signals, as reported in Tab. 2.11. This introduces a sort of asymmetry in the channel, which will be further discussed in Sec. 4.3.4. Note that for the ED we reported the values for the chip

Table 2.11.: Sensitivity comparison.

SF	GW Sensitivity (dBm)	ED Sensitivity (dBm)
7	-130.0	-124.0
8	-132.5	-127.0
9	-135.0	-130.0
10	-137.5	-133.0
11	-140.0	-135.0
12	-142.5	-137.0

SX1272 [28]; for the chip SX1276, the sensitivity values for a LoRa communication with 125 kHz bandwidth are slightly different, going from -118 to -136 dBm.

Simulation software 3

3.1 ns-3 `lorawan` module

To perform simulations of network with LoRaWAN devices, we leverged the ns-3 simulator, a popular discrete-event network simulator written in C++. In particular, the first version of the ns-3 `lorawan` module used for the evaluations presented in this thesis was proposed in [39]. The module models the protocol stack of LoRaWAN EDs and GWs at the PHY and MAC layers, including all the protocol features, as well as the behavior and functionalities of the LoRaWAN NS. The module was then extended with the support for downlink traffic and re-transmissions [40], and then further improved to support various network configurations and test the proposed ameliorations, with the aim of gaining a deeper understanding of the role played by each configurable parameter and identifying unforeseen behaviors.

The `lorawan` module [41] consists in a collection of C++ classes, representing the network components at different layers (PHY, MAC). Additionally, these classes leverage "helper" objects to keep track of interference, correctly manage the state of the DC, update the list of available channels, compute the path loss between two locations and perform other similar tasks. Other classes are designed to model different kinds of applications, and NS's capabilities.

Currently, the module supports both confirmed and unconfirmed messages, permits the configuration of multiple network parameters, and implements a realistic model of the GW chip, accounting for the eight available demodulators, as better detailed in the following. Note also that in Sec. 6, we combine this module to a new ns-3 implementation of a battery-less capacitor-based IoT node, testing whether such a solution is feasible in LoRaWAN networks. More details about the ns-3 implementation of that specific application can be found in Sec. 6.

3.1.1 PHY layer model

The abstraction of the PHY layer implements the transmission chain at this level, manages the different states of the device, and aims at making interference computations more manageable. In particular, interference between colliding packets at the PHY layer is computed by considering a model of the GW layer, and a pair of look-up tables representing the chips' sensitivities and conditions that should be fulfilled by the received power. These models are employed to determine the success of a packet reception. If the packet's reception power is above the receiver's sensity, the receiver can lock on the packet. Then, the simulator performs other controls to determine whether packets survive eventual collisions, considering the time overlap and power difference between the two signals, which should be above the co-channel rejection parameter. This value is defined in the interference matrices, which model the LoRa modulation capabilities [15, 32]. Further details about the interference modeling can be found in [39].

The classes modeling the PHY layer are responsible of the switching between the different states, which can either be triggered by MAC layer operations or by the energy level. The states represent the chip's transmission, reception, power save mode and listening phases, which are characterized by different values of the current consumption. The representation of the ED's states, paired with a model for the energy consumption in each one of these states used in [42], provides an integration with ns-3's `energy` module, and allows users to run simulations in which the power level of EDs is also accounted for. Additionally, to model the intermittent behavior of a battery-less device, an Off and a Turn On state have been also introduced, as well as an Idle state which considers the fact that in real devices the current consumption when waiting for the opening of the reception windows is higher than that in the Sleep state. More details on these aspects are provided in Sec. 6.

Additionally, the classes modeling the GW's PHY implement the GW behavior, featuring a realistic modeling of the parallel decoding capabilities available in Semtech's SX1301 chip (which is currently employed in GWs). Indeed, it assumes that a single LoRa gateway has 8 demodulators working in parallel, as explained in [25]. These 8 demodulators are connected to the same antenna, and any SF can be received without prior configuration on any receive path, meaning that multiple packets can be demodulated simultaneously on different frequency channels. Incoming packets that find the GW's PHY layer model in a state where all 8 demodulators are occupied, are dropped. On the other hand, the PHY layers of EDs, are structured to be only able to receive a packet at a time.

3.1.2 MAC layer model

The `lorawan` module implements the LoRaWAN specification of Version 1.1, including devices of Class A and Class C.

Channel access is managed by keeping track of the available frequencies and of the set of sub-bands in the region being considered. Before performing any transmission (i.e., forwarding the packet to send to the object modeling the PHY layer), the MAC layer model checks whether a transmission on the desired frequency would break DC limitations. If this is the case, the transmission can either be dropped or delayed, taking then place when the DC restrictions allow for it. Otherwise, the packet is transmitted, and its parameters (i.e., the frequency used for the communication and the transmission duration) are then used to determine the following silent time imposed by the DC in that sub-band. The frequency for UL transmission is chosen at random among the available UL channels for the EDs, and is determined by the NS in case the GW is to perform the LoRa transmission.

The classes modeling the MAC layer complexity are also responsible of the logic handling the opening of the receive windows and re-transmissions of a packet. In particular, if the packet that is being transmitted is confirmed and no ACK is received in any of the two receive windows, EDs are assumed to be able to re-transmit that same packet up to a maximum number of times that can be set by the NS. If a new packet needs to be sent by the application layer, the packet that is currently in the re-transmission process is dropped, and the new transmission is performed as soon as possible. Re-transmissions are otherwise performed as soon as possible (always taking the DC restrictions modeled by the channel access manager into account) if no successful reception of an ACK was completed after the closing of the second receive window.

3.1.3 Application layer models

The behavior of the Application Layer is modeled differently based on the device under consideration. EDs are generally assumed to be equipped with a `PeriodicSender` application that generates traffic with a fixed packet length and period; the generated traffic can be either confirmed or unconfirmed. However, when analyzing specific scenarios, such as the feasiblity of battery-less devices, applications that generate traffic also according to the stored energy are considered, and are better detailed in Sec. 6. GWs, instead, run a `Forwarder` application that is connected both to the LoRaWAN MAC Layer and to the reliable point-to-point connection that links each GW to the NS, and whose only task is to act as a bridge between the two.

Finally, the NS node runs a `NetworkServer` application that is split in three main components, to better model the logic of the controller, which is responsible of the whole network management. The `NetworkStatus` component holds a data structure modeling the NS's view of the network, which takes shape as long as it receives messages from the EDs. This representation contains a list of addresses of the EDs participating in the network, together with their channel access parameters, such as SF and maximum number of re-transmissions. This data, together with a list of packets received from each ED, can be accessed by the `NetworkScheduler`, responsible of scheduling DL transmission events to respond to any UL transmission that needs a reply by the NS or to send any DL commands, such as that implementing the ADR mechanism. Finally, the `NetworkController` creates and sets all required fields of such DL packets, based on the data available in the `NetworkStatus`'s data structure. For instance, it is the `NetworkController`'s responsibility to set the ACK bit, to aggregate the correct MAC Command to implement the desired ADR algorithm, and defining which parameters to use for the DL transmission. More details on the module architecture are available in the online documentation [43].

3.1.4 Other module features

Aside from models describing the behavior of LoRaWAN networks, the `lorawan` module also contains propagation models that are useful to perform simulations in different contexts, from the traditional Smart City scenario, also in the presence of buildings and spatially correlated shadowing, to open air scenarios where only the path loss is present, to more challenging propagation environments, such as industrial plants, where the communication can be heavily affected by channel impairments. In the rest of this thesis, the specific propagation models employed are described when illustrating scenarios of interests.

We also remark that the simulation module contains some usability features to facilitate the analysis and the setup of the network: for instance, helper classes are provided to facilitate users in correctly setting up working LoRaWAN networks. These functions can be used to automate the creation of reliable connections between each GW and the NS, and the setting of the SF parameter at each ED, either considering user-defined distributions or based on distance between the ED and the nearest GW, in order to ensure that the two devices will be able to communicate. Furthermore, a packet counting facility was created to keep track of all messages that are exchanged in the network and easily compute some metrics of interest. Such metrics include the packet success rate for confirmed and unconfirmed traffic, and the delay from UL packet generation to delivery to the NS and from UL packet generation to delivery of the ACK to the ED. Further details on the computation of metrics are given in each chapter, as needed.

3.2 Comparison with other LoRaWAN simulators

Some of the available tools to simulate LoRaWAN networks are described in [44], where the authors survey LoRa and LoRaWAN simulators present in the literature. Among those providing an open-source implementation, we briefly describe here the two main ones, LoRaWANSim and a ns-3 `lorawan` module different from that employed in this thesis.

LoRaWANSim is emplyed for the analysis presented in [45], and is a network-layer extension of the Python-based LoRaSim simulator, which was previously limited to simulate the physical layer communication. LoRaWANSim differs from the simulator employed in this thesis in different aspects. However, in addition to the above mentioned GW modeling, The main drawback is the assumption of perfect orthogonality between SFs, neither it considers DC limitations. Furthermore, downlink traffic is not taken into account, just focusing on the UL performance. Finally, this implementation neglects the parallel reception capabilities of the GW.

A second LoRaWAN simulator is presented in [46, 47] proposing a LoRaWAN module for ns-3. In this implementation, the error model is derived from complex baseband Bit Error Rate (BER) simulations. Furthermore, we observe that this implementation models the GW reception capabilities in a different way, considering one receiver for each SF on each uplink frequency band, for a total of 18 parallel receivers (8 demodulators for 3 mandatory frequency channels in the EU configuration), each one capable of locking on one signal transmitted on a specific frequency with a specific SF. In this way, however, the GW will not be able to receive two or more signals sent with the same SF on the same frequency, irrespective of their relative received power, thus being more conservative than a real GW. On the contrary, the simulated GW may be able to successfully decode up to 18 signals, provided that the SFs of the signals in the same band are different, while a practical GW can only decode up to 8 signals.

These differences between the implementations turn out to be particularly significant when analyzing traffic scenarios with confirmed transmissions, as those that will be analyzed in the following of this thesis.

4 LoRaWAN in traditional IoT scenarios

LoRaWAN is gaining momentum thanks to its ease of deployment and the possibility of configuring several network parameters that affect different network performance indexes, such as energy efficiency, fairness, and capacity, thus, in principle, making it possible to adapt the network behavior to the specific requirements of the application scenario. Unfortunately, the complex and sometimes elusive interactions among the different network components make it rather difficult to predict the actual effect of a certain parameters setting, so that flexibility can turn into a stumbling block if not deeply understood. However, the deployment of a dense IoT network is expensive and time consuming, and performance assessments using simulations and mathematical models become essential to gauge the effect of network parameters and estimate the performance at a reduced cost. Also, these tools can be leveraged to explore possible improvements to the standard's default settings and protocol, as well as the join impact of the available parameters, so that to improve performance both in general use cases or in specific application settings.

In this chapter, we present both a mathematical model for LoRaWAN networks and a simulation framework, which is used in different contexts also in the rest of this thesis. The aim of this chapter is the evaluation of LoRaWAN performance in traditional IoT scenarios, focusing on open air or urban propagation environments. Therefore, the main objective is not assessing the feasibility of a specific application, but rather modeling and studying of the impact of the technology parameters, such as SF allocation, RX2 settings, number of transmissions on traditional networks metrics, like the Packet Delivery Rate (PDR).

In Sec. 4.1 we comment some previous studies on LoRaWAN performance evaluation, briefly describing experimental works and focusing on studies that propose analytical and simulation approaches.

Sec. 4.2 provides a mathematical model to estimate the performance of a LoRaWAN gateway serving a set of devices that may or may not employ confirmed traffic. The model features a set of parameters that can be adjusted to investigate different GW and ED configurations, making it possible to carry out a systematic analysis of various trade-offs. The results given by the proposed model are validated through realistic ns-3 simulations that confirm the ability of the model to predict the system performance with high accuracy, and assess the impact of the assumptions made in the model for tractability. Additionally, we show how the model can be leveraged to maximize different performance metrics.

Finally, in Sec. 4.3, we employ simulations to shed light on the complex interactions in a single-GW scenario. More specifically, we analyze the effect of some LoRaWAN built-in features and configurations, including the GW's limitations in terms of DC and number of parallel reception paths, and the number of allowed re-transmissions for confirmed traffic, and the pre-configured data rate used in downlink transmissions. The simulation-based analysis reveals various trade-offs and spotlights some inefficiencies in the design of the LoRaWAN standard. Furthermore, we show how significant performance gains can be obtained by wisely setting the system parameters, possibly in combination with some novel network management policies (e.g., enabling selective prioritization of downlink transmissions at the gateway).

4.1 State of the art

In the last years, the LoRaWAN technology has been the subject of many studies, which analyzed its performance and features with empirical measurements, mathematical analysis and simulative tools.

Among the works evaluating LoRaWAN through experimental campaigns, we remind [48, 49], where the authors test the coverage range and packet loss ratio, but without investigating the impact of the parameters setting on the performance. In [33], Bor et al. examine the impact of the modulation parameters on the single communication link between an ED and its GW, without considering more complex network configurations. Other works test LoRaWAN coverage in different propagation environments, such as cities [30, 50, 51], industrial settings [52], harbors [53] and rural settings [54, 55], or comparing the performance in different scenarios [56]. Other evaluations, instead, involve the analysis of the employed ADR algorithms, such as [57].

In the following subsections, we focus on previous literature works that analyzed LoRaWAN performance through mathematical modeling or simulation tools.

4.1.1 State of the art in LoRaWAN modeling

The work presented in [58] is one of the first to address the issue of scalability, using stochastic geometry to model interference in a LoRaWAN network. However, in this study, Georgiou et al., consider scenarios with only UL traffic. Similarly, Li et al. [59] use a stochastic geometry model to jointly analyze interference in the time and frequency domains in LPWANs, considering Sigfox and LoRaWAN technologies. It is observed that when implementing a packet repetition strategy, i.e., transmitting each message multiple times, the failure probability reduces, but, clearly, the average throughput decreases because of the introduced redundancy. The work in [60] adopts a Markovian approach to model the over-the-air activation procedure. In [61], Ferré et al. proposes closed-form expressions for collision and packet loss probabilities and, under the assumption of perfect orthogonality between SFs, it is shown that the Poisson distributed process does not accurately model packet collisions in LoRaWAN. Network throughput, latency and collision rate for uplink transmissions are analyzed in [62] using queueing theory and considering the Aloha channel access protocol and the regulatory constrains (i.e., DC) in the use of the different sub-bands. In this work, Sorensen et al. point out the importance of a clever splitting of the traffic in the available sub-bands to improve the network performance, for UL traffic only. In [63], Adelantado et al. address high-level questions about LoRaWAN's suitability for a range of smart city applications, from metering to video surveillance, by modeling the system as a superposition of different Aloha networks. They conclude that, even if the long coverage range of a single GW makes the infrastructure able to serve several devices, the network must be carefully dimensioned to meet the application requirements.

Croce et al., in [64], consider the features of the technology at the PHY layer, by focusing on the capture effect and imperfect orthogonality between SFs: after performing empirical measurements, they model these effects and derive the throughput achieved by the network for different cell configurations and number of GWs. In [65], the problem of network scalability is faced through mathematical modeling and Python-based simulations, taking into account also the capture effect, and evaluating the impact of SF allocation and power control. In these works, however, the main focus is on the PHY layer, and downlink traffic and re-transmissions are not considered.

In [66] Bankov et al. present a mathematical model of the network performance, taking into account factors such as the capture effect and a realistic distribution of SFs in the network. However, the model does not include some important network parameters, preventing the study of their effect on the network perfomance. This study is further improved in [67], where the authors provide a model based on Poisson arrival processes which takes DL communications, re-transmissions and capture effect into account. However, the analysis holds only in limited-size networks, where nodes can employ any transmission rate and their received powers are similar. A step further is made in [68] where Capuzzo et al. develop a model that makes it possible to consider various parameters configurations, such as the number of ACKs sent by the GW, the SF used for the downlink transmissions, and the DC constraints imposed by the regulations. In this work, mathematical modeling has been applied to assess the network performance with respect to various metrics, but multiple packet re-transmissions have not been considered.

Finally, in [69], Bouguera et al. focuse on the energy consumption of LoRa radio chipsets; in the work presented in [70], Khan et al. propose a model to calculate energy consumption and delay for reliable UL traffic in a LoRaWAN network. The results for a limited number of devices are compared to real test-bed measurements and to the outcome of ns-3 simulations. The analysis, based on Markov-chain theory, neglects the DC constraints in the different sub-bands, and assumes that ACKs are always sent in one specific receive window (either RX1 or RX2). Other mathematical models regarding LoRaWAN energy performance present in the literature are discussed in Sec. 6.

4.1.2 Related works on LoRaWAN simulations

In the study presented in [45], Pop et al. feature a system-level analysis of LoRaWAN with the LoRaSim simulator, and give significant insights on bottlenecks and network behavior in presence of downlink traffic. However, besides pointing out some flaws in the design of the LoRaWAN medium access scheme, this work does not propose any way to improve the performance of the technology.

In [46], system-level simulations in ns-3 in a multi-GW scenario are employed to assess the performance of confirmed and unconfirmed messages and show the detrimental impact of confirmation traffic on the overall network capacity and throughput in a smart city scenario. As expected, the availability of a multi-GW infrastructure brings several benefits to the network. In addition to the possibility of employing lower SFs in a dense deployment, a higher number of GWs makes it possible to share the load of downlink traffic, overcoming DC limitations and increasing the number of reply messages sent in RX1, which prevents sensor nodes from opening both receive windows. Here, the only proposed solution is the use of multiple gateways, without deeply investigating the specificities of the LoRaWAN standard. In [47] a module for the ns-3 simulator is proposed and used for a similar scope, comparing the single- and multi-gateway scenarios and the use of unconfirmed and confirmed messages. In this case, Reynders at al. correctly implement the GW's multiple reception paths, but do not take into account their association to a specific UL frequency, which usually occurs during network setup: indeed, the number of packets that can be received simultaneously on a given frequency can not be greater than the number of reception paths that are listening on that frequency. Also in this case, the study only focuses on the performance analysis, without proposing any improvement.

In [71] Kouvelas et al. show how the use of a persistent-Carrier Sense Multiple Access (p-CSMA) MAC protocol when transmitting UL messages can improve the packet reception ratio. However, attention must be payed to the fact that having many EDs that defer their transmission because of a low value of p may lead to channel under-utilization. In [72], Zucchetto et al. investigate, via simulation, the impact of DC restrictions in LPWAN scenarios, showing that rate adaptation capabilities are indeed pivotal to maintain reasonable level of performance when the coverage range and the cell load increase. However, the effect of other parameters setting on the network performance is not considered.

Varsier et al., in [73], instead, rely on a custom MATLAB simulator and tune the network parameters based on the requirements of an real metering application. Generally, these works identify DL traffic as a weak point in the LoRaWAN technology, because of either the strict DC limitations in Europe or its deleterious effect on UL traffic, and point to GW densification as a partial solution to the problem.

As we will see in the following discussion, the network will benefit from a wise SFs allocation, while an unwise setting of this parameter can easily be damaging for both the EDs's and network performance. Therefore, many works in the literature have investigated this aspect, i.e., how to optimal allocate SFs or how to design ADR algorithms that modify them, to improve the network capabilities. In [74] Cuomo et al. propose an ADR algorithm that assignes SFs by using a waterfilling algorithm, equalizing the aggregate time on air of the multiple SFs. This work is further extended in [75], with the proposal of two more algorithms for ADR. In particular, EXPLoRa-KM relieves critical regions, characterized by a significant number of collisions, and computes how to adjust the SF allocation leveraging the K-means algorithm. Conversely, the EXPLoRa-TS algorithm performs an equalization of the traffic load (measured in symbol times) among the SF channels, by considering the fact that each device, can transmit a variable amount of data at a different sending rate, according to the application. Thus, different traffic types (more or less aggressive) can be recognized by the NS, which can leverge this information to formulate a better policy for SF allocation, increasing the scalability of the network.

The authors in [76, 77] target the original ADR algorithm proposed by [36], suggesting possible ameliorations. Generally, the modified algorithms yield an increase of network scalability, fairness among nodes, packet delivery ratio and robustness to variable channel conditions. The proposals have been evaluated using Matlab and FLoRa, an open-source framework for end-to-end LoRa simulations in OMNeT++, respectively. In [78], Reynders et al. compute the optimal SFs distribution to minimize the collision probability and propose a scheme to improve the fairness for nodes far from the station by optimally assigning SFs and transmit power values to the network nodes, in order to reduce the packet error rate. Their proposal is validated using ns-3 simulations [47]. In [79] Benkahla et al. propose an ADR algorithm for LoRaWAN networks with mobility, which targets the minimization of transmission time and energy consumption, as well as packet loss. The SF to allocate is then chosen by estimating the future position of the node. The EARN algorithm proposed in [80], instead, also consider the CR and the capture effect to tune the link performance of the EDs, and evaluate it using SimPy, a discrete simulator on Python [81]. In [82], Lima et al. propose the APRA algorithm to optimize LoRaWAN's resources allocation, to improve scalability and energy performance by acting on multiple LoRaWAN transmission parameters, such as SF, power and bandwidth, evaluating its efficancy through the LoRaSim simulator.

4.2 A mathematical model for single-gateway LoRaWAN networks

The work presented in the first part of this section is an extension of our conference paper [68], where we modeled a wide network with Poisson packet arrivals, considering the DC limitations and a set of network parameters. Here we revise the model by developing a novel approach to accurately consider the limited availability of reception chains at the GW, the peculiarities of the two receive windows, and the DC constraints. Additionally, we include packet re-transmissions and the capture effect. Compared to previous models, our model takes into account the coexistence of unconfirmed and confirmed traffic, while offering the possibility of estimating the network behavior under several network configurations with minimal effort. The results obtained through this model are compared with those given by the LoRaWAN simulator, presented in Chapter 3, further attesting the accuracy of the proposed approach and exploring the impact of some common assumptions. Finally, we also show possible usages of the model to evaluate a wide variety of network configurations with limited effort. The results are obtained by considering the European default settings, previously described in Sections 2.3, 2.4, but our conclusions have broader interest, being valid also for other regions.

4.2.1 Packet life cycle

In LoRaWAN networks, messages transmitted by EDs to the GW are subject to multiple causes of losses, which affect their reception at the GW. Specifically, we can identify the following:

- *Interference*: packets sent in the same frequency channel and with the same SF collide. A transmission can survive a collision event if its received power is sufficiently higher than that of the other overlapping signals (capture effect).

- *GW already in transmission*: the GW can not lock on a UL packet while performing a DL transmission.

- *GW starting a transmission*: an ongoing packet reception may be interrupted if the GW needs to send a DL packet.

- *No available demodulation chains at the GW*: all demodulators are already busy decoding incoming signals.

Moreover, confirmed UL messages cause the NS to generate ACKs that need to be transmitted by the GW. Such DL transmissions may as well be impaired by a number of events:

- *Unavailability of receive windows*: this event occurs when all available GWs are prevented from transmitting in both the receive windows because of the DC constraint or other ongoing transmissions.

- *Interference*: DL packets transmitted in RX1 can collide with UL packets transmitted by other EDs in the same channel and with the same SF.

The network model provided in the following accounts for all these events.

4.2.2 Model formulation

The aim of the model is to characterize the behavior of a LoRaWAN network with a single GW, which receives packets from a set of EDs and needs to reply in one of the two receive windows when an ED requires confirmation. The system performance is assessed in terms of packet success probability, following the approach used in [68] and extending it with a more accurate characterization of the GW behavior. This performance metric is proxy to other fundamental metrics, such as throughput and network capacity, which can be straightforwardly derived from it. The following sub-sections are structured as follows. The reference scenario, model assumptions, system parameters and their effects are described in Sec. 4.2.3, together with a brief presentation of the structure of the model and its underlying rationale; Sec. 4.2.4, then, describes some relevant quantities and parameters of the proposed model. We then delve into the analytical formulation by decoupling the analysis of the UL traffic and DL messages, and derive the formulas for DL success probabilities. Finally, we describe different performance metrics and their computation. Note that, because of the mutual dependency of some values, some terms may be described and introduced before the corresponding equation can be derived, in which case references are provided in the text.

4.2.3 Scenario and assumptions

We consider a scenario where the EDs are randomly and uniformly distributed around a single GW. Application-layer packets are generated according to a Poisson Process with aggregate packet generation rate λ [pck/s], and can be either confirmed or unconfirmed.

For tractability, we assume perfect orthogonality between different SFs, i.e., only packets employing the same SF can collide. In this case, one of the two

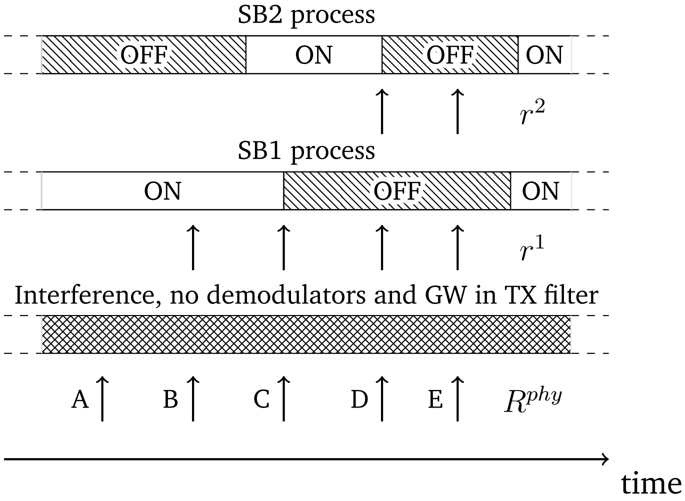


Figure 4.1.: Representation of the model's packet filtering structure. R^{phy} is the rate of UL traffic (see (4.4)), while r^1 and r^2 represent the rate of ACKs sent in SB1 and SB2, respectively (see (4.16), (4.18)).

packets can survive if its received power is sufficiently higher than that of the colliding packet (collisions with more than two packets happen with negligible probability and are not considered). While the orthogonality assumption has been shown to have an impact on the PHY-layer performance of UL only traffic [32], the results discussed in Sec. 4.2.10 show that the effect is much more limited in the presence of confirmed traffic, where the performance is severely limited by other factors.

Fig. 4.1 shows the structure of the packet reception model, consisting in successive filtering of Poisson processes. At the base of the figure, arrows are used to represent the UL traffic generated by the EDs, including both new packet transmissions and re-transmissions of failed packets. This process is assumed to be Poisson for tractability, ignoring the fact that re-transmissions of a certain packet are correlated in time because of DC limitations. An initial filtering of this process excludes some arrivals, modeling packet losses due to interference from other EDs, unavailability of GW demodulators, or ongoing DL transmissions from the GW. This yields a process with a reduced rate, which now represents the packets that are correctly received by the GW.

When the received UL message requires confirmation, an ACK must be sent by the GW during one of the two receive windows of the target ED. The ability of the GW to perform such a transmission is modeled through two independent alternating renewal processes, in which the system alternates between the ON and OFF states. The two processes represent the opportunity of sending the ACK in RX1 or RX2, respectively, which are opened on SB1 or SB2, i.e., on the shared or dedicated sub-band. If a confirmed packet finds a process in the ON state, it means that the GW will be able to send an ACK in that sub-band. In this case, the process will switch to the OFF state to model the unavailability of that sub-channel for a certain period of time following the ACK transmission, due to the DC restrictions.

Since the sub-bands are disjoint, we assume that the two processes are uncorrelated, neglecting the fact that the very packets that need to be served in SB2 are those that found SB1 in the OFF state. If the DL packet finds at least one of the two processes in the ON state, an ACK is sent. If the ACK is sent on SB1 (hence, using frequencies shared by UL and DL traffic), it can be destroyed by the interference created by other EDs. If the ACK is sent on SB2, instead, it is assumed to be always successful.

For the sake of clarity, the following list describes some examples of the life cycle of the packets in Fig. 4.1:

(A) This packet is lost because of interference or GW transmission or unavailability of demodulators. Hence, it does not pass the first filter.

(B) This is an unconfirmed UL packet, which is successfully received by the GW. It does not generate any ACK.

(C) This is a confirmed packet successfully received by the GW. It generates an ACK, which finds the SB1 process in the ON state. The ACK is successfully sent, and the SB1 process switches to the OFF state.

(D) This is another confirmed packet which is successfully received by the GW. Since the GW has just sent an ACK for packet (C), it cannot reply in SB1 due to DC constraints; SB2 is however in the ON state, and the GW can thus reply to the ED, making the second process switch to the OFF state.

(E) This is another confirmed packet, which gets a treatment similar to that of packet (D). However, since the GW has transmitted the ACK for packet (D) and is still under the DC constraints, it cannot reply to packet (E) in either of the two receive windows (both SB1 and SB2 processes are in the OFF state). The DL packet is hence discarded, and the ED will re-transmit the UL message at a later time.

4.2.4 Tunable parameters

Our model offers some tunable parameters to increase its flexibility, enabling the evaluation of the network performance in various configurations with minimal effort. The model makes it possible to specify the following values:

- $\mathcal{SF} = \{7, \ldots, 12\}$ indicates the set of all SFs.
- α: fraction of application-layer traffic requiring confirmation;
- p_i^u, p_i^c: fraction of devices generating unconfirmed and confirmed traffic with a specific SF $i \in \mathcal{SF}$, respectively. Note that

$$\sum_{i \in \mathcal{SF}} p_i^u = \sum_{i \in \mathcal{SF}} p_i^c = 1;$$

- h: number of times an application-layer unconfirmed packet is transmitted;
- m: maximum number of transmission attempts for confirmed packets;
- δ_{SB1} and δ_{SB2}: ratio between silent time and transmission time in SBk, corresponding to the DC constraint. For instance, in Europe, we have $\delta_{SB1} = 99$ and $\delta_{SB2} = 9$ corresponding to a DC of 1% in SB1 and 10% in SB2. In general, when $\delta_{SBk} > 0$ the DC constraint applies to all devices transmitting in subchannel SBk. Instead, the setting $\delta_{SB_k} = 0$ corresponds to a DC constraint of 100%, which means that there is no limitation on the transmission time[1];
- τ_1 and τ_2: prioritization flags. If $\tau_k = 0$, the GW prioritizes reception operations over transmission during the k-th receive window, with $k = 1, 2$. In this case, the GW will drop any DL message that needs to be transmitted while a UL reception is ongoing. Instead, if TX is prioritized ($\tau_k = 1$), the reception of any incoming packet will be interrupted in order to send the ACK;
- C: number of UL frequency channels. Note that each UL channel can also be used for DL transmissions. Instead, the channel in SB2 is DL only;
- $T_i^{ack_2}$: duration of the transmission of the ACK in RX2 when using SF i. (The standard requires the use of SF 12 in RX2 as a pre-configured

[1] This setting is not allowed by current RF recommendations but is considered in this study to gain insights on the impact of DC limitations in the considered scenarios.

setting, corresponding to $T_{12}^{ack_2}$. Note that this default setting can be changed by the NS, and accordingly in our model.)

- T_i^{data} and $T_i^{ack_1}$ indicate the time durations of a data packet and of an ACK transmitted in SB1 with SF i, respectively. If ACKs transmitted in SB2 use SF12, irrespective of the SF employed in the UL transmission, then $T_i^{ack_2} = T_{12}^{ack_1}$, $\forall i \in \mathcal{SF}$.

In the formulas, the notation generally respects the following scheme. The probability is indicated with S or F if it corresponds to a "success" or "failure" event, respectively; if this rule does not apply, the probability is denoted simply as P. The superscript indicates the considered event, while the subscript the SF. For example, in (4.6), the symbol S_i^{INT} represents the probability of successfully surviving interference (INT) when using SF i. Different uses of the notation are specified in the text. The following sections provide a mathematical formulation for some relevant quantities in this model.

4.2.5 Uplink traffic rates

The assumption of perfect orthogonality between different SFs makes it possible to split the network traffic in different logical channels that do not interfere with each other. The traffic load for each SF i is split uniformly over the given C frequency channels (since EDs pick a random UL frequency for each transmission attempt). Thus, the traffic generated at the application layer by the EDs using confirmed and unconfirmed messages is, respectively, given by

$$R_i^{c,app} = \frac{p_i^c \cdot \lambda}{C} \cdot \alpha, \tag{4.1}$$

$$R_i^{u,app} = \frac{p_i^u \cdot \lambda}{C} \cdot (1 - \alpha). \tag{4.2}$$

Since EDs using unconfirmed traffic will perform h transmissions of each application-layer packet, the PHY rate of these devices can be computed as $R_i^{u,phy} = R_i^{u,app} \cdot h$. For EDs transmitting confirmed messages, instead, the number of re-transmitted packets depends on the success of both the UL transmission and the corresponding ACK. We indicate as $P_{i,j}^{DL}$ the probability that a confirmed UL packet sent with SF i is successfully received and acknowledged at the j-th transmission attempt, which will be derived in (4.29). Therefore, we have that the rate of confirmed packets transmitted at SF i, $R_i^{c,phy}$, is given by the product of the application-level rate, $R_i^{c,app}$, and the average number of times a confirmed packet is transmitted at the PHY layer.

$$R_i^{c,phy} = R_i^{c,app} \left[\sum_{j=1}^{m-1} j \cdot P_{i,j}^{DL} + m \left(1 - \sum_{j=1}^{m-1} P_{i,j}^{DL} \right) \right]. \tag{4.3}$$

The first summation in the square brackets of (4.3) takes into account transmissions that are successfully received before the mth attempt, while the second term considers the case when the packet is transmitted m times (irrespective of whether the last transmission is successful or not).

The total traffic for a single frequency channel and for SF i is therefore given by

$$R_i^{phy} = R_i^{u,phy} + R_i^{c,phy}. \tag{4.4}$$

In general, the distribution of the SFs for the transmitted packets at the PHY layer will differ from the native distribution of SFs among the devices, $\{p_i^u, p_i^c\}$, because of re-transmissions. Thus, we define

$$d_i = \frac{R_i^{phy}}{\sum_j R_j^{phy}}, \tag{4.5}$$

as the ratio of PHY layer packets that are transmitted at SF $i \in \mathcal{SF}$.

4.2.6 PHY layer probabilities

A UL packet is successfully received by the GW if all the following conditions are met: (i) it does not overlap with another UL transmission using the same SF on the same frequency, or it overlaps with another UL packet, but the received power is sufficiently large to allow for correct decoding despite the interference (capture), (ii) it does not overlap with a GW DL transmission in any channel, and (iii) it finds an available demodulator. These conditions are represented by the first filter in Fig. 4.1.

Since packets are generated following a Poisson process, the probability of event (i) is given by two components. The first is the probability that there are no other arrivals during the $2T_i^{data}$ vulnerability period across the packet arrival instant. The second, considers a collision with one packet, and the fact that the receiver successfully captures the frame. For the UL, we consider the capture probability $\mathbb{W}^{GW}$ as computed in [66]. Since these two events are disjoint, the probability of surviving interference (event (i)) is given by the sum of the two components, which results in

$$S_i^{INT} = e^{-2T_i^{data} R_i^{phy}} + 2T_i^{data} R_i^{phy} e^{-2T_i^{data} R_i^{phy}} \cdot \mathbb{W}^{GW}, \tag{4.6}$$

where, in the right-most term, we computed the probability that either of the two colliding packets is captured (collision events with more than two packets are neglected).

To compute the probability of event (ii), we observe that a UL message is always lost when it arrives at the GW during the transmission of an ACK. Otherwise, the GW will start the reception of the UL message, which will take a time T_i^{data}. If reception on SBk is prioritized (i.e., $\tau_k = 0$), this process cannot be interrupted, and the UL message will be successfully delivered to the NS. Conversely, if $\tau_k = 1$, i.e., we prioritize transmission on SBk, the reception of the UL packet may be aborted at any time during the period T_i^{data}, in order to give priority on the ACK transmission. Therefore, the vulnerability period is given by the ACK transmission time $T_s^{ack_k}$, to which we need to add the interval T_i^{data} only if $\tau_k = 1$. Denoting by b_s^k the probability that an ACK is transmitted on SBk with SF $s \in \mathcal{SF}$ (which will be derived later in (4.22)), the average vulnerability period is then given by $\overline{T_k} = \sum_{s \in \mathcal{SF}} b_s^k T_s^{ack_k} + T_i^{data} \cdot \tau_k$. Now, according to the Poisson Arrivals See Time Averages (PASTA) property, the probability that a UL packet arrival falls in the vulnerability period of channel SBk, with $k = 1, 2$, can be expressed as

$$F_i^{TXk} = \frac{\sum_{s \in \mathcal{SF}} b_s^k T_s^{ack_k} + T_i^{data} \cdot \tau_k}{E_{ON}^k + E_{OFF}^k}, \tag{4.7}$$

where the denominator is the mean renewal time of the SBk process, given by the sum of E_{ON}^k and E_{OFF}^k, i.e., the expected times the SBk process spends in the ON and OFF states during a renewal period (ON-OFF cycle), which will be computed in (4.21) and (4.23). Then, assuming (for ease of analysis) that events in SB1 and SB2 are independent, the probability that a UL packet is successfully received (event (ii)) is given by

$$S_i^{TX} = (1 - F_i^{TX1})(1 - F_i^{TX2}). \tag{4.8}$$

Next, we compute the probability of event (iii), i.e., that at least one demodulator out of 8 is available. Each demodulator chain is modeled through an alternating renewal process, where the demodulator can be in an “available” state A, when idle or in a “locked” state L, when occupied with the reception of another signal. We assume that the different demodulators are activated in succession: if all are available, an incoming signal will be received by the first demodulator; if the first demodulator is in the L state, the packet will be handled by the second demodulator, and so on. Let E^L be the expected time a demodulator will be locked on a incoming signal. Since the occupation will last for the duration of UL LoRa packets at the PHY layer, we have:

$$E^L = \sum_{i \in \mathcal{SF}} d_i \cdot T_i^{data}. \tag{4.9}$$

The average time the first demodulator is in the A state, instead, is computed as the average inter-arrival time of UL packets, regardless of their SF and selected frequency:

$$E^{A,1} = \frac{1}{C \cdot \sum_{i \in \mathcal{SF}} R_i^{phy}}. \tag{4.10}$$

Then, the process of packets that require the second demodulator is filtered by the probability of finding the first demodulator occupied. Thus, the expected time the second demodulator is available is given by

$$E^{A,2} = \frac{E^{A,1}}{P^{L,1}} = \frac{1}{P^{L,1} \cdot C \cdot \sum_{i \in \mathcal{SF}} R_i^{phy}}, \tag{4.11}$$

where $P^{L,1}$ is the probability that the first modulator is in the L state (see (4.13)). With a similar reasoning, we compute the expected time for which the j-th demodulator is available as

$$E^{A,j} = \frac{E^{A,j-1}}{P^{L,j-1}} = \frac{1}{\prod_{\ell=1}^{j-1} P^{L,\ell} \cdot C \cdot \sum_{i \in \mathcal{SF}} R_i^{phy}}. \tag{4.12}$$

The probability $P^{L,\ell}$ of finding the ℓ-th demodulator in the L state, in turn, can be expressed as

$$P^{L,\ell} = \frac{E^L}{E^{A,\ell} + E^L}. \tag{4.13}$$

Then, a packet finds an available demodulator (event (iii)) with probability:

$$S^{demod} = 1 - \prod_{j=1}^{8} P^{L,j}. \tag{4.14}$$

The overall UL packet success probability, considering events (i), (ii) and (iii) described above, is finally expressed as

$$S_i^{UL} = S_i^{INT} \cdot S_i^{TX} \cdot S^{demod}. \tag{4.15}$$

4.2.7 ACK transmission

Once a confirmed packet is correctly received by the GW, an ACK needs to be transmitted back to the ED. Eq. (4.15) gives the probability of successful

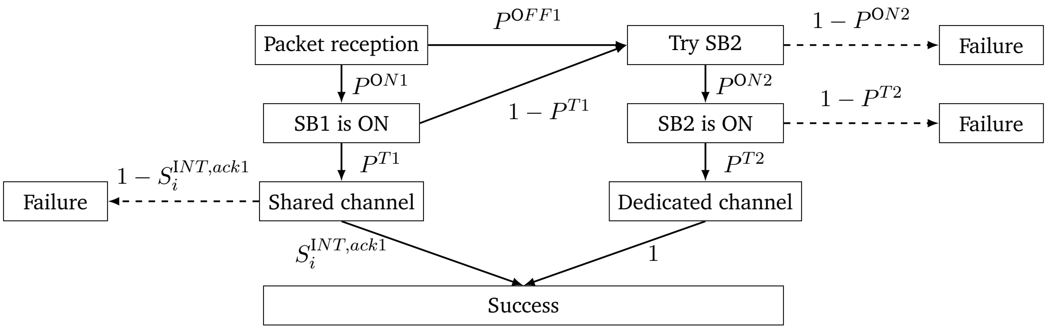


Figure 4.2.: Diagram for successful ACK reception.

packet reception at the GW. Therefore, the rate of ACK messages that the GW will try to send in SB1 is:

$$r_i^1 = R_i^{c,phy} \cdot S_i^{UL}. \tag{4.16}$$

A visual representation of the possible ACK life cycles considered in the model is shown in Fig. 4.2. Labels refer to the probabilities of the different events, which we derive next. In general, an ACK is transmitted in SBk if both the following conditions hold: (i) $\tau_k = 1$ (TX is prioritized) or $\tau_k = 0$ and the GW is idle; (ii) SBk is available (i.e., not blocked by DC constraints). If either condition is not satisfied, the ACK is dropped.

Let T denote the event "the GW *may* transmit," which depends on the TX/RX prioritization policy. If $\tau_k = 1$, the GW can transmit the DL packet whenever it needs to; otherwise, if $\tau_k = 0$, the GW can transmit in SBk only if no reception is ongoing. We denote by $P^{T,k}$ the probability of T, which can be computed as

$$P^{T,k} = \begin{cases} 1, & \text{if } \tau_k = 1; \\ e^{-\sum_{i \in \mathcal{SF}} C \cdot R_i^{phy} T_i^{data}}, & \text{if } \tau_k = 0; \end{cases} \tag{4.17}$$

where the second expression is the probability that no UL packet was generated in the last T_i^{data} seconds.

If SB1 is not available, the GW will try to process the ACK in SB2. Such packets form a process with rate

$$r_i^2 = r_i^1[P^{OFF,1} + P^{ON,1}(1 - P^{T,1})], \tag{4.18}$$

where $P^{ON,1}$ and $P^{OFF,1}$ are the probabilities of finding SB1 in the ON and OFF state, respectively, and $(1 - P^{T,1})$ is the probability that the GW is not available for DL transmission. The ON and OFF probabilities for the SBk process, with $k = 1, 2$, are given by

$$P^{ON,k} = \frac{E^{ON,k}}{E^{ON,k} + E^{OFF,k}}, \tag{4.19}$$

$$P^{OFF,k} = \frac{E^{OFF,k}}{E^{ON,k} + E^{OFF,k}}, \tag{4.20}$$

where $E^{ON,k}$ and $E^{OFF,k}$ are the mean sojourn times in ON and OFF states, respectively, which are computed as follows. By considering the arrival rate of successful UL packets in the k-th sub-band, we have:

$$E^{\text{ON}k} = \frac{1}{\sum_{i \in \mathcal{SF}} C \cdot r_i^k}. \tag{4.21}$$

Note that the switch from the ON to the OFF state will be caused by a packet sent in any of the C UL channels: therefore, we need to multiply the rates r_i^k of arrivals to SBk with SF i by the number of available channels.

In order to compute the expected duration of the OFF periods, we first need to derive the probability distribution b_i^k of the SFs used for ACK transmissions, which is given by

$$b_i^k = \frac{r_i^k}{\sum\limits_{s \in \mathcal{SF}} r_s^k}. \tag{4.22}$$

In our model, the OFF period accounts for the time the GW is prevented from transmitting a new data packet, which includes the time to send the ACK using the given SF, plus the waiting time imposed by the DC limitations. We hence have

$$\begin{aligned} E^{\text{OFF},1} &= \sum_{s \in \mathcal{SF}} b_s^1 (T_s^{\text{ack}_1} + \delta_{SB1} \cdot T_s^{\text{ack}_1}), \\ E^{\text{OFF},2} &= \sum_{s \in \mathcal{SF}} b_s^2 (T_s^{\text{ack}_2} + \delta_{SB2} \cdot T_s^{\text{ack}_2}). \end{aligned} \tag{4.23}$$

(Note that, by including the parameter δ_{SBk} as defined in Sec. 4.3.2, we can change the DC limitations in the k-th sub-band, thus making it possible to analyze its impact.)

Finally, we remark that DL packets sent by the GW in SB1 also have to avoid interference from other EDs. In the absence of collisions, the vulnerability period is given by the sum of two terms. The first term corresponds to the case of no UL transmissions starting while the DL packet is being sent (T^{ack_1}); the second term represents the event where no UL transmissions started before the ACK is sent. Note that if $\tau_1 = 0$ the second term is not present, since in that case the ACK would not be generated at all. Furthermore, an ACK can survive an interfering packet sent by another ED in case of capture, which happens with probability $\mathbb{W}^{ED}$ (equivalent to the $\mathbb{W}^{Mote}$ as derived in [66]). Therefore, the probability that the ACK does not collide with a UL packet in SB1, or is captured despite the collision, is equal to

$$S_i^{INT,ack_1} = e^{-R_i^{phy}(T_i^{ack_1}+\tau_1 \cdot T_i^{data})} + R_i^{phy}(T_i^{ack_1} + T_i^{data}) \cdot e^{-R_i^{phy}(T_i^{ack_1}+T_i^{data})} \cdot \mathbb{W}^{ED}. \tag{4.24}$$

For packets sent in SB2, instead, the reception is assumed to be always successful, since the 869.525 MHz channel is dedicated to DL communication and the GW only transmits one packet at a time (note that this assumption does not hold in the case of multiple GWs).

4.2.8 DL success probability

Given that a confirmed UL packet sent with SF i has been successfully received by the GW, the probability that the corresponding ACK is also successfully returned to the ED is expressed as

$$S_i^{\mathrm{DL}} = S_i^{\mathrm{SB1}} + S^{\mathrm{SB2}}, \tag{4.25}$$

where S_i^{SB1} describes the probability of a successful ACK transmission in SB1 with SF i, while S^{SB2} accounts for the probability that SB1 is not available, and the ACK is successfully sent in SB2. These probabilities, in turn, can be expressed as follows:

$$S_i^{\mathrm{SB1}} = P^{ON,1} \cdot P^{T,1} \cdot S_i^{INT,ack_1}, \tag{4.26}$$

$$S^{\mathrm{SB2}} = [P^{OFF,1} + P^{ON,1} \cdot (1 - P^{T,1})] \cdot P^{ON,2} \cdot P^{T,2}. \tag{4.27}$$

Fig. 4.2 can be used as a reference for the computation of this quantity.

Finally, we can compute the success probabilities over m transmissions. We recall that, for the sake of simplicity, we neglect the time correlation of packet re-transmissions due to DC constraints, (the impact of this approximation will be analyzed by simulation). We recall that $P_{i,j}^{UL}$ indicates the probability that a UL packet with SF i is successfully received at the GW at exactly the j-th transmission attempt, which can be computed as:

$$P_{i,j}^{UL} = S_i^{UL}\left(1 - S_i^{UL}\right)^{j-1}. \tag{4.28}$$

Then, the ED successfully receives the ACK at exactly the j-th attempt if both the UL and the DL transmissions succeed. The probability $P_{i,j}^{DL}$ of this event is hence given by:

$$P_{i,j}^{DL} = \left[1 - (S_i^{UL} S_i^{DL})\right]^{j-1} \cdot (S_i^{UL} S_i^{DL}). \tag{4.29}$$

Once all intermediate quantities are computed, the model can be summarized by two inter-dependent equations:

$$\begin{cases} S^{UL} = f(S^{UL}, S^{DL}), \\ S^{DL} = g(S^{UL}, S^{DL}). \end{cases}$$

where $S^{UL} = [S_7^{UL}, \dots, S_{12}^{UL}]$ and $S^{DL} = [S_7^{DL}, \dots, S_{12}^{DL}]$, while $f()$ and $g()$ are implicit functions given by the chaining of the sequence of operations that yield (4.15) and (4.25), respectively.

This system admits a fixed-point solution, which can be found through fixed-point iteration. From a practical perspective, when initialized with the states $S^{UL} = S^{DL} = [1, 1, 1, 1, 1, 1]$, the iterative process has always reached convergence to the stable fixed point after a few iterations (order of few units) for all the parameter combinations considered in this work. The proof of the system's convergence is provided in Appendix A. An implementation of the model, allowing the interested readers to easily replicate the results shown in this section, is publicly available at [83].

Performance metrics

To evaluate the system performance, we consider three classes of key performance indicators, namely: reliability, delay, and fairness metrics, which are better detailed in the remainder of this section together with the methodology to determine their value using the proposed model. Once a set of parameters is fixed, the model can be solved and the performance metrics can be estimated starting from S^{UL} and S^{DL}. Conversely, it is possible to employ the model to optimize a given performance metric, finding the parameter setting that maximizes it, as shown in Sec. 4.2.10.

Reliability Metrics We consider three PDR indexes, namely:

- *Unconfirmed Uplink PDR (UU)*: fraction of (application-layer) unconfirmed packets that are successfully received by the GW;
- *Confirmed Uplink PDR (CU)*: fraction of (application-layer) confirmed packets that are successfully received by the GW, irrespective

of whether or not the corresponding ACK is successfully returned to the ED;

- *CD*: fraction of (application-layer) confirmed packets that are successfully acknowledged by the NS.

Clearly, CD $\leq$ CU, since a packet needs to be successfully received by the GW in order to be acknowledged. Note that the CU metric captures the performance of applications for which it is important to deliver packets to the NS and ACKs are only used to stop re-transmissions (and thus avoid a useless increase in traffic), while CD is more interesting for applications that require the EDs to get explicit feedback from the NS, for instance containing control information addressed to the ED.

We obtain the UU and CU values by averaging the UL success probability (UU_i and CU_i for unconfirmed and confirmed packets, respectively) for each SF i over the SF distribution, i.e.,

$$\mathrm{UU} = \sum_{i \in \mathcal{SF}} (p_i^u \cdot \mathrm{UU}_i) = \sum_{i \in \mathcal{SF}} \left(p_i^u \cdot \sum_{j=1}^{h} P_{i,j}^{UL} \right), \tag{4.30}$$

$$\mathrm{CU} = \sum_{i \in \mathcal{SF}} (p_i^c \cdot \mathrm{CU}_i) = \sum_{i \in \mathcal{SF}} \left(p_i^c \cdot \sum_{j=1}^{m} P_{i,j}^{UL} \right). \tag{4.31}$$

Similarly, CD is computed as the probability of success for a confirmed packet within the available re-transmission attempts

$$\mathrm{CD} = \sum_{i \in \mathcal{SF}} \left(p_i^c \cdot \sum_{j=1}^{m} P_{i,j}^{DL} \right). \tag{4.32}$$

Delay Metrics We define two delay metrics, considering confirmed traffic only: Δ^{UL} measures the time from the first transmission attempt to the successful delivery to the GW of an UL confirmed packet, while Δ^{DL} accounts for the time from the first transmission of a confirmed packet to the successful reception of the corresponding reply. Delays are computed for successful packets only, and the propagation delay is assumed to be negligible. To compute these metrics with our model, we assume the RETRANSMIT_TIMEOUT value to be a uniformly distributed random variable with mean μ, and consider that EDs employ the shared sub-band with δ_{SB1} DC limitations. Therefore, the average time between two transmissions of the same MAC-layer packet by a device is given by:

$$\gamma_i = (\delta_{SB1} + 1) \cdot T_i^{data} + \mu. \tag{4.33}$$

The average delay from the successful reception of a packet at the GW to the transmission of the ACK is given by:

$$\phi_i = S_i^{\mathrm{SB1}} \cdot (1 + T_i^{ack_1}) + S^{\mathrm{SB2}} \cdot (2 + T_i^{ack_2}), \tag{4.34}$$

where we take into account that the ACK will be served in SB1 (opened after 1 second) with probability S_i^{SB1}, and in SB2 (opened after 2 seconds) with probability S^{SB2}.

If a packet is re-transmitted m times, each re-transmission j is associated with a certain UL success probability $P_{i,j}^{\mathrm{UL}}$. The average delay at each SF $i \in \mathcal{SF}$ can be computed as:

$$\Delta^{\mathrm{UL}} = \sum_{i \in \mathcal{SF}} p_i^c \cdot \left(\sum_{j=1}^{m} \bar{P}_{i,j}^{\mathrm{UL}} \left(T_i^{data} + (j-1) \cdot \gamma_i \right) \right), \tag{4.35}$$

where we define $\bar{P}_{i,j}^{\mathrm{UL}} = P_{i,j}^{\mathrm{UL}} / \sum_j P_{i,j}^{\mathrm{UL}}$ to obtain the distribution of successful UL packet transmissions.

Similarly, we can compute the average ACK delay:

$$\Delta^{\mathrm{DL}} = \sum_{i \in \mathcal{SF}} p_i^c \cdot \left(\sum_{j=1}^{m} \bar{P}_{i,j}^{\mathrm{DL}} \left(T_i^{data} + (j-1) \cdot \gamma_i + j \cdot \phi_i \right) \right), \tag{4.36}$$

where, in addition to the inter-transmission time between two packets, we also account for the time to perform the ACK transmission.

Fairness Finally, we consider the fairness of the system in different scenarios. Indeed, EDs employing confirmed traffic or higher SFs will use more system resources (e.g., channel occupancy), possibly affecting the application performance of devices that employ different settings. To this aim, we use Jain's fairness index, defined as

$$J(\mathbf{x}) = \frac{\left(\sum_{i=1}^{n} x_i \right)^2}{n \cdot \sum_{i=1}^{n} x_i^2}, \tag{4.37}$$

where n is the total number of user categories, each with throughput x_i. Note that $1/n \leq J(\mathbf{x}) \leq 1$, and the system is perfectly fair if $J(\mathbf{x}) = 1$. In particular, in the following section, we will consider the fairness among devices employing different SFs. Furthermore, since all the devices have equal packet generation rate, and transmit packets with the same length, instead of the throughput we can simply consider the UL success probability, i.e., UU for nodes employing unconfirmed traffic and CU for devices transmitting confirmed messages. Therefore, the fairness is computed by taking $\mathbf{x} = [\mathbf{x^u}, \mathbf{x^c}]$, where the elements correspond to $x_i^u = UU_i$, and $x_i^c = CU_i$, as defined in (4.30), (4.31).

4.2.9 Network simulations

In order to validate our model, we compared the performance estimates obtained from the model with those observed in more realistic simulations, in which most of the simplifying assumptions of the model are removed.

This section describes how we employed the LoRaWAN ns-3 module described in [84] to perform such a validation. To be noted that the more accurate modeling of the LoRaWAN standard considered in the simulator comes at the cost of a much larger computational time to assess the system performance. Indeed, for the same parameter set, the performance evaluation is basically instantaneous when employing the theoretical model, while each ns-3 simulation run takes in the order of tens of seconds, with execution times rapidly increasing when the traffic load, the number of devices or the number of required randomized runs grow.

The merit of the simulator is that it strives to be as realistic as possible, also taking into account some factors that are overlooked by the model for tractability reasons. For instance, the assumption of perfect orthogonality between transmissions employing different SFs is removed, and the simulator relies on the link-level model provided in [15] to determine the actual reception probability in case of overlapping transmissions, which also accounts for the capture effect.

The simulation setting is as follows.

- *Traffic load* – The number of EDs is fixed to 1200, and the EDs' application layer is set to periodically generate packets to be transmitted by the MAC layer. The traffic load in the network is modified by varying the packet generation period. It is to be noted that this periodic traffic generation pattern is likely more realistic than the Poisson traffic assumed in the

Table 4.1.: Values of T^{data}, T^{ack} and SF distributions p. Payload of data packets is 10 bytes; ACKs have no payload.

SF	T^{data} [s]	T^{ack} [s]	p_{equal}	p_{EXPLoRa}
7	0.051	0.041	0.166	0.487
8	0.102	0.072	0.166	0.243
9	0.185	0.144	0.166	0.135
10	0.329	0.247	0.166	0.076
11	0.659	0.495	0.166	0.038
12	1.318	0.991	0.166	0.019

model. Nonetheless, the good match of simulation and analytical results confirms that the Poisson assumption is valid when the number of nodes is sufficiently large.

- *Channel allocation* – We consider the typical frequency allocation scheme for Europe, as reported in Tab. 2.7. Therefore, the number of different frequency channels for UL is $C = 3$.

- *Duty cycle* – The simulator considers the DC limitations applied in the European region [37], which corresponds to setting $\delta_{SB1} = 99$ and $\delta_{SB2} = 9$ in the model.

- *Channel model* – Differently from the model, simulated LoRaWAN nodes experience a log-distance propagation path loss, as for an open-air scenario. Thus, farther devices will suffer increased loss, and their performance will be penalized with respect to EDs that are close to the GW. Note that we do not include fast-fading components, which are supposed to be averaged out by the LoRa modulation, nor time-dependent variations in the channel, which remains constant throughout the entire simulation. Also, the channel is assumed to be symmetric, and DL transmissions will suffer the same impairments as in the UL.

- *SF distribution* – EDs are located around the single GW in a circular area of radius 2500 m, which allows for communications with any SFs with negligible channel error probability (in the absence of interference). Instead, the positions of the nodes are randomly picked at each simulation run. SFs are assigned uniformly (see Tab. 4.1, p_{equal}). A different SF distribution (p_{EXPLoRa}) is considered in some scenarios, to evaluate the impact of this parameter on the different metrics.

- *Interference and capture effect* – To model interference, in the simulator we consider the collision matrix provided in [15] and the overlapping time between packets, as described in [39].[2] A packet survives interference from a signal modulated with the same SF if its power is at least $CR_{dB} = 6$ dB higher than the colliding one. In order to provide a comparison with this scenario, in the analytical model we leverage the assumption of uniformly distributed EDs around the GW to compute the capture probabilities as in [66], which results in $\mathbb{W}^{GW} = 0.1796$, and $\mathbb{W}^{ED} = 0.5682$. We remark that different distributions of EDs around the GW can be modeled by adapting this derivation.

Since the GW implementation in the simulator attempts to emulate the behavior of a real device, a UL packet is successfully received when all the following conditions are satisfied:

1. The packet finds an available demodulator;
2. The packet's reception is not interrupted by DL transmissions;
3. Once the reception is finished, the packet was not corrupted by interference.

To count packets at the PHY layer coherently with the simulator implementation, the model's packet loss probabilities due to lack of demodulators (F_{NMD}), GW transmission (F_{GWTX}) and interference (F_{INT}) are plotted in the following section using, respectively, the following expressions:

1. $F_{\text{NMD}} = 1 - S^{demod}$;
2. $F_{\text{GWTX}} = E_i \left[S^{demod} \cdot (1 - S_i^{TX}) \right]$;
3. $F_{\text{INT}} = E_i \left[S^{demod} \cdot S_i^{TX} \cdot (1 - S_i^{INT}) \right]$;

by exploiting (4.6), (4.8), and (4.14), and where $E_i \left[\cdot \right]$ indicates the expectation over the distribution of SFs and S^{demod} the probability that, in the simulations, a packet can lock on an available demodulator.

[2] Note that, in the simulator, the capture event is determined also considering the partial overlapping of the colliding packets.

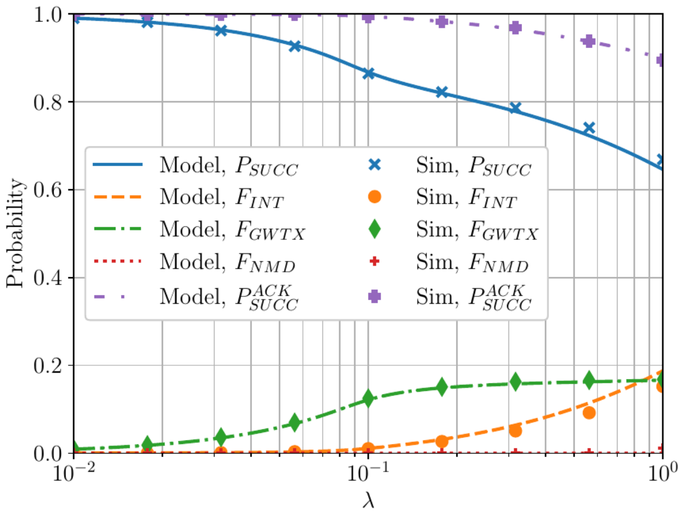


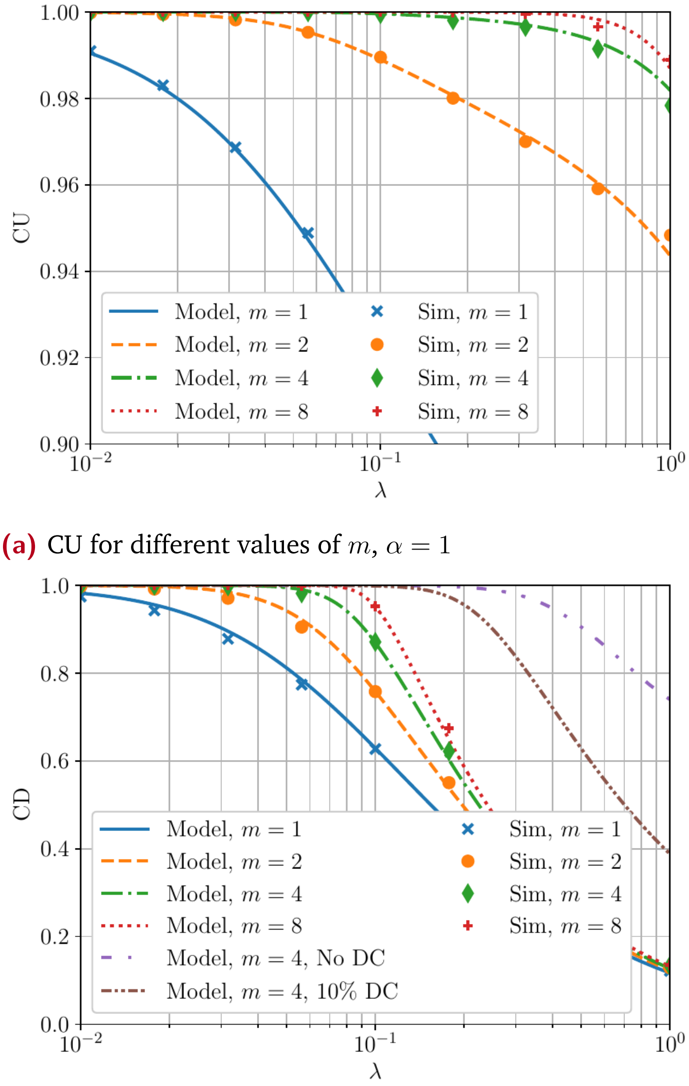
Figure 4.3.: PHY-level performance with $m = 8$, $\alpha = 1$.

4.2.10 Results

This section provides a comparison between the performance estimated with the proposed model and by the ns-3 simulator. Results are presented for both PHY and MAC layer, and the impact of the model's assumptions is shown to be mostly negligible, or at least acceptable. Finally, some results will showcase how the model can be used to gain insight on the behavior of the LoRaWAN technology in a quick and effortless way, analyzing the effects of various parameters on the performance of the network. In the plots of this section the analytical results are represented by lines, while markers correspond to simulation outcomes.

Fig. 4.13 shows the packet outcome probabilities at the PHY layer in a network employing confirmed traffic. Although obtained with different approaches, such probabilities are overall consistent, proving the effectiveness of the model.

The good match between model and simulation is also reflected in Fig. 4.4, which shows the CU and CD metrics for a network in which all EDs generate confirmed traffic ($\alpha = 1$), and for different values of m. Also in this case, the model results are quite close to those given by the simulations. Fig. 4.4a shows that the number of available transmissions helps the correct delivery of the message at the MAC layer, providing performance above 0.9 also for relatively high traffic levels, when an average of one packet per second is generated by the network at the application layer. The CD performance shown in Fig. 4.4b exhibits a similar behavior, but reaches much lower values mostly because the rate of DL messages that the GW can generate is limited by the DC restrictions. The fact that this loss in performance is caused by the GW's



(a) CU for different values of m, $\alpha = 1$

(b) CD for different values of m, $\alpha = 1$

Figure 4.4.: Comparison of model and simulation results in terms of CU and CD.

DC is confirmed by the lilac dash-dotted line in Fig. 4.4b: to obtain these results, the DC restrictions were lifted by setting $\delta_{SB1} = \delta_{SB2} = 0$ in the model, producing markedly better results when compared to the corresponding green curve, where DC is enabled. Another example of the model's flexibility in considering also non-standard settings is given by the densely dash-dotted brown line, which represents the CD metric when $\delta_{SB1} = \delta_{SB2} = 9$, i.e., when transmissions in both sub-bands are subject to a DC of 10%. Although being an ideal setting, this case shows that even a small increase in the DC allowance in SB1 can yield considerable performance gains.

Fig. 4.5 compares simulation and theoretical results, in terms of UU, CU and CD, when different fractions of confirmed traffic are employed in the network. For this comparison, we set the network application layer packet arrival rate

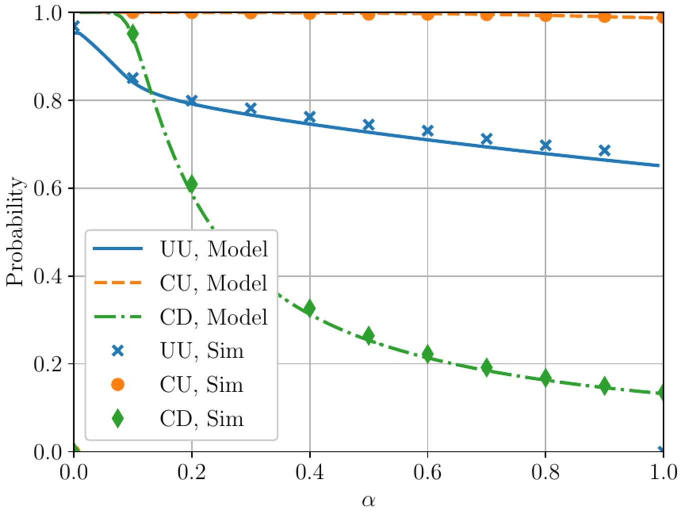


Figure 4.5.: Performance when varying the fraction of confirmed traffic, with $\lambda = 1, m = 8, h = 1$.

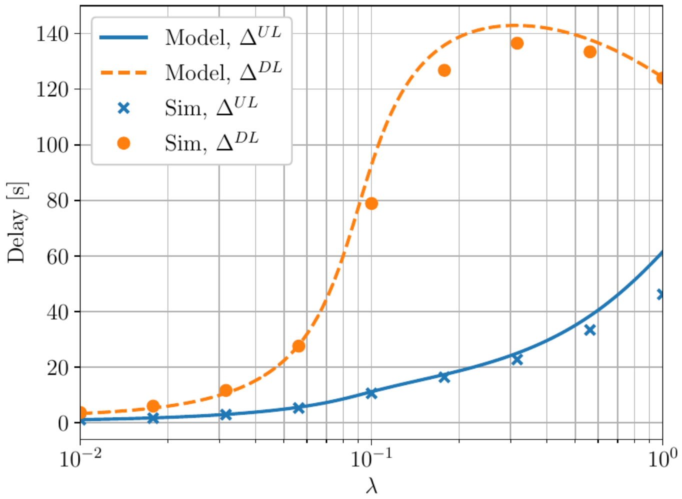


Figure 4.6.: Delays for a confirmed traffic network, $m = 8$.

to $\lambda = 1$ pck/s, the maximum number of transmissions for confirmed traffic to $m = 8$, and the number of repetitions for unconfirmed traffic to $h = 1$. As the fraction of EDs employing confirmed traffic increases, the data delivery performance decreases for all the EDs, in particular for nodes employing unconfirmed traffic which do not have the chance of re-transmitting their packets. The match between the simulator and the model is confirmed to be excellent for all values of α.

The final metric that we evaluate through both model and simulation is the delay, as described in Sec. 4.2.8. Fig. 4.6 shows how delays generally increase with the traffic load, since more re-transmissions are needed to successfully deliver a packet. Note that for high values of λ the average ACK

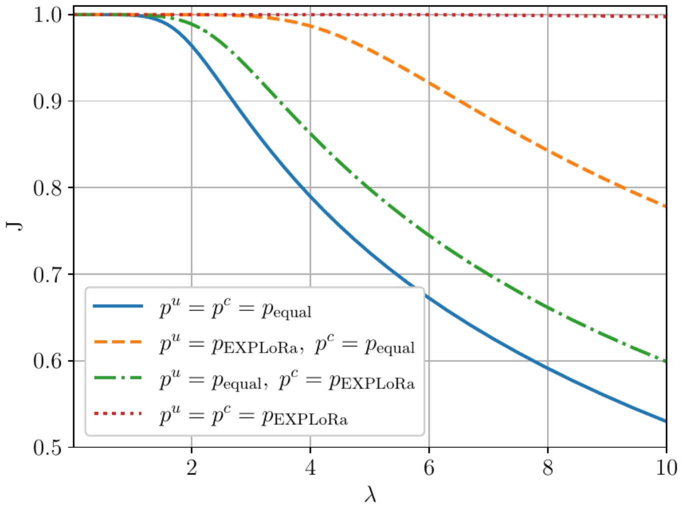


Figure 4.7.: Fairness for different SF distributions when $m = 8, h = 8, \tau = 1, \alpha = 0.3$.

delay Δ^{DL} decreases: this is explained by the fact that devices employing higher SFs, (which may increase the average delay due to their longer inter-packet transmission times) heavily suffer from interference and are often dropped (unsuccessful packets are not considered in the delay computation). Although not shown here, it is worth noting that the model formulation makes it easy to extract per-SF metrics that can help troubleshoot the network configuration under study.

We now analyze how the fairness varies with the traffic load for different configurations of α, p^u and p^c. We consider the SF distributions p_{equal} and p_{EXPLoRa} as defined in Tab. 4.2. The p_{EXPLoRa} distribution, first presented in [74], aims at equalizing the aggregate time on air of each group of devices employing the same SF to minimize the collision probability. In Fig. 4.7 we can observe that, when the SFs are uniformly allocated independently of the traffic type (i.e., $p^u = p^c = p_{\text{equal}}$), the fairness decreases for an increasing traffic intensity. Indeed, as the traffic grows, nodes employing lower SFs will suffer less from interference because of the shorter transmission times. The fairness grows when $\alpha = 0.3$ and $p^c = p_{\text{EXPLoRa}}$, since with this configuration 30% of the generated packets will use lower SFs with higher probability, diminishing the channel and GW occupancy. However, since the traffic load is high and the fairness is measured on the uplink performance (UU and CU), the beneficial effect of allocating SFs according to the p_{EXPLoRa} distribution are more evident when it is used for most of the devices, i.e., the 70% of nodes employing unconfirmed traffic. Finally, the maximum fairness is achieved when the SFs are allocated using p_{EXPLoRa} both for p^u and p^c (dotted line in Fig. 4.7). Note

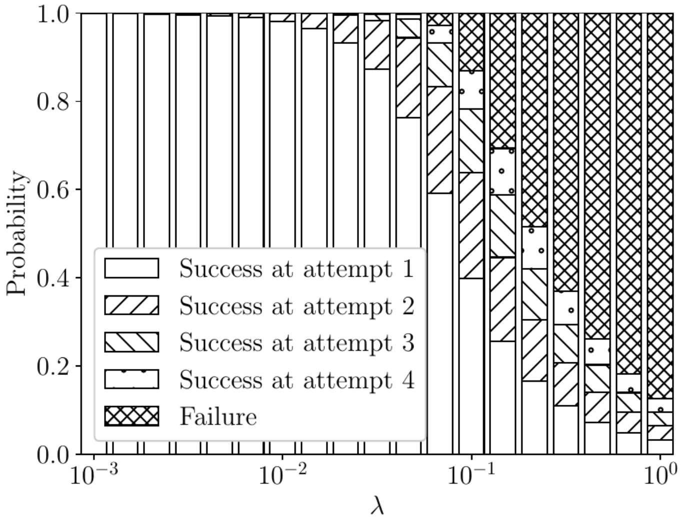


Figure 4.8.: Distribution of re-transmissions, $m = 4$, $\alpha = 1$.

Table 4.2.: Configurations employed in Fig. 4.9

Configuration	τ_1	τ_2	m	h	$p^u = p^c$
C1	1	1	1	1	p_{equal}
C2	0	1	1	4	p_{EXPLoRa}
C3	0	1	4	4	p_{EXPLoRa}

that, when $\lambda \leq 1$, the load in the network is low enough to have $J = 1$ for every p^u, p^c, since the collision probability is low and the GW is not busy with ACK transmissions.

An example of insight that the analytical model can offer is presented in Fig. 4.8, which shows the fraction of traffic that achieves success after a certain number of re-transmission attempt for different traffic loads, derived from $P_{i,j}^{DL}$. This data, for instance, can be used to estimate the power consumption at the nodes: for low traffic loads the vast majority of MAC layer packet transmissions succed with just one PHY layer transmission attempt. As the traffic load increases, the fraction of devices needing multiple re-transmissions to correctly receive an ACK correspondingly increases. After a certain point, packet reception fails with such a high rate that most EDs need to employ the maximum number of transmissions and, despite the high energy expenditure, still fail to receive an ACK from the GW.

Finally, we show how the model can be applied to investigate the impact of different network parameters on the performance. In the example of Fig. 4.9, 30% of the EDs employ confirmed traffic, and we show results obtained with the proposed mathematical model. The parameter configurations are summarized in Table 4.2. Configuration C1 provides a baseline: priority is given

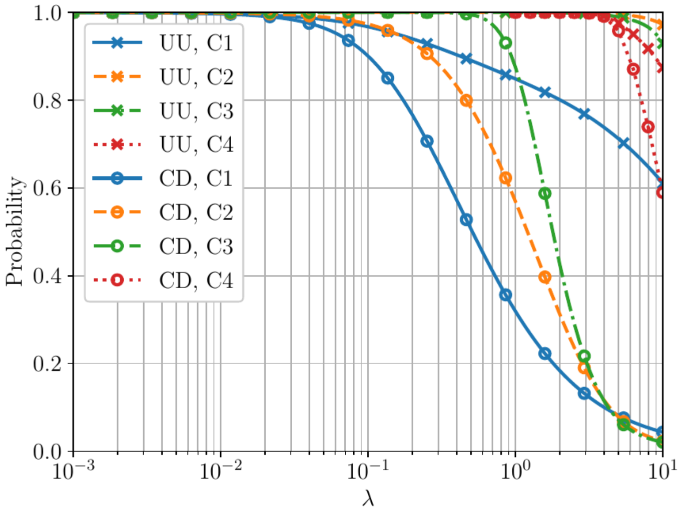


Figure 4.9.: UU and CD performance for different network configurations, $\alpha = 0.3$.

to DL transmission in both windows, devices employ a single transmission attempt for both confirmed and unconfirmed traffic, and SFs are uniformly distributed. In this case the curves have a shape similar to those shown in Fig. 4.4 for $m = 1$, but, since fewer devices require ACKs, the GW is able to receive more packets and profitably send replies, leading to better performance. To improve UU a second configuration (C2) considers the prioritization of ACK transmissions in RX2, where their reception suffers less interference. Moreover, unconfirmed packets are sent multiple times and we use $p^u = p^c = p_{\text{EXPLoRa}}$. This configuration provides a considerable improvement with respect to the UU metric, and some gains are also achieved in the CU performance. To improve also the results for confirmed traffic, a further step (configuration C3) is to set $m = 4$. This provides a significant improvement of CU, at the cost of a (minimal) decrease in UU performance.

As a final step, we fully leverage the analytical model to identify the optimal parameter configuration (i.e., m, h, p_u and p_c) for each plotted traffic load, with the objective of maximizing the average of UU and CU. The red curves of this setting (C4) show how this optimization process enabled by the model can significantly improve the global performance of the network, significantly improving the CD performance at the price of a very small reduction in packet success rate for unconfirmed devices.

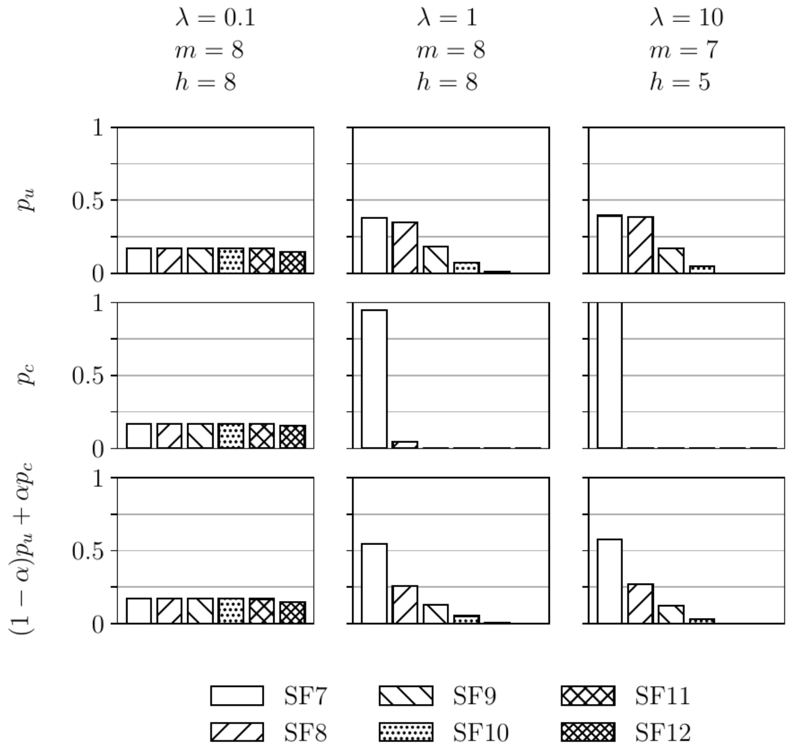


Figure 4.10.: Optimal values of p_u, p_c, m and h as computed through model-driven optimization, for various values of λ.

The optimization problem that is solved to obtain configuration C4 is defined as:

$$\begin{aligned} \max_{p_u, p_c} \quad & \mathrm{UU} + \mathrm{CD} \\ \text{s.t.} \quad & 0 \leq p_i^u \leq 1 \\ & 0 \leq p_i^c \leq 1 \\ & \sum_i p_i^u = 1 \\ & \sum_i p_i^c = 1 \end{aligned} \tag{4.38}$$

where we explore the entire space defined by m, h and λ, by solving (4.38) to find the best p_u and p_c, and finally pick the best solution for each λ. The search is performed using the trust region method as implemented by the `scipy` library, and we always set $p_i^u = p_i^c = 1/6$ as the initial parameter value for the algorithm.

Fig. 4.10 displays the parameters of configuration C4 for some representative values of λ, showing p_u in the first row, p_c in the second row, and a combination of the two weighed on α on the third row. For a low value of generated traffic ($\lambda = 0.1$, first column), we see that the optimization stops almost immediately, yielding a distribution that is very similar to the initial value of p_u and p_c.

In this case, as can also be seen in Fig. 4.9, since the traffic load is low the performance is indeed very good for high values of m and h, and needs little optimization of the SF distributions. For $\lambda = 1$, instead, the optimization process yields a more distinctive value of p_c, setting almost all devices to use SF7. This is motivated by the fact that, RX1 is set to employ the same SF used in the UL. Therefore, having most of the confirmed devices employ an SF as low as possible is advantageous, since it guarantees faster ACK transmissions in the DL and, as a consequence, shorter silent times imposed by the DC, and a larger set of devices can thus be served. Devices employing unconfirmed traffic, instead, are set to use a variety of SF values. Notably, the selected values are such that the aggregated distribution considering both unconfirmed and confirmed traffic (visible in the third row) takes a shape that is very similar to that of p_{EXPLoRa}. This behavior is even more marked when $\lambda = 10$, with the notable difference that higher SF values are not used in the optimized network: this is because of the limited number of demodulators at the GW (a factor which is accounted for in our model). Indeed, although using all SF values would bring an additional gain, a packet with high SF value occupies a demodulator for quite a long time, increasing the probability that other incoming packets are dropped because of unavailability of reception chains at the GW. Finally, we note that m and h are consistently set to their maximum values (8 here) up to $\lambda = 1$. After this value, instead, it pays off to reduce the number of repetitions employed by both unconfirmed and confirmed EDs.

Although this analysis showcases the potential of the mathematical model to identify the optimal settings, an evaluation of the trade-offs associated to parameter configurations and their effect on other metrics of interest, such as delays and energy consumption, needs a deeper investigation, which we leave for future work.

4.2.11 Conclusions

In this section, we presented a model for the performance evaluation of a LoRaWAN network in the presence of both confirmed and unconfirmed traffic, taking into account the influence of different settings of multiple network configuration parameters.

The model is able to capture both the PHY layer and MAC layer performance, and describes the multiple events that affect both UL packet reception and DL transmission: interference, capture effect, availability of demodulator, DC constraints, ongoing transmissions and receptions. We validated the model results with ns-3 simulations, showing the consistency among the two sets of results. Finally, we presented some examples of how the model can be employed to analyze the effects of possible changes to the standard parameter settings, and to identify optimal configurations with minimum effort.

Several extensions of this work are possible. A first improvement to the model is the inclusion of multi-GW scenarios, where UL packets are potentially received by several GWs, and the network DL capacity is increased. A second aspect of interest is to leverage the proposed model to better investigate trade-offs among different network parameters in various scenarios, or when specific performance requirements are provided. A third possible improvement would involve characterizing the capture effect for non-uniform spatial distribution of the devices. Finally, a fourth direction is to employ the proposed model to identify optimal network settings when different metrics of interest are used as optimization functions, as we showed in the results section with some simple cases. We point out that the target of the model was to explore the capabilities of LoRaWAN networks, thus, in this work, we neglected some features of LoRa, such as the interference between overlapping packets modulated with different SFs. The model can be extended by including this, as well as other specific features of the LoRa technology. Such extensions are left for future work.

We remark that all figures contained in this section, covering both model evaluations and simulation results, can be easily reproduced using the tool available at [83].

4.3 A thorough simulation study of LoRaWAN performance under different parameter settings

As presented in Sec. 2.3 and Sec. 2.4, the LoRaWAN standard offers large flexibility in the network configuration, which is another attractive factor. Indeed, the NS can choose the SF used by the different nodes, the duration of the receive windows, the transmission/reception channels, the priority of acknowledgement and downlink data packets, and so on. By properly setting these parameters it is hence possible to support reliable/bidirectional communications and to change the balance between communication reliability, delay, energy-efficiency and system capacity. However, while the effect of certain parameters settings can be predicted in simple scenarios, with a relatively low number of nodes, the interactions among the different mechanisms of the system become much more complex and less intuitive in large-scale scenarios.

In the following of this section, we leverage realistic network simulations to gain insight on the performance of LoRaWAN technology in such scenarios, and show how even small adjustments in MAC layer parameters can significantly affect the system performance (e.g., the packet success ratio). By doing so, we highlight some inherent issues raised by the DC limitations in European ISM bands, and propose some improvements to mitigate the impairments that LoRaWAN may experience at scale. Such simple ingenuities can help increase the number of devices that can be served by a single gateway, postponing the potential collapse of the network in overcrowded scenarios and reducing the network management costs created by inefficient network layouts.

In this study we differ from the existing literature described in Sec. 4.1.2 in that we target large networks with bidirectional traffic, a scenario that makes it possible to observe some unforeseen effects rising from the interaction of multiple nodes served by one single GW and NS. Furthermore, in our analysis we examine one by one the role played by the configurable network parameters, as detailed in Sec. 4.3.1, thus highlighting some pitfalls that can affect the network performance. We then propose possible counteractions that require some small changes at the MAC layer, and we evaluate their effectiveness in some representative scenarios.

4.3.1 Available network settings

The analysis carried out in this work leverages the ns-3 `lorawan` module described in Chapter 3. Next we give a brief introduction to the network configuration options that are available in the simulator and that make it possible to control the behavior and features of both GW and EDs.

Gateway DC In the simulator we have the opportunity of turning on or off the DC restriction at the GW to analyze its impact on the network performance.

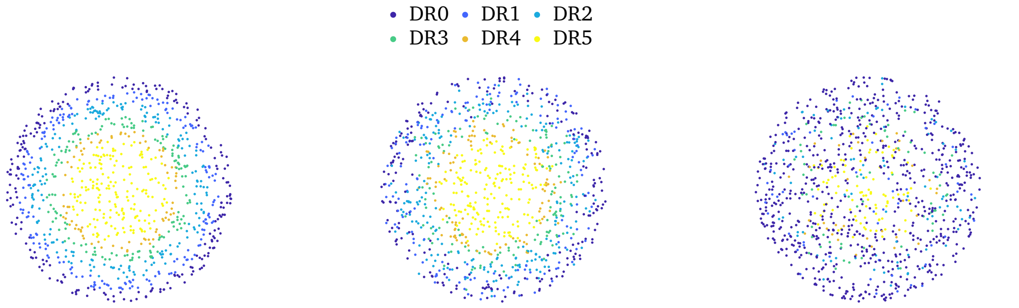
Transmission/reception priority LoRaWAN GWs cannot receive and transmit simultaneously. In commercial GWs, if there is the need to send an ACK, the reception of any incoming signal will be immediately interrupted to start the DL transmission. In this section, we call this standard behavior “transmission (TX) prioritization.” In addition to this mechanism, here we also consider the RX prioritization option, according to which DL transmissions are performed only if the GW is not engaged in any packet reception. In case the GW is receiving an UL message at the time when it should be sending an ACK, it will continue the reception procedure and postpone the DL transmission to the second receive window, if available. If a reception is also occurring when the second receive window opens, the ACK is definitively dropped. This feature could easily be implemented in future LoRa chips, only requiring minimal changes to the behavior of the NS and GWs.

Sub-band prioritization The LoRaWAN standard requires that RX1 is opened on the same channel where the corresponding UL was received, while RX2 is opened on a dedicated DL channel, which in Europe also features more lenient DC restrictions (10% instead of the 1% allowed on the other channels). In the simulator, we have enabled a mode that switches this setting, making it possible to open RX1 on the dedicated DL channel, and RX2 on the channel used for the UL communications. The effect of this trick will be illustrated in Sec. 4.3.4.

Acknowledgment data rate The LoRaWAN specifications recommend that ACKs transmitted on RX1 should use the same SF for the UL transmission, while transmissions on RX2 use the lowest available data rate (SF=12). To explore other options, the simulation module has been modified to enable the use of higher data rates on both the reception windows. This setting involves a trade-off between robustness and efficient use of the available DC and time resources. Note that such an option can actually be implemented in LoRaWAN through a dedicated MAC command.

Number of transmission attempts For confirmed traffic, the maximum number m of transmission attempts for the same message is configurable, and can take values in the set $\{1, 2, 4, 6, 8\}$. The number of transmission attempts for unconfirmed traffic, instead, is fixed to $h = 1$.

Full-duplex GW As mentioned, currently GWs cannot transmit and receive simultaneously. However, it might be interesting to investigate the potential performance gain obtained by implementing a full-duplex GW. This functionality may be realized by co-locating two GWs or combining a GW with a simple LoRa chipset to be used for transmissions only, leaving the GW free to receive messages. Note that, because of this, whenever the Full Duplex Gateway (FDGW) configuration is employed, the distinction between RX and TX prioritization loses meaning. In order to test this functionality, we added a new mode to the `lorawan` module in the ns-3 simulator that allows for ideal full duplex communication.



(a) Open air scenario: only path loss.

(b) Path loss and shadowing.

(c) Urban scenario: path loss, shadowing and buildings.

Figure 4.11.: Distribution of Data Rates for different channel models.

Number of reception paths The number r of parallel reception paths in the GW is a parameter that can be toggled in the simulator. Beside the standard value $r = 8$, we also considered the values $r = 3$ and $r = 16$ to study how the parallel reception capabilities of the GW can affect the overall system performance.

4.3.2 Reference scenarios

We considered two main simulation scenarios. Since we are interested on the optimization of MAC layer parameters, we assume a single GW serving multiple EDs, which generate packets periodically, with equal period but random phases. Furthermore, the traffic generated by the devices can be either confirmed, unconfirmed, or mixed, i.e., with half of the devices requiring acknowledgments and the other half sending unconfirmed packets.

In the first scenario, we assume that EDs are randomly distributed within the coverage range of the GW, and we only consider path loss.

The second scenario consists of a more *realistic urban deployment*, where EDs are randomly located outside or inside buildings having different height and wall width, following the model described in [16]. Here, the channel propagation is affected by path loss, spatially correlated shadowing, and attenuation due to the presence of buildings, as described in [39]. To obtain a realistic setup, we consider the traffic model described in the Mobile Autonomous Reporting (MAR) reports [16], according to which the devices send packets with periods that vary from 30 minutes to 24 hours, as described in Tab. 4.3. The number of devices is also varied to estimate the capacity (in terms of number of active devices) that can be supported by a GW in a realistic scenario.

Table 4.3.: Interarrival times in realistic simulations.

Inter-arrival time	% of devices
1 day	40%
2 hours	40%
1 hour	15%
30 minutes	5%

To ease the interpretation of the results, we neglect short-term fading phenomena that may affect the received signal power, also considering that the chirp modulation is rather robust to multi-path fading. The effects of the channel model on the distribution of the SFs (and, thus, of DRs) can be observed in Fig. 4.11, where dots show the position of randomly placed EDs around the central GW, while colors are used to represent the bitrate of each device, i.e., its DR value (see Tab. 2.9). The bitrate is the highest permitted by the signal received power at the GW, according to the sensitivity thresholds in Tab. 2.9. Note that the rate distribution becomes more erratic in presence of long-term shadowing and wall attenuation factors that affect the propagation.

4.3.3 Performance metrics

A packet transmission at the PHY layer can have five possible outcomes:

- *Success* (S): the packet is correctly received by the GW.
- *Lost because under sensitivity* (U): the packet arrives at the GW with power lower than the sensitivity, and the GW can not lock on it.
- *Lost because of interference* (I): the packet is correctly locked-on by the GW, but its reception fails because of the interference from overlapping packets with enough power to disrupt signals orthogonality.
- *Lost because of saturated receiver* (R): as mentioned, a GW usually has multiple reception paths, each configured to listen to a specific channel frequency. A packet transmitted on a certain channel is lost because of saturated receiver if it gets disregarded by the GW because all the reception paths for that channel are already locked receiving some other signals. Note that such a packet would be correctly received, if not for the saturation of the receiver. In other words, this error event would never occur if the GW had a sufficient number of reception paths for each frequency channel.

- *Lost because of GW transmission* (T): the packet reception gets disrupted by the transmission of a DL packet (which could either be ongoing at the packet arrival time, or started during the packet reception, in case the GW gives priority to transmission).

In the case of unconfirmed traffic, we label a packet as *successful* when it is successfully received at the GW that, in turn, forwards it to the NS through a reliable connection. For confirmed traffic, we distinguish two cases, depending on whether the DL packets carry information (e.g., the UL packet is a query to the NS, and the corresponding DL packet is the reply), or are just an ACK used to stop retransmissions of the UL packets. In the first case, the transmission is successful when both the UL and the successive DL packet are successfully received by the NS and ED, respectively, within the available transmission attempts. In the second case, instead, we assume that a transmission is successful if at least one of the generated UL packets is delivered to the NS, irrespective of whether the ACK is received by the device.
Accordingly, we define two performance metrics:

CD probability that both the confirmed UL packet and the corresponding DL packet are correctly received in one of the available transmission attempts;

UL-PDR probability that an UL packet is correctly received (whether or not the ACK is requested), in at least one of the available transmission attempts (which are just 1 for unconfirmed traffic, and m for confirmed traffic). Note that this is analogue to the sum of UU and CU metrics as defined in Sec. 4.2.8.

4.3.4 Performance evaluation

In this subection we first provide the baseline for our performance analysis considering the default settings, which reveals some issues with the current LoRaWAN standard. Then, we study the impact of the configurable parameters, and finally validate the effectiveness of the proposed improvements using the simulator described in Chapter 3.

Baseline performance analysis

To begin with, it is interesting to compare the performance attained by confirmed/unconfirmed traffic in the mixed and homogeneous scenarios, for the same offered traffic at the application level. For these results, we used the

Table 4.4.: Default parameter settings.

Parameter	Value
GW DC	On
TX/RX priority	TX priority
Sub-band prioritization	Off
RX2 data rate	Lowest (SF 12)
Number of TX attempts m	8
Full-duplex GW	No
Number of reception paths r	8

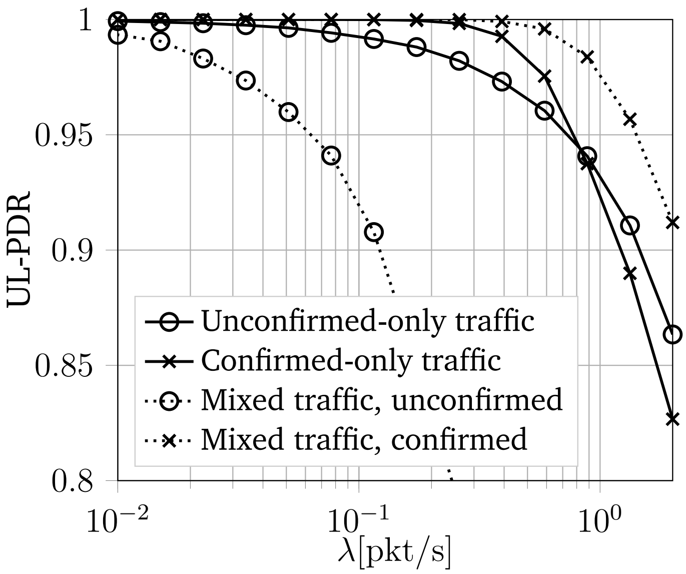


Figure 4.12.: Baseline UL-PDR performance for different kinds of traffic.

default settings, as described in Tab. 4.4. The solid lines in Fig. 4.12 show the UL-PDR for the confirmed-only and unconfirmed-only cases (crossed and circle markers, respectively), while the dashed lines refer to the performance experienced by the two types of traffic sources in the mixed scenario. It is apparent that the mixture of confirmed and unconfirmed traffic sources favors the first class of sources, but penalizes much more severely the latter, with respect to the corresponding homogeneous traffic cases. Focusing on the homogeneous traffic scenarios, we can see that the use of confirmed traffic maximizes the UL-PDR index up to an aggregate traffic load of almost $\lambda = 0.8$ pkt/s at the application layer (not including retransmissions). Beyond this point, it is more convenient to use unconfirmed-only communications. The reason behind this behavior becomes apparent in Fig. 4.13, which reports the fraction of packet losses caused by different events (I, R, T, see Sec. 5.1.2) for the two homogeneous scenarios. The results are obtained for an offered traffic of

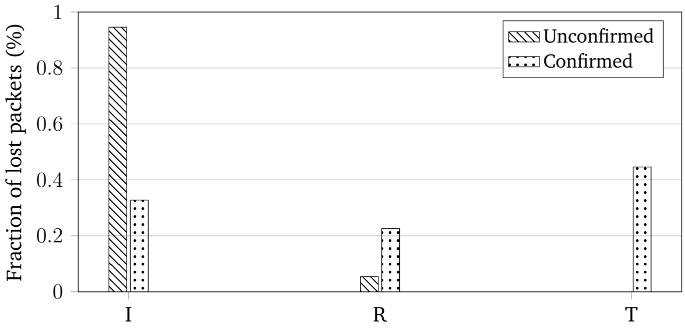


Figure 4.13.: PHY outcomes for traffic achieving the same UL-PDR.

$\lambda = 0.8$ pkt/s, for which the UL-PDR is the same for both the homogeneous scenarios. We can observe that, with only unconfirmed traffic packet losses are mainly due to the interference (I) produced by multiple UL transmissions. Instead, confirmed traffic (with $m = 8$), in addition to losses caused by interference, also suffers from other impairments, such as the saturation of the GW's reception paths (R), and collision with ACKs (T), which plays a major role among the causes of failure. Therefore, confirmed traffic may enhance the data collection capabilities of the network as long as the overall load is light, but it can yield significant degradation of the PHY layer performance for higher loads, which in turn impairs scalability. In the remaining of this subsection we will investigate the impact that the parameters introduced in Sec. 4.3.1 can have on the performance metrics, and explore some simple precautions that can significantly improve both performance and fairness in LoRaWAN.

Gateway DC

The impact of the DC restriction at the GW is visible only when confirmed traffic is required by the EDs. The solid line with cross markers in Fig. 4.14 shows the baseline CD performance obtained in the case of only confirmed traffic. The solid line with circle markers, instead, gives the CD that can be obtained by removing the DC constraint at the GW. Comparing the two curves it is clear that the DC restriction at the GW represents a severe bottleneck in terms of CD since successfully received UL packets may not be acknowledged by the NS in the due time because of the DC limitations of the GW. Furthermore, the missed ACKs exacerbate the UL traffic load, triggering retransmissions of otherwise successfully delivered UL packets.

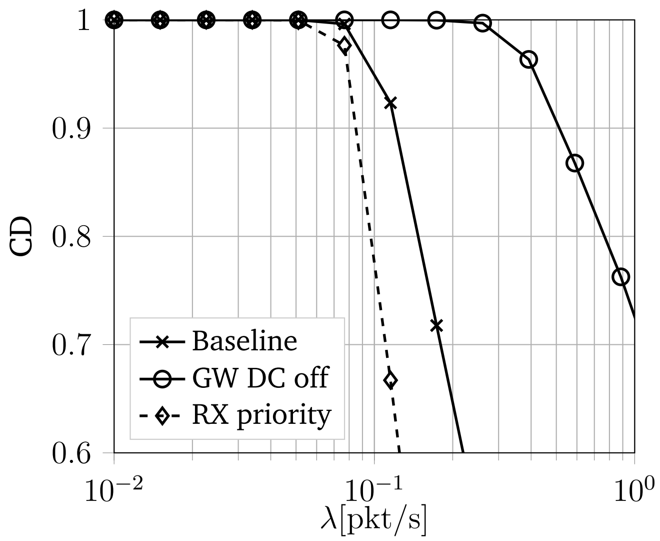


Figure 4.14.: CD of a network with only confirmed traffic sources.

Priority of transmission over reception

The effects of reception (RX) prioritization at the GW have been investigated both in terms of CD (Fig. 4.14) and UL-PDR (Fig. 4.15). It is worth to observe that RX prioritization can be implemented at the GW by simply avoiding transmissions of DL packets if at least one of the eight parallel receive chains is occupied.

Fig. 4.14 shows that giving priority to RX yields a CD loss. In fact, as λ increases, the number of UL packets that are successfully received by the GW increases more rapidly than in the default case where TX is prioritized, and the probability that the GW is in the reception state quickly approaches 1, thus preventing the GW from transmitting ACKs. This, in turn, triggers packet re-transmissions from the devices. On the other hand, as shown in Fig. 4.15, in mixed traffic scenarios, the RX prioritization improves the performance of both confirmed and unconfirmed traffic sources in terms of UL-PDR. In summary, giving priority to RX at the GW makes it possible to receive more UL packets, but this can yield to congestion in the DL channel.

More generally, DL packets could be marked by the NS based on their importance for the ED (which can either be explicitly signaled through a MAC header bit or inferred by the NS based on the application that is generating the data flow). If ACKs are required, the DL packet could be marked as prioritized over reception, and immediately sent by the GW. If, on the other hand, confirmation is merely used to stop retransmissions and the ED is only interested in

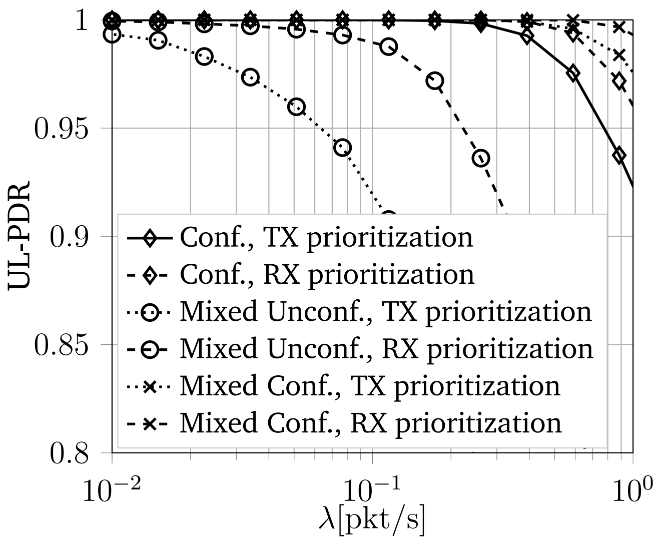


Figure 4.15.: UL-PDR performance for confirmed and mixed traffic when RX or TX is prioritized.

maximizing its UL-PDR, then ACKs could be marked as low priority, and the GW would send them only if idle.

ACK variations

We now analyze the effect of two variations to the standard acknowledgment mechanisms, named Sub-band swapping and ACK Data Rate, that try to alleviate the bottleneck due to the DC restrictions at the GW and improve the system performance in terms of throughput and energy efficiency.

1) Sub-band swapping: As mentioned before, RX1 is always opened on the same sub-band used for the UL transmission, while RX2 is opened on a sub-band reserved to DL transmission, whose DC is 10%. Therefore, ACKs sent in RX1 will compete with other UL transmissions, generating and suffering interference, and can rapidly consume the 1% DC of that sub-band. We hence explored whether any benefit could come from swapping the sub-bands used for RX1 and RX2: we hence implemented a sub-band swapping scheme, according to which RX1 opens in the DL-reserved sub-band, while RX2 opens in the shared sub-band used for the UL transmissions.

2) ACK Data Rate: By default, LoRaWAN devices use the highest available SF (and thus the lowest DR) in RX2, in order to increase the probability that the downlink packet is received correctly. However, this can be detrimental, since longer transmission times of ACKs will rapidly consume the DC budget at the GW. To study which effect is dominant, we implemented the "ACK Data Rate

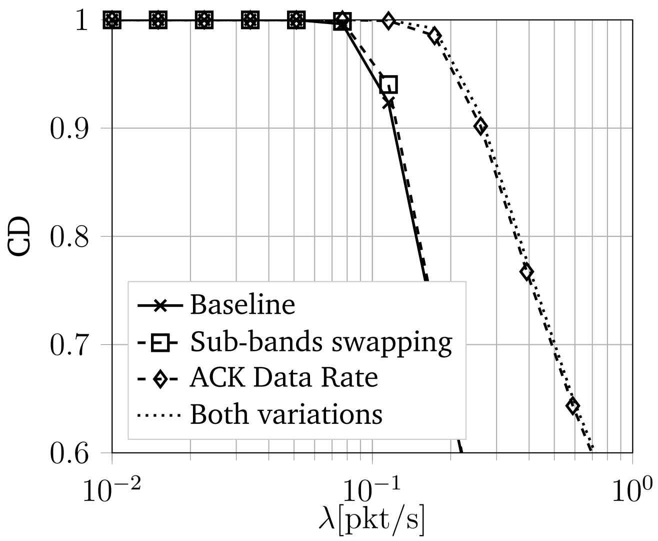


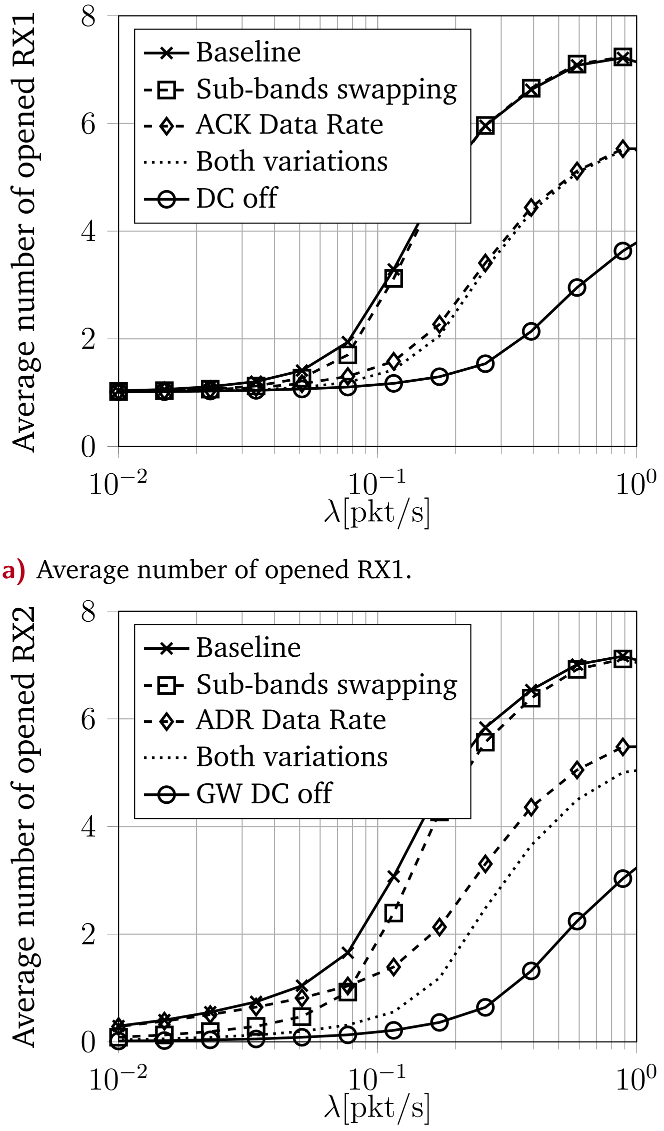
Figure 4.16.: Effect of improvements on CD.

scheme", where all DL transmissions are always performed at the same DR used for the corresponding UL transmission.

In Fig. 4.16 we report the CD achieved by using the default setting (solid line with cross markers), each one of the ACK improvement schemes (dotted lines with square and diamond markers, respectively), and both the improvements together (dashed line without markers). We can observe that the sub-band swapping has a very marginal (yet positive) impact in terms of CD, which implies that the interference produced by UL transmissions on DL reception is not very significant. Conversely, the use of the same DR in all receive windows brings a significant gain in terms of CD over the baseline. We can hence conclude that the use of the lowest DR in RX2 can severely limit the performance of the system, in particular when the missed reception in RX1 is not due to channel impairments, but rather to DC limitations of the GW in that sub-band.

A better strategy to provide efficient and reliable DL transmissions is hence to implement independent rate-adaptation strategies on all DL sub-bands, rather than following the very conservative policy of retransmitting at the basic rate to increase robustness, but at the cost of lower spectral efficiency.

The two ACK improvement schemes also have a positive impact on the energy consumption of the EDs. Indeed, the sub-band swapping mechanism makes it possible to return a larger number of ACKs in RX1, thanks to the looser DC constraint of the DL-reserved sub-band, thus avoiding the need to open RX2. This effect can be observed in Fig. 4.17, which shows the average number of times RX1 (above) and RX2 (below) are opened by the EDs, with the



(a) Average number of opened RX1.

(b) Average number of opened RX2.

Figure 4.17.: Effects of the proposed ACK improvements on the average number of opened RX1 and RX2 windows.

max number of retransmissions set to $m = 8$. The gain, however, tends to vanish as the traffic increases, since both sub-bands will then be used to return ACKs. We can also notice that, by using the same DR in both receive windows we significantly reduce the average number of opened receive windows per transmission, also for a relatively high traffic. Indeed, transmitting DL packets at a higher rate contributes to alleviate the DC impairments, allowing the GW to serve more devices. In turn, this reduces the number of retransmissions and, consequently, the number of RX1 and RX2 that need to be opened.

The effect of the proposed ACK variations on the UL-PDR metric are depicted in Fig. 4.18 for a network of only confirmed traffic sources. In this case, both

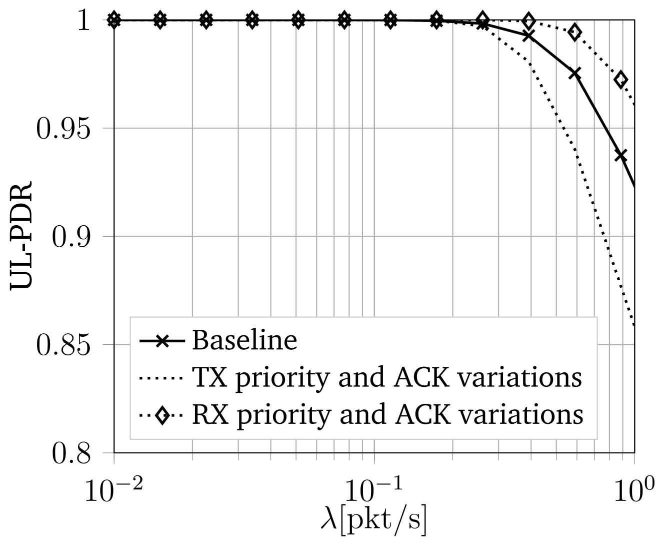


Figure 4.18.: UL-PDR performance in case of only confirmed traffic, when ACK improvements and RX priority are applied.

sub-band swapping and ACK Data Rate mechanisms yield worse performance, when the GW adopts the standard TX prioritization policy. This is easily explained if we consider the type of DL traffic that a saturated network (i.e., one where the ACKs queues are always full) will generate when the proposed improvements are turned on and off: in the default case, long DL transmissions using low data rates will be followed by long waiting times due to the DC. During these silence periods, the GW will be forced to listen to the network, resulting in an improved UL-PDR performance. If, on the other hand, the GW sends short DL packets, it can do so more frequently, and in turn lose more UL packets because of DL transmissions. This behavior, however, can be counteracted by prioritizing RX over TX: as Fig. 4.18 shows, with this configuration we get the best of both worlds, achieving simultaneously the improvements on UL-PDR and the energy saving benefits obtained with the sub-band swapping and use of the "ACK Data Rate" schemes.

One final consideration regards networks in which some devices are interested in the UL-PDR metric, while other need to maximize their CD: in this case, the considerable improvement in CD brought by the proposed acknowledgment variations would yield a slight loss in UL-PDR, which could be further reduced by implementing the dynamic transmission prioritization scheme proposed in Sec. 4.3.4.

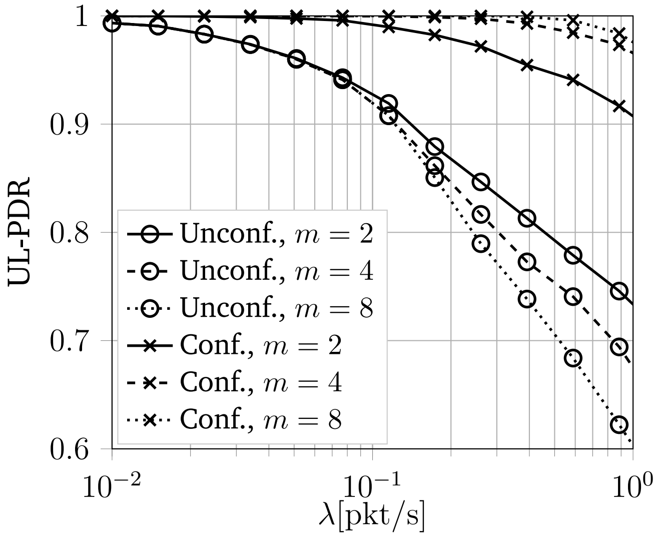


Figure 4.19.: UL-PDR for mixed traffic, different values of m.

Number of transmission attempts

Our results showed that increasing the maximum number m of transmission attempts improves the CD by 5-10% (though with sharply diminishing returns as m grows larger). On the other hand, as we can see in Fig. 4.19, smaller values of m can slightly improve the fairness in terms of UL-PDR in mixed traffic scenarios. In particular, at $\lambda = 1$ pkt/s, choosing $m = 4$ instead of $m = 8$ does not change significantly the UL-PDR for confirmed traffic, but yields an improvement in the UL-PDR of unconfirmed traffic, proving the sensitivity of the network performance to the setting of this parameter.

Full-duplex gateway

The impact of a full-duplex GW scheme described in Sec. 4.3.1 is shown in Fig. 4.20, where UL-PDR performance is reported both for the standard GW configuration and for the FDGW. As expected, this solution achieves a rather marked gain in terms of UL-PDR performance.[3]

Number of available demodulators

Our simulation results (not reported here due to space constraints) showed that UL-PDR performance increases with the number of demodulators in the GW, but with diminishing returns after 8 reception paths, as interference still causes a relevant portion of locked-on packets to be lost. Having a chip with

[3] Note that, when FDGW is employed, packets that are being received by the GW are still lost if a transmission on that same channel is performed due to the strong interference.

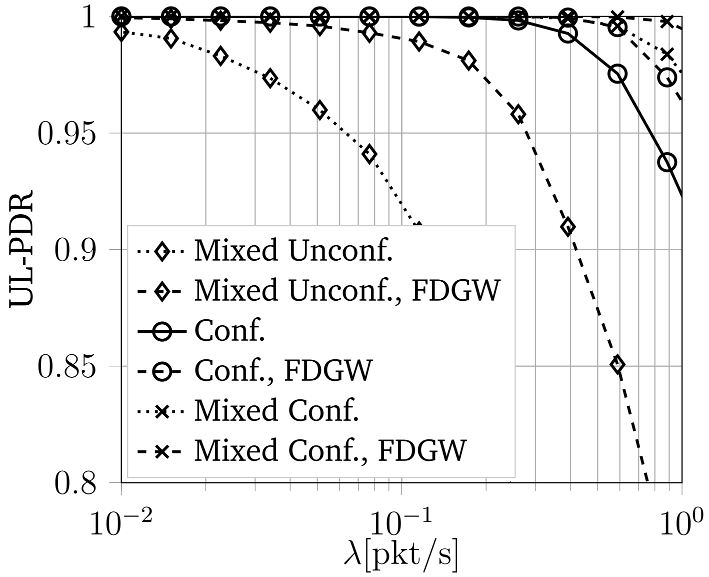


Figure 4.20.: Effects of Full Duplex GW on UL-PDR.

only three parallel demodulators, on the other hand, may enable cheaper gateways, and yield a slightly lower but still appreciable performance. The number of available demodulators, on the other hand, has no impact on the CD, for which the bottleneck is the DL channel due to the DC constraints: by the time the additional demodulators can make a difference in the reception probability of UL packets, DL channels at the GW will already be saturated, limiting the maximum achievable CD.

Best configurations in realistic scenarios

A final simulation campaign had the objective of estimating the impact of the settings described in Sec. 4.3.1 on the performance of a sensor network in a realistic urban scenario. Fig. 4.21a and Fig. 4.21b show the UL-PDR and CD performance obtained in this scenario with baseline and “optimal” parameter settings. The optimal setting actually depends on the target performance index. More specifically, to optimize the UL-PDR, the GW was set in the RX prioritization mode and the maximum number of transmissions was set to $m = 4$, while to optimize the CD, we set $m = 8$ and the GW in TX prioritization mode. In both cases, the GW applies sub-band swapping and ACK Data Rate mode. Note that the results have been plotted against the number of devices in the cell, in order to give an idea of the gain in network size that is achievable through a clever setting of the network’s operational parameters.

Fig. 4.21a shows how the UL-PDR can improve by using the proposed settings configuration, and accommodate up to 4 times the number of unconfirmed devices that it would be possible to serve with standard settings. Similarly, Figure 4.21b shows that the number of devices that can be provided a CD larger than 0.95 doubles when the proposed variations are applied.

Additional observations

EDs locking on uplink packets The LoRaWAN standard does not allow direct transmission between EDs. Nonetheless, the simulation outcomes revealed that, when an ED opens its receive window to listen for DL packets, the device can actually lock onto a message sent in UL by a second ED. Experimental trials with real LoRa devices confirmed this incorrect behavior. This is due to the fact that the same preamble is used in both UL and DL transmissions, so that a receiver is not aware of the transmission source until the packet is completely received and inspected. At the same time, an ED can also lock on a DL message intended for another receiver, experiencing, thus, a waste of energy and time, as the packet will eventually be discarded. The problem of EDs locking on UL messages could be easily avoided by using different preambles for UL and DL transmissions: in this way, the receiver would completely avoid the reception of UL packets and could return to sleep mode for the remaining duration of the ED receive window.

Sensitivities asymmetry In Sec. 2.4.7 and Tab. 2.11 we had introduced the sensitivity requirements of GW and EDs. We can observe that the requirements for EDs are more relaxed, mainly to reduce the manufacturing cost. However, the gap between the capabilities of the two kinds of device causes an asymmetric coverage range between UL and DL transmissions, and it may happen that the SF used by an ED to reach the GW is not sufficient to correctly deliver the return packet to the same ED because of its worse sensitivity: as an example, an UL transmission that arrives at the GW with a power of -128 dBm and SF=7 may generate a DL transmission that, assuming a symmetric channel, will arrive with the same power at the ED, and thus below its sensitivity. While such an asymmetry is not an issue when all nodes send unconfirmed UL traffic, it may become a problem in case of confirmed traffic, since some EDs could be prevented from receiving a DL packet in the first receive window, in which the NS uses the same SF and carrier frequency of the UL message, forcing the systematic opening of the second receive window. This problem can be easily mitigated at the NS by checking that the reception power of the

Table 4.5.: Proposed setting configurations to increase network performance for different types of traffic.

Parameter setting	Unconfirmed	Confirmed		Mixed	
	UL-PDR	UL-PDR	CD	UL-PDR	CD
GW prioritization	RX	RX	TX	RX	TX
Sub-band swapping	–	–	Yes	Yes	Yes
ACK data rate	–	Yes	Yes	Yes	Yes
Number of transmissions m	1	8	8	4	8

packets coming from one ED is not in an interval such that the situation described above can occur (assuming symmetric channel). In such a case, the NS can use the appropriate MAC commands to inform the device that future DL transmissions in RX1 will use a higher SF than that one used in the UL.[4]

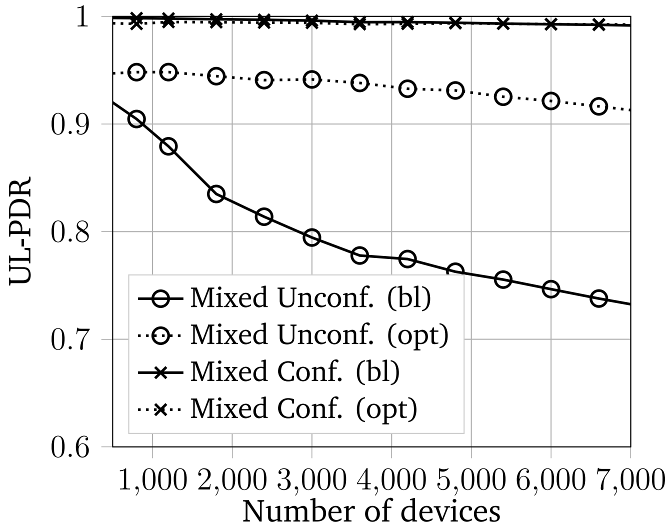
4.3.5 Conclusions

In this section we presented a systematic analysis of the impact of the tunable parameters in LoRaWAN some of which proved to be particularly meaningful. We observed that with a standard configuration, the presence of confirmed traffic sources can significantly degrade the performance of unconfirmed traffic, due to the additional interference generated by DL (ACK) transmissions. Considering only confirmed traffic, instead, the most critical factor appeared to be the DC constraint of the GW, which throttles the DL channel that soon becomes the bottleneck of the system in presence of bidirectional flows. On the other hand, giving priority to RX at the GW can bring benefits to the network when most of the traffic is in the UL direction, but can be detrimental when the GW needs to transmit frequently in DL. Furthermore, we have shown that a rate adaptation strategy and the swapping of the frequencies to be employed in the two DL opportunities can significantly reduce the time the EDs spend in the reception state, thus improving their energy efficiency. Conversely, simulations showed that other system parameters, such as the maximum number of transmission attempts and the number of parallel received paths, appear to be already well configured and dimensioned. Thanks to the analysis described above, we can identify the best settings configurations for various combinations of traffic type and metric of interest, as summarized in Tab. 4.5.

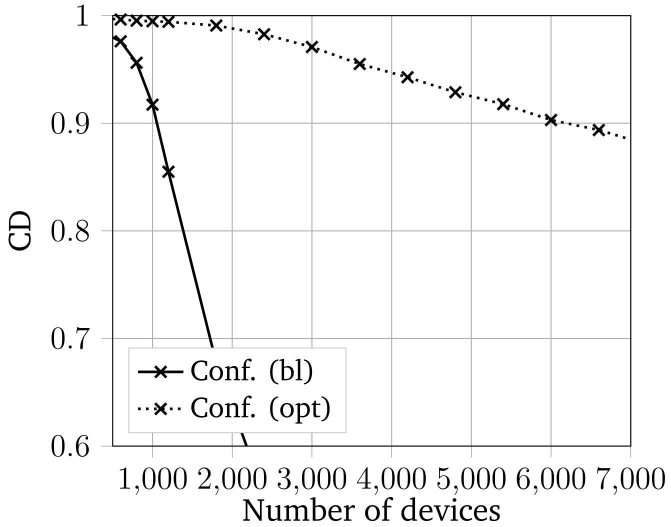
[4] The simulations in this section were performed by setting the ED's SF based on the ED sensitivities, as to avoid this asymmetry problem.

Note that this analysis has been carried out by considering a single GW. In a multi-GWs scenario, UL transmissions are successful when correctly received by at least one GW. Similarly, DL packets can be transmitted, in principle, by any GW in the coverage range of the target receiver. Therefore, we expect better performance for both UL and DL traffic. It is also expected that most of the observations made in the single GW scenario will hold for the multi-GW case. Incrementing the number of GWs, and hence the number of data collecting devices in the network, however, is expected to diminish the gains brought by solutions that attempt at improving the GW's capacity in the UL, e.g., full-duplex GW and RX priority. Finally, we note that the degrees of freedom in system configuration grows considerably with the number of GWs. Although not all configurations are meaningful, the optimization space becomes significantly larger, making the analysis of the possible interactions among different configurations even more complex. A discussion about proper parameter setting in a multi-GW case, therefore, requires an in-depth analysis, which we left for future work.

Overall, the analysis conducted in this section shows how the interplay among the different tunable parameters provided by the system is often subtle and difficult to predict, calling for the development of efficient system design and configuration tools, and motivating the investigation of possible side effects of new policies before deployment.



(a) UL-PDR performance, baseline (bl) and optimal (opt) parameter configuration.



(b) CD performance, baseline (bl) and optimal (opt) parameter configuration.

Figure 4.21.: Simulation results for a realistic scenario.

5 LoRaWAN in non-traditional IoT applications

Although LoRaWAN technology has been designed for traditional IoT scenarios, with no strict requirements in terms of communication reliability, delays and targeting sporadically generated traffic, it is interesting to test it also in non-traditional applications, evaluating whether its low-power and long-range features can be leveraged in other systems. In this chapter, we consider two application scenarios where the LoRaWAN technology has to work under stricter requirements: an industrial IoT scenario, where the communication reliability should be close to 1 while minimizing the communication delay, and a drone tracking application, where the traffic is generated very frequently (up to the DC limitation).

The evaluation of the performance of a LoRaWAN network in industrial scenarios is presented in Sec. 5.1. In this case, several Industrial Internet of Things (IIoT) nodes communicate to a central controller in order to provide monitoring and sensing information that can be used to optimize the efficiency of industrial processes and reduce costs. In particular, we consider confirmed and unconfirmed traffic, multi-gateway deployments, the usage of different classes of devices, and a non-standard channel plan. Furthermore, we analyze the higher-layer impact of different models of LoRa PHY layer with industrial channel models. We show that, with proper configuration, LoRaWAN is able to serve IIoT sensing applications with a packet success rate over 90%, providing at the same time limited communication delays.

In Sec. 5.2, instead, we focus on a drone tracking application. A typical scenario consists in the use of Unmanned Aerial Vehicles (UAVs) for surveillance or target-search missions over a wide geographical area. In this case, it is fundamental for the command center to accurately estimate and track the trajectories of the UAVs by exploiting their periodic state reports. In Sec. 5.2, we design an *ad hoc* tracking system that exploits the LoRaWAN standard for communication and an extended version of the Constant Turn Rate and Acceleration (CTRA) motion model to predict drone movements in a 3D environment. We analyze the trade-off in setting the main parameters of the communication system and ADR scheme, showing how our tracking system can handle large swarms of drones at distances up to 4 km. Simulation results on a publicly available dataset show that our system can reliably estimate the position and trajectory of a swarm of UAVs, significantly outperforming baseline tracking approaches.

5.1 Performance analysis of LoRaWAN in industrial scenarios

One of the key concepts in Industry 4.0 is that of Industrial Internet of Things (IIoT), with the aim of connecting production machinery to information systems and business processes, in order to optimize the industrial operations and quickly reconfigure the production chains to respond to demand changes [85]. The industrial scenarios differ from other IoT environments in three specific aspects: the industrial wireless channel, the traffic patterns, and the QoS requirements. Indeed, when compared to a typical IoT smart city scenario, a smart factory scenario presents higher interference and stronger signal attenuation. Furthermore, packets are generated more frequently and need to be delivered with higher reliability and controlled latency.

The most popular LPWAN technologies are LoRaWAN, Sigfox and NB-IoT: these solutions are typically employed in IoT scenarios such as smart cities, but their robustness to channel impairments and high energy efficiency also appeal to the monitoring use cases of IIoT.

Differently from other IoT technologies such as Sigfox, LoRaWAN allows for the creation of a proprietary network, giving full control of the infrastructure and of the network configuration. In general, a tighter control of the network makes it possible to improve the communication latency and reliability compared to public networks, which normally target a large number of diverse use cases.

Despite the growing body of literature in this field, a number of key questions remain unanswered:

1. Which channel and interference models are appropriate to accurately simulate a LoRaWAN deployment in an industrial scenario? How does this choice affect higher-layer performance?

2. How many infrastructure nodes (gateways) do we need to support IIoT applications?

3. What kind of influence do deployment decisions such as channel plan and class of LoRaWAN devices have on the network performance?
4. To improve reliability, is it more convenient to use confirmed traffic or to blindly replicate uplink transmissions?
5. Is LoRaWAN compliant with key communication service requirements for asset and process monitoring in IIoT applications?

The aim of this section is to address such questions and provide insights on the potential and limits of legacy LoRaWAN technologies in the IIoT domain. To this end, we will present a set of detailed simulation results that focus on the most significant high-level performance metrics for industrial applications, using the realistic, measurement-based channel model [86].

In particular, we introduce some specific PHY aspects that have not been studied in the literature so far, such as the presence of Out-Of-Band (OOB) emissions. Moreover, we analyze the MAC features affecting the system level performance that are most relevant for industrial scenarios, such as transmission reliability and latency.

The rest of the section is structured as follows: Sec. 5.1.1 reviews the literature that deals with this topic. Our simulation setup is presented in Sec. 5.1.2, while in Sec. 5.1.3 we analyze the network performance by considering the questions presented above, and discuss the corresponding results. Finally, Sec. 5.1.4 contains our conclusions and presents possible future developments.

5.1.1 State of the art

Even though different works present applications of wireless communications in industrial scenarios (e.g., [87, 88, 89]) it is a common opinion that, because of the multiple and contrasting requirements of IIoT applications, there is no unique solution that fits all the use cases [90, 91]. In [92] Queralta et al. compare different LPWAN technologies for the Machine-to-Machine (M2M) scenario from the technical point of view, and concludes that LoRaWAN is suitable for both smart city applications and industrial use cases where a small volume of data is required, but in this work no evaluation results are provided. The works in [40, 45] evaluate LoRaWAN networks where confirmed traffic is used to increase the reliability of the communication. However, the authors recongnize that it is potentially damaging for the system performance if transmission and reception parameters are not properly configured. For instance, repetitions can decrease the network throughput and scalability, making devices unnecessarily consume more energy. A study on how the network parameters could be set to reduce these negative effects is proposed in [84], where extensive ns-3 [93] simulations analyze the impact of many network configurations on the system performance when using both confirmed and unconfirmed traffic; however, only a smart city scenario with a single GW is considered. In [94], instead, Hoeller et al. propose time and antenna diversity to increase LoRa performance in terms of communication reliability. Repetition coding in particular is proven to be an effective choice, provided that SFs and the number of replicas are always chosen in order to avoid flooding the network with useless repetitions. Improving the underlying network infrastructure is another approach that can be used to achieve a higher reliability. In [46], for example, Van den Abeele et al. simulate in ns-3 a multi-GW LoRaWAN deployment with confirmed traffic in a smart city scenario. As expected, the availability of a multi-GW infrastructure brings several benefits to the network. In addition to the possibility of employing lower SFs in a dense deployment, a higher number of GWs makes it possible to share the load of downlink traffic, overcoming DC limitations and increasing the number of reply messages sent in RX1, which prevents sensor nodes from opening both receive windows.

The aforementioned works target an urban scenario, since the LoRaWAN technology was first proposed as a solution for IoT in large areas and open-air environments. Measurements of LoRa coverage range and SNR within an industrial plant are presented in [52]: here, Haxhibequiri et al. find that a single GW is sufficient to cover an area of 34000 m^2, and simulations based on the measured values estimate a network capacity of about 6000 nodes. The performance in terms of robustness to the noise and packet loss at the PHY layer is presented in [95], which analyzes the LoRa technology in an industrial setting, where machinery is deployed. The work proposed in [96] explores LoRa performance in different indoor settings, also providing a characterization of the channel. In [97], Neumann et al. present the results of an experimental campaign assessing the LoRaWAN performance in an indoor deployment with single-GW and only unconfirmed UL traffic, exploring different metrics, such as throughput, coverage and power consumption. Energy efficiency of LoRaWAN devices in an industrial scenario and the trade-off between transmission periodicity and battery replacement costs are also investigated in [98]. In [42], Luvisotto et al. compare the performance of LoRaWAN and IEEE 802.15.4 for an industrial scenario with unconfirmed traffic, proving the effectiveness of LoRaWAN.

Recent works have considered possible improvements of the LoRaWAN technology to address industrial requirements, such as latency and reliability, with new techniques. In [99], Rizzi et al. show that a scheme combining Time Division Multiple Access (TDMA) and frequency hopping is compatible with soft real-time applications, and that proper time, frequency and SF planning makes it possible to obtain the same performance of well-established technologies for industrial applications, such as WirelessHART and ISA100.11a, achieving a network capacity of 6000 EDs transmitting a packet every minute. Similarly, in [100], Leonardi et al. ameliorate the network performance by using cyclically scheduled super-frames split in intervals reserved for different types of communications. The authors of [101] consider the integration of a LoRaWAN deployment with 4G/5G networks for the IIoT, with a LoRa GW acting as a base station connected to the cellular network, and performing all the required LTE signaling, control and security procedures.

In the work presented in this section, instead, we investigate the performance of a LoRaWAN deployment, as defined in the specifications, and show how it performs in industrial scenarios when using multi-gateway deployments, confirmed traffic, Class C devices, and alternative channel plans. Different from previous works that suggest deeper changes to the medium access logic of the LoRaWAN protocol, all the solutions explored in this work can be implemented by simple configurations that can be applied also to already widely available devices, and can thus be more easily deployed and maintained.

5.1.2 Modeling of an industrial scenario

In this section, we describe in detail the scenario and the models we considered in our simulations. We use the ns-3 `lorawan` module, described in Chapter 3. Finally, we describe additional information regarding the LoRaWAN configurations.

The industrial scenario

We consider a LoRaWAN deployment in an industrial plant and assume that nodes are uniformly spread in a rectangular area with size 200x200 m^2, similar to the case considered in [52]. We assume EDs are wireless sensor nodes for process and asset monitoring, e.g., collecting measurements of temperature, pressure, or flow rates. We simulate a network composed of 300 EDs, i.e., one device every $\sim$133 m^2, that transmit periodic updates with a period $T \in \{30, 300000\}$ s, thus getting an application-layer offered traffic load of

$\lambda \in \{0.001, 10\}$ pkt/s. Note that this deployment is quite different from a typical smart city use case [16], where a GW serves thousands of devices over an area with a radius of several kilometers. We simulate networks with only unconfirmed or only confirmed traffic. In the latter case, the number m of maximum transmissions per packet is typically set to 1 or 4. Similarly, we consider the possibility of repeating the transmission of unconfirmed packets k times, as a measure to increase robustness of UL communication. This simple repetition coding increases the reliability and does not occupy the GW with DL ACK transmissions, which have been proved to be a bottleneck when the goal is to maximize the delivery of UL data [84, 46]. Application-layer packet size is set to 20 bytes for UL messages, while DL packets are assumed to carry only the ACK contained in the LoRaWAN header. When a single GW is simulated, it is placed on the middle point of one of the edges of the rectangle containing the EDs, while uniform random allocation within the limits of the rectangle is chosen for scenarios with multiple GWs. Given the short distance between the EDs and the GWs, all EDs are configured to employ SF 7. Similarly, RX2 SF is set to 7 to maximize the data rate. This choice is motivated by some preliminary tests which showed that, for the simulated values of the offered traffic λ, this is the most efficient solution.

Channel model

In this study, we assume that LoRaWAN works in the 863-870 MHz band, employing the minimum set of channels required by the specifications (see Tab. 2.7). The industrial wireless channel is particularly complex, and models need to take into account the presence of obstacles, signal reflections, delays, and intermittent noise caused by machines. In this work we account for three main effects typically present in this environment: (i) path loss and large-scale effects, which consider the signal attenuation caused by the distance between transmitter and receiver and by the presence of obstacles, (ii) small-scale effects, which consider reflections of the signal in the surrounding environment, and (iii) shot noise.

The total loss due to large-scale effects, $L(d)$, can be expressed in dB as the sum of path loss and shadowing, i.e.,

$$L(d)_{dB} = PL(d)_{dB} + \chi_{dB}. \tag{5.1}$$

The term χ_{dB} is usually modeled as a zero-mean Gaussian random variable with standard deviation σ, as defined in Tab. 5.1. For the path loss component,

Table 5.1.: Path loss coefficients for industrial propagation channel.

Reference	Scenario	d_0 [m]	L_0 [dB]	η	σ [dB]
[103]	Assembly Room	2	25	1.72	3.8
	Electronics Room	2	26	1.96	2.29
	Mechanical Room	2	26	1.79	5.07
[86]	LOS	15	57.67	2.25	5.65
	NLOS, lightClutter	15	64.42	1.94	4.97
	NLOS, heavyClutter	15	69.73	2.16	5.16

$PL(d)$, the loss is commonly computed according to the distance from the transmitter using a single-slope propagation loss model [86, 102]:

$$PL(d)_{dB} = \begin{cases} L_0 + 10\eta\log_{10}\left(\frac{d}{d_0}\right), & d > d_0 \\ L_0, & d \leq d_0 \end{cases}. \tag{5.2}$$

The estimation of the path loss coefficients in industrial scenarios has been the subject of many studies, such as [86, 102, 103]. In this study, we compare the system performance in multiple propagation scenarios, for which the path loss and shadowing terms have different values. In [103] test transmissions are performed from a height of 15 m, with the receiver placed near the ground, and the path loss coefficients are estimated for three rooms in a plant, which differ for the density of machines, the position of objects in the space and the material they are made of. We refer to the three scenarios modeled in this study as *Assembly Room*, *Electronics Room*, and *Mechanical Room*. In [86] the receiver is placed at a height of 6 m, while transmitting sensors are placed 2 m above the ground level. This work defines three additional propagation scenarios: line-of-sight (LOS) when a direct path exists between transmitter and receiver, and Light/Heavy Clutter when the height of the obstacles which obstruct the path are comparable to/much higher than the height of the sensor, respectively. The parameters for the selected models are listed in Tab. 5.1; in Section 5.1.3 we compare the behavior of the LoRaWAN system when using these models with the outcome of experimental measurements obtained from the literature [52], and motivate the choice of a model that will be used throughout the remainder of the section.

Small-scale effects account for the reflections of the transmitted signal in the propagation environment. These variations in the signal strength are modeled through a Nakagami-m distribution [104]. The value of the coefficient for the Nakagami-m distribution m_N is computed according to the channel model we implement. The channel model provided in [86] estimates temporal fading coefficients with log-normal distribution with mean 12.3 dB and standard deviation 5.4 dB. In the other cases, we used $m_N = 1.41$, as estimated in [105].[1]

Finally, we take into account the fact that an industrial environment may contain machinery creating additional background noise. In [106] and [107] the authors characterize the shot noise in an industrial setting, estimating the Cumulative Distribution Functions (CDFs) of the peak amplitude with respect to the noise floor, of the pulse duration and of the interval between consecutive pulses. The duration of the pulses turns out to be between tens of nanoseconds and a few microseconds, while the pulse spacing ranges from 100 ns to 204 μs. In our simulations, with a LoRa packet lasting tens of milliseconds, the introduction of an interference source with these features would lead to a high increase of the computational cost, which would be unnecessary given the interference model we consider (see Sec. 5.1.2). Although PHY level simulations showed that shot noise has no tangible impact on the demodulation performance of a single LoRa symbol, aggregate effects might still be present when an entire LoRa transmission is taken into consideration. Thus, here, we take a conservative approach, and account for the effect of shot noise by increasing the additional average noise power within a typical LoRa transmission, by a value of 3 dB based on the statistical description of the noise provided in [106].

Interference models

The literature features several approaches to model the interference in LoRaWAN networks. According to [32], the interference among the spreading factors can be modeled through a “co-channel rejection matrix,” where each element $e_{i,j}$ represents the power gap in dB that a signal transmitted with SF i must have to survive an interferer using SF j. As in [39], we assume that the interleaver will spread out the interferer’s energy on the entire received packet. Thus, if two received signals overlap for a certain time T_I, the interference power will be multiplied by the ratio T_I/T, where T is the duration of the desired signal, before being compared to the threshold. In the following, we will call this approach *Simple*, as it only employs this threshold matrix to decide whether a packet is lost because of interference.

[1] We recall that the coefficient m_N for the Nakagami propagation model can be obtained from the K-factor of Rician channel models as $m_N = (K^2 + 2K + 1)/(2K + 1)$.

A different solution is proposed in [46, 42]. Here, the received signal is divided into chunks with constant Signal to Interference plus Noise Ratio (SINR). The error rate of each chunk is computed by multiplying the length of the chunk by the BER corresponding to the SINR and SF of the signal, and the packet success probability is derived as the product of each individual chunk's success probability. Additionally, [42] leverages the two aforementioned approaches by combining Signal to Interference Ratio (SIR) and SNR in a measure called "equivalent SIR," for which the BER for different SFs are computed. In the following, this approach will be referred to as *Spectrum*, as it employs the `spectrum` framework provided by ns-3.

Finally, in this study we also consider the effect of OOB emissions. OOB is an undesired effect in radio systems that consists in the emission of power in frequency bands adjacent to the nominal transmission band. In particular, OOB emissions for LoRa are described and measured in [108]. Although the OOB power level is much lower than in the nominal band, the OOB emissions contribute to increasing the amount of interference, and this could be relevant in particular situations, e.g., when two devices are in close proximity, or when a strong signal coming from an ED in one channel interferes with a weak signal coming from a different ED operating in a neighboring channel. The impact of OOB can be exacerbated by the so-called Carrier Frequency Offset (CFO), i.e., the offset that affects low-cost frequency oscillators. The CFO can shift the carrier frequency up to 30 kHz. GWs, which employ more expensive circuitry, are designed to be robust to these impairments and can still correctly demodulate transmissions whose frequency error falls within a range of ± 30 kHz [24]. However, such an overlap may increase the effect of OOB. We hence introduce the *SpectrumOOB*, where out-of-band emissions contribute to the interference suffered by the desired signals. In this model, we add side-lobes to transmissions in the frequency domain, possibly spilling into adjacent channels. Similarly to the Spectrum interference model, the decrease in performance caused by this additional interference is then modeled through the "equivalent SIR" method employed in [42].

LoRaWAN settings

In the analysis here reported, we focus on the minimum frequency set mandated by the specifications for EU region (cfr. Tab. 2.7). In this analysis, we assume that EDs access the channel according to a pure ALOHA scheme and, hence, are subject to the duty cycle limits that, in Europe, are specified by the ETSI recommendations [109]. Note that, by using a combination of LBT and

Adaptive Frequency Agility (AFA) access schemes, the DC constraint can be relaxed to 2.8% for any 200 kHz spectrum [109]. A comparison between ALOHA and LBT+AFA access schemes in LPWANs is presented, e.g., in [110, 111, 112], where the authors generally conclude that LBT+AFA can indeed bring a (limited) performance gain over ALOHA, in particular in green-field scenarios (i.e., when all nodes use the same access method), even though [110] shows that ALOHA turns out to be more robust than LBT+AFA in mixed scenarios. In this study, we choose to focus on the ALOHA mode only, which is the chosen access scheme for LoRa in Europe.

In this study, we assume that all devices will use the same data rate, either SF 7 on a 125 kHz channel (DR 5) or SF 7 with a 250 kHz channel (DR 6) for UL communication. When devices employ SF 7 with a 125 kHz bandwidth, three channels at 868.1, 868.3 and 868.5 MHz are available for UL transmissions. Instead, when SF 7 is used with a 250 kHz bandwidth, all UL transmissions will use the same frequency at 868.3 MHz, foregoing the benefit of frequency orthogonality within the considered network deployment.

Note also that, as previously described in Sec. 2.4.3, for both Class A and Class C devices, a re-transmission is triggered if no reply is received within 2 s after the end of the transmission plus the time needed to detect a LoRa signal. This guarantees that the ACK delay performance of Class C devices is, at worst, equal to that of Class A devices within a time interval that extends to the end of the second receive window.

Performance metrics

For the network evaluation, we define the following metrics, with the aim of capturing the communication performance, and assessing the capabilities and effectiveness of a LoRaWAN deployment for sensing purposes in an industrial plant.

- The *Uplink Packet Delivery Ratio* measures the fraction of UL packets successfully delivered to the GW over those generated by the applications at the EDs. A success corresponds to the correct delivery of *at least* one copy of a UL packet. According to whether the UL traffic is carried by confirmed or unconfirmed packets, this metric is named *CU* or *UU*, respectively.

- The *delay* of a UL transmission is defined as the time between the packet generation and the moment it is successfully received by at least one GW and becomes then available to the NS.

- The *Confirmed Packet Success Rate (CPSR)* is the probability that both the UL packet and the corresponding ACK are correctly received by the NS and by the ED, respectively. This metric is meaningful only for confirmed traffic, and represents the capability of the system to communicate to the EDs, i.e., transmitting short control messages in response to UL updates or queries, or assessing the reliability of the connection, avoiding further re-transmissions.

- Finally, the *ACK delay* is defined as the time from the generation of a confirmed UL packet to the successful reception of the corresponding ACK at the ED.

5.1.3 Simulation results

This section describes the simulation results and analyzes the trade-offs involved in the design of a LoRaWAN solution for industrial applications. Given the large number of error and propagation models available in the literature, a first batch of simulation results explores their impact on the network performance in the considered scenario. Based on these considerations, an error and a propagation model are chosen for the rest of the simulations to investigate the higher-layer effects of the density of GW deployment, the class of device and the chosen channel plan.

Comparison of error and propagation models

Fig. 5.1 shows the simulation outcomes when the Simple, Spectrum and SpectrumOOB interference models are used to evaluate the CU and CPSR performance. When confirmed traffic and re-transmissions are considered, the different interference models yield quite similar performance. This suggests that the additional interference due to OOB emissions has limited impact on the MAC performance in the considered scenario. It should be noted that a slight increase in the SINR experienced by devices was observed in simulation, especially for higher values of λ. In these cases, however, the MAC level performance is largely dominated by direct collisions between transmissions overlapping in time and frequency. Similarly, the overall effect of OOB emissions at the network level when frequency shifts due to imprecise oscillators were included in the simulator was found to be negligible. These results motivate us to neglect the effects of OOB and imperfect oscillators, and to use the Spectrum interference model in the following analysis. This model is preferred to the SpectrumOOB model for its lower computational

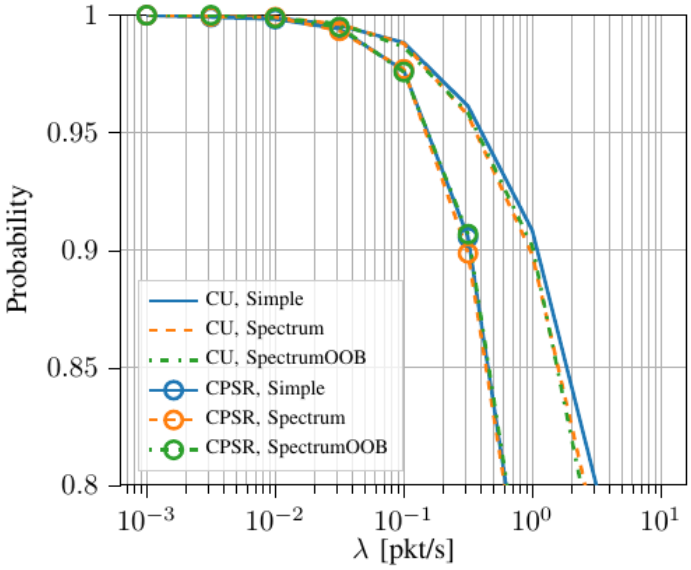


Figure 5.1.: Comparison of different interference models in terms of CU and CPSR, in a network served by a single GW.

complexity, while maintaining the effects of propagation environments that the Simple model ignores.

Fig. 5.2 shows the distribution of the simulated Received Signal Strength Indicator (RSSI) according to the different propagation models listed in Tab. 5.1. The figure shows that the models described in [103] and [86] provide quite different outputs. When we compare these results with the outcomes measured experimentally in [52], also shown in Fig. 5.2 for distances of 50 m and 150 m, it becomes apparent that the models in [86] provide better estimates of the experimental RSSI measurements. In particular, the samples collected in [52] at 50 m, which were almost always in LOS conditions, are closely represented by the LOS model, while measurements at over 150 m, in non line-of-sight (NLOS) conditions, can be matched using the Heavy Clutter propagation model. Based on this, the results presented in the remainder of this section have been obtained by considering a piecewise propagation loss model, in which the LOS parameters are used when the transmitter-receiver distance is lower than 100 m, and the Heavy Clutter model is used for distances over 100 m.

We also remark that, in addition to RSSI measurements, [52] provides SNR values. When comparing such values with those computed in our simulations, we found that the latter were consistently 20 dB above the experimental ones. The reason is that the SNR provided by LoRa chipsets saturates to around 12

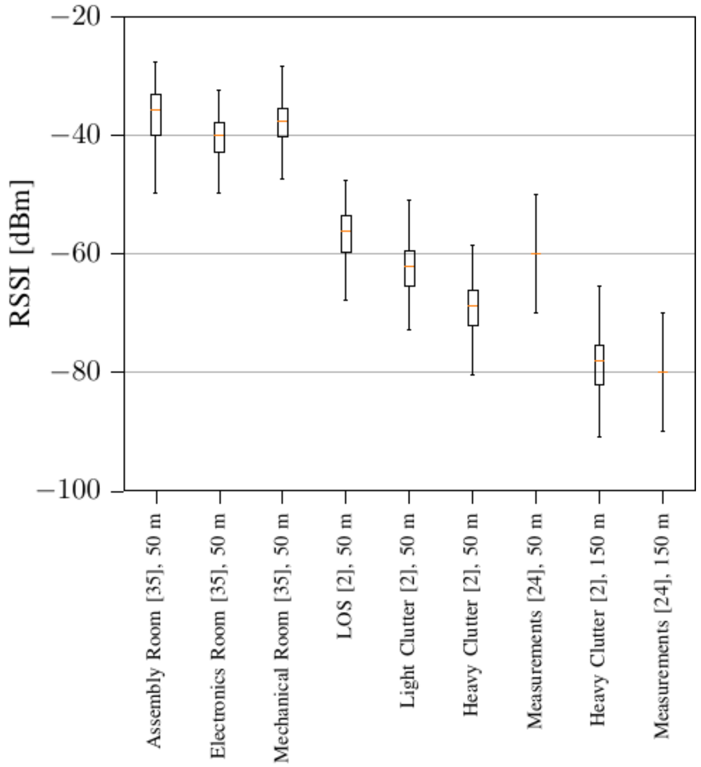


Figure 5.2.: RSSI computed from different propagation models and experimental measurements.

dB for RSSI values significantly above sensitivity, irrespective of the actual RSSI value [113].

Increasing the number of deployed GWs

In this section, we evaluate the network performance for different GW densities. The GWs are uniformly placed inside the same area of the EDs. Having multiple GWs improves the network performance by increasing the DL transmission capacity and the diversity in the wireless channels. The former advantage alleviates the effect of the duty cycle limitations that impose silent periods to the GWs: with multiple GWs, the probability that at least one is available to transmit a DL packet increases.

Fig. 5.3 shows that a larger number of GWs positively affects the ACK delivery rate for confirmed traffic (CPSR) even with $m = 1$. Although we have not reported the plots for space constraints, we also ran simulations for $m = 4$ and high traffic load, i.e., $\lambda = 10$ pck/s in this specific scenario, obtaining a CPSR of 99.7% with 10 GWs, and above 99.99% with 20 GWs. Very similar gains were observed for the UU and CU metrics.

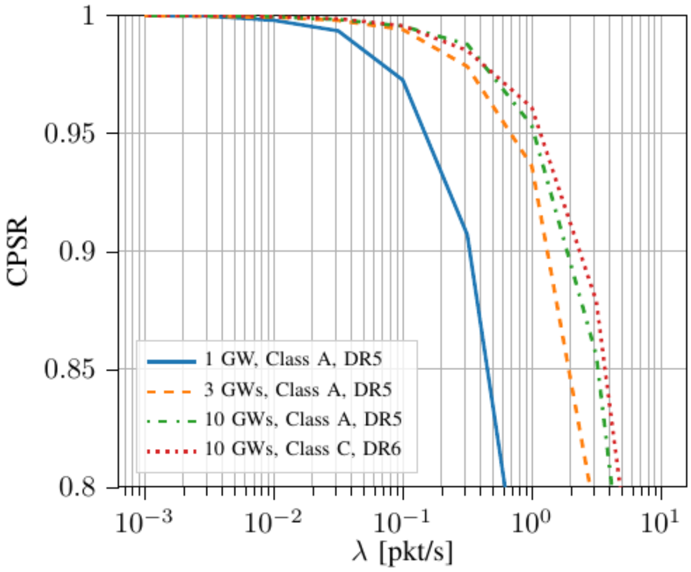


Figure 5.3.: Effect of gateway deployment density in terms of CPSR for $m = 1$.

Fig. 5.4 shows the Empirical CDFs of UL and ACK delays for confirmed traffic in a network served by 10 GWs, with $m = 4$ and $\lambda = 1$ pck/s. Despite the fact that, with these settings, the network is able to eventually deliver over 99.9% of its UL MAC-layer packets and 99% of its DL messages, the plot highlights the large delay due to duty cycle limitations and the receive window design for Class A devices, for which the minimum possible ACK delay is a little over 1 second. For a fraction of devices, the first receive window remains empty, and therefore they have to wait for a DL message in the second receive window. If this reception fails, the next re-transmission opportunity is available well over 9 seconds after the generation of the first UL message, due to duty cycle restrictions. In addition to this forced pause, in which the ED is in sleep mode, the standard requires an additional back-off, uniformly distributed between 1 and 3 seconds before the re-transmission attempt, further increasing the delay experienced by the EDs, which corresponds to the slope of the curve in Fig. 5.4. These results confirm that increasing the number of GWs can be a very effective strategy to increase the probability of correct message delivery in both the UL and DL directions, as predicted. Furthermore, coupling this setting with an increase of the re-transmissions is shown here to be a viable strategy to improve reliability in industrial scenarios.

Although increasing the GW density is a very simple and straightforward way to boost the reliability of the transmissions and to reduce the packet delay, placing GWs uniformly over the coverage area in industrial deployments might not always be possible, and companies might be forced to place GWs at the

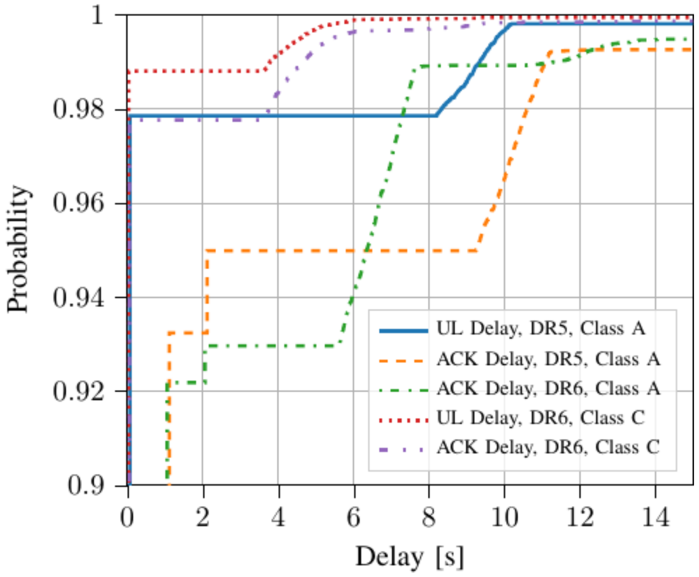


Figure 5.4.: Empirical CDFs for different delay metrics and network configurations.

edge of the facility, as done in [52]. This configuration was actually proven to be beneficial because of the increased channel diversity in [64].

Class C devices and DR6

In this section, we describe the simulation results obtained when Class C EDs and DR6 are employed in the network. Employing a bandwidth of 250 kHz halves the duration of the transmissions, yielding a very small gain in both the UL and ACK delay metrics with respect to the default configuration. However, this shorter time on air also cuts roughly by half the waiting times before re-tries caused by duty cycle restrictions, allowing the devices to re-transmit about 5 seconds after their initial attempt, instead of around 9 seconds when DR5 is employed. This advantage comes at the cost of additional interference, because all the EDs contend for the same frequency. Therefore, the application of this single-channel plan may not always be beneficial for all the devices. Indeed, in Fig. 5.4, the CDF of ACK delay for DR6-Class A setting is lower than for a DR5-Class A setting up to a delay of approximately 6.2 s. The combination of Class C and DR6, instead, provides very significant gains for both the delay metrics, reducing the minimum possible ACK delay from 1 s to the bare transmission time of a pair of UL and DL packets, which is in the order of a hundred milliseconds. This gain requires the LoRaWAN sensor node constantly attached to a main power source, or at least accepting a drastic reduction of battery lifetime. Nonetheless, this result proves that satisfactory delays both in the UL and in the DL are possible for over 99% of the devices in

LoRaWAN with the right combination of network settings, for an offered traffic of $\lambda = 1$ pck/s. Finally, the CPSR with DR6-Class C setting is more effective than doubling the number of GWs in the network, as can be seen from Fig. 5.3

Confirmed traffic and repetitions for reliability

In this section, we assess the effect on the UU metric of employing k "blind" repetitions of each UL packet in a network using unconfirmed traffic, as opposed to employing up to m transmissions per packet when using confirmed traffic. In the first case, we assume that the application always transmits each packet k times. In the second scenario, instead, re-transmissions are performed by the EDs only if no ACK is received. For these simulations we consider Class A devices with three channels to transmit. Furthermore, we set $\lambda = 10$ pck/s to better highlight the differences between the two approaches.

In both cases, we can observe that increasing the number of GWs has a positive effect on the UU performance, albeit with diminishing returns. With confirmed traffic, a sufficient number of GWs is necessary to have a stable network: if the NS is not able to promptly respond to every ACK request, UL traffic can quickly increase because of re-transmissions, further worsening the situation. The number of repetitions k, instead, is a critical parameter for the unconfirmed traffic case. In general, we can see that the blind repetition of UL messages yields better performance than relying on confirmed transmissions because the former can leverage the time diversity in the repetitions, without suffering from the GW unavailability due to DL transmissions.

Fig. 5.5 shows the difference between the unconfirmed and the confirmed solutions, for different values of $k = m$ and of the number of GWs serving the network. Especially for under-provisioned networks, the unconfirmed scenario provides significant gains with respect to the confirmed scenario, provided that the number of re-transmissions is well dimensioned. It should also be noted that, for a sufficiently large number of GWs, the difference between the two approaches becomes negligible, as the GWs can serve all devices, also when the traffic is increased because of repetitions.

Evaluation of communication service metrics

In this section, we evaluate the performance reached by LoRaWAN with respect to the service requirements for process and asset monitoring as defined in [114]. In particular, we consider sensors generating periodic measurements of a continuous value, such as temperature or pressure. In this case, the

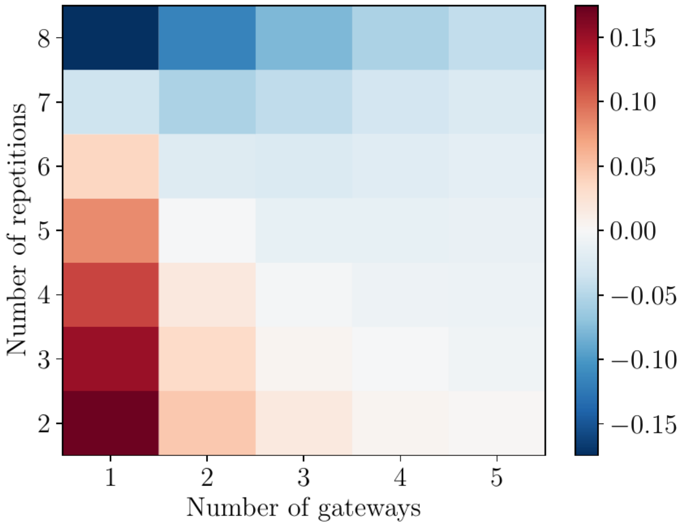


Figure 5.5.: Improvement in UU performance when using repetitions instead of confirmed traffic.

maximum allowed end-to-end latency is 100 ms, and the period between consecutive data transmissions is 60 s with uniformly distributed deviations of ±25% around the target value. When a packet delivery is not completed within the specified end-to-end latency, the system is said to experience a failure. Two metrics are considered:

- *Communication service reliability*: ability of the communication system to perform as required for a given time interval [114]. In our case, the reliability is quantified with the mean time between failures;

- *Communication service availability*: fraction of time during which the communication service operates without failures over the total amount of time the system is expected to deliver the service, in percentage.

Due to the heavy duty cycle limitation and the slow bit rate, only a single transmission can fit within the end-to-end latency requirement described above, considering that LoRa transmissions typically last in the order of tens of milliseconds. In this case, since SF orthogonality is not leveraged (only DR 5 and DR 6 are used, in order to limit the delay and channel occupation), and since the LoRaWAN MAC protocol targets applications with sporadic traffic, only a few tens of devices can be supported with a communication reliability of 10 minutes, and an availability over 90%. These results are obtained with the best configuration we identified, i.e., Class C devices and 10 GWs with 3 frequency channels.

If the application requirements are more lenient, with an end-to-end latency of 60 s (thus allowing up to $m = 8$ repetitions), the best configuration can instead obtain a service reliability of 12 hours, with a communication service reliability above 99.99%.

5.1.4 Conclusions

The requirements of IIoT are stricter than those of other IoT services: therefore, LoRaWAN networks should be carefully configured in an industrial setting. In this subsection, we assessed the capability of the LoRaWAN technology in such scenarios by answering some key questions about how to properly configure the network.

The first outcome concerns the choice of an appropriate channel model, which is validated through empirical measurements for LoRa devices in industrial settings found in the literature. Results showed that, for a high-level performance analysis, the Spectrum interference model provides a good balance between accuracy and computational complexity. Secondly, we showed that the increase of the GW density can improve all the metrics, and reduce the negative effect of duty cycle limitations: this, in turn, ameliorates the capability of transmitting DL messages, which brings benefits in terms of both ACK delay and reliability. Indeed, with a proper deployment and parameters configuration, and considering more lenient but reasonable application requirements, we have shown that it is possible to achieve a delivery probability of over 99%, both in the UL-only and in the bi-directional communication. The frequency plan was a third point of investigation: using DR6 (i.e., employing a wider channel instead of three separate narrower channels) brings a slight degradation in the delay performance for the first transmission attempt, but improves it when more transmissions are needed, since the waiting time imposed by the duty cycle and the collision probability are both reduced. The combination of this frequency plan with the usage of Class C devices makes it possible to further reduce the delays for both UL and DL reception. Furthermore, we also demonstrated that, when a few GWs are deployed, the usage of blind repetitions provides better performance than using confirmed traffic in terms of UL message delivery.

A LoRaWAN deployment can, therefore, meet the requirements of monitoring IIoT applications, if combined with a mindful choice of the devices and the network topology as well as a careful tuning of the parameters. Network designers must consider both the specific application requirements and the metrics of interest in order to devise a network deployment that satisfies the industrial needs without over-provisioning. However, if the envisaged applications dictate strict requirements, as presented in Sec. 5.1.3, LoRaWAN still struggles to achieve such performance.

In the future, this work can be extended with the inclusion of Class B devices, which trade energy efficiency for latency of DL messages. This, combined with the introduction of longer data packets in the DL direction, would address those industrial applications that require periodic communication with the sensor nodes. Another interesting possibility of extension is the simulation of an outdoor industrial scenario, such as an industrial harbor, where the larger distances force nodes to use different SFs, which is advantageous because of the quasi-orthogonality between SFs but, at the same time, could require further analysis for the choice of optimal channel plan and GW positioning.

5.2 Remote tracking of UAV swarms

Over the last few years, UAVs have entered the mainstream: the commercialization of low-cost drones for amateur and professional use is quickly increasing the number of flying units, which will soon be measured in millions, according to the U.S. Federal Aviation Administration (FAA)[2]. Their integration in cellular networks, both as end-users and as coverage extension devices [115], is already being discussed, and 5G systems are expected to make use of UAVs of different sizes, from small-scale low-altitude drones to communication satellites [116]. Although energy and battery concerns are still critical [117], the use of UAVs is being proposed for several kinds of scenarios, from remote infrastructure monitoring [118] to disaster monitoring [119] and relief [120].

The popularity of UAVs has grown exponentially over the past few years, and their widespread use could enable a real Internet of Flying Robots [121] in the near future. Drones are used for environmental monitoring in a wide range of scenarios, from traffic jam detection [122] to industry and agriculture [123], and are posed to become a key Smart City infrastructure [124]. UAVs are also being used in combination with ground-based robots to help them perform complex tasks [125]. However, disaster management and relief is perhaps the most interesting application for UAVs: drones can easily avoid ground-level obstacles and flooded areas by flying over them, surveying the extent of the damage [119] or helping with search and rescue operations [120] and communications.

[2] FAA Aerospace Forecast, Fiscal Years 2019-2039: `https://www.faa.gov/news/updates/?newsId=93646`

As the capabilities of UAVs evolve towards the full support of safety-critical civilian applications, as well as military battlefield support operations [126], accurate positioning of drones is going to become more and more important. UAVs often have on-board Global Positioning System (GPS) receivers, and filtering [127] and data fusion techniques, often integrating camera image processing [128], can significantly improve the positioning accuracy by combining several measurements into a single solution that is more robust and precise than any individual approach. However, in order to track or control a swarm of several UAVs, mission control must be able to follow and even anticipate the drones' trajectories. This requires the UAVs to frequently report their position to mission control, often with no available infrastructure and over a range of several kilometers.

The required accuracy of positioning algorithms largely depends on the application: wildlife monitoring [129], mostly concerned with counting animals, can accept a wide error as long as the UAVs register images of the packs or swarms, while urban pollution monitoring [130], precision agriculture [131], and hydrogeological monitoring [132] can require precision between 10 m and 50 m, since they usually involve the reconstruction of 2D or 3D models from multiple images, and the accuracy of the position of the point of view can determine the quality of the final reconstruction. Finally, formation control usually requires precision well below 10 m (and often even below 1 m [133]): in this case, centralized solutions that communicate over a range of several kilometers are suboptimal, and state of the art solutions use distributed optimization with low-range and high-capacity communication technologies.

In this section, we propose a system that can remotely track the position of a swarm of up to 50 UAVs, moving independently in a three-dimensional environment. Our system is based on the LoRaWAN communication standard, which is used by the UAVs to transmit short updates over a range of several kilometers. This allows for the deployment of the drones over wide areas with minimal infrastructure, enabling their use in remote or disaster-struck regions. Due to the low bitrate and limited duty cycle imposed by the LoRaWAN specifications in some regions (including Europe), tracking updates cannot be extremely frequent, so our system uses predictive Kalman-based filtering to track UAVs between updates, using a novel 3D motion model that we developed, called 3D-CTRA. Our model extends the well-known Constant Turn Rate and Acceleration (CTRA) model, widely used in vehicular scenarios, by adding a third dimension which allows it to represent even complex banking maneuvers accurately. We also study a simpler model, named CTRA+, which considers linear motion on the vertical axis. In both cases, the tracking mechanism is the same: each UAV periodically transmits its state, including the orientation, speed and acceleration, and the control station evolves the motion model to estimate the drone's position. In this way, even sporadic updates allow the system to track the UAV.

As stated above, the management of swarms of drones presents some additional challenges, as LoRaWAN uses random access and updates from different UAVs can interfere. By exploiting the orthogonality of different SFs [52], we can reduce interference. However, this comes at a price: increasing the SF, the range of communication increases, but the time intervals between transmissions required to respect the duty cycle constraint can last several seconds, thus making the tracking more difficult.

This section analyzes the performance of such a system with different communication parameters, e.g., when using a single channel with a bandwidth of 250 kHz instead of three 125 kHz channels, as well as with different ADR policies. The joint optimization of the motion model and communication system presents some interesting trade-offs, which we explore in extensive ns-3 simulations using the UAV mobility traces from the Mid-Air public dataset. In our simulations, we compare the two mobility models we proposed against a baseline Constant Speed (CS) motion model, in which the tracked position evolves according to a uniform rectilinear motion. The results show that an appropriate setting of the communication parameters and tracking model can give significant performance gains, enabling accurate tracking over larger swarms and longer distances.

In particular, the contributions of the analysis presented in this section can be summarized as follows:

- We defined two mobility models in 3D space, whose adherence to real drones' trajectories was verified experimentally by using a public dataset. We also derived the motion equations and designed a Kalman-based solution to track the UAVs' positions over time;

- We defined a tracking system based on LoRaWAN, considering different frequency allocation schemes and ADR options in order to support swarms of different sizes and in different scenarios. While LoRaWAN has been studied for massive scenarios with thousands of sensors transmitting infrequently, we consider a scenario where fewer nodes transmit at the maximum duty cycle, showing interesting communication trade-offs;

- We considered the effect of sending payload data as well as the tracking information through the LoRaWAN link, effectively piggybacking sensor readings;
- We ran extensive ns-3 simulations in several scenarios, considering communications-related metrics as well as directly measuring the tracking error of the system.

A preliminary and partial version of the analysis presented in this section appeared in the proceedings of the 2020 IEEE INFOCOM workshops [134], which presented the basic features of our tracking system for a single UAV. The analysis here reported significantly expands the conference version by analyzing the communication system thoroughly and considering the effects of interference when the tracking system is extended to swarms of UAVs instead of a single one.

The rest of the section is organized as follows. Sec. 5.2.1 presents the state of the art on UAV applications, focusing on the tracking frameworks that can be used to estimate drone trajectories. Sec. 5.2.2 presents the CTRA+ and 3D-CTRA models, including the relative update equations, while Sec. 5.2.3 describes the LoRaWAN standard and the frequency plan needed for our application. The simulation settings, including the different ADR solutions we implemented, and the results are described in Sec. 5.2.4, while Sec. 5.2.5 presents our concluding remarks and ideas for future work.

5.2.1 Related works

The target tracking problem is a well-studied research topic, and is usually solved by representing the target's motion using simple models and estimating its position with a Bayesian Filtering (BF) algorithm. The best-known BF algorithms used in this context are the Kalman Filter (KF) [135] and the Particle Filter (PF) [136]. Long-term forecasting can be achieved by simply applying the predictive step of the BF to the last available state estimate. However, this solution does not provide good performance when updates are infrequent, especially if the model is inaccurate. In this perspective, our work tries to minimize broadcasting operations while ensuring accurate position estimation.

The tracking problem has been widely explored in 2D vehicular scenarios [137], often using the CTRA model [138], which considers an accelerating vehicle with constant turn rate. A similar model for drones moving horizontally was presented in [139], including Gaussian noise on the motion parameters. A more complex model with several possible maneuvers was described in [140], adapting the CTRA settings to draw the correct trajectory. In general, motion models for drones are based on 2D CTRA or simpler models with constant speed [141] or orientation [142]. To the best of our knowledge, CTRA+ and 3D-CTRA are the first models that can represent 3D maneuvers with the same flexibility that CTRA has in the 2D space.

It is also possible to track passive UAVs, i.e., find their position without any communication, although at a much shorter range. In [143], a visual system is shown to detect and track a UAV with high accuracy for a distance of up to 200 m. This method has the advantage of having no communication overhead, but is inherently limited by the capabilities of the visual system, and will only work if the UAVs are in line of sight. It is also possible to use the degree of arrival of external communication signals to allow the swarm to track the relative position of its members by acting like an antenna array [144]: if the UAVs share channel information and invert the channel, they can derive their relative positions. However, this method only works for a high SNR, and so it inherently limits the area that can be covered before the swarm loses formation control. The same operation can be performed using Terahertz communications [145], although only for very short distances (below 10 m in the paper).

Although LoRaWAN is a communication technology designed for IoT applications [14], we tested its performance in this specific use case to verify if it could adequately support control communication. To extend LoRaWAN capabilities in the scenario of multiple UAVs in the same network, we leveraged the ADR mechanism provided by the standard, as described in Sec. 2.4.4 and Sec. 5.2.4. Several different implementations of this mechanism can be found in the literature. In [74], the authors propose a mechanism based on a waterfilling algorithm: by adequately setting the modulation parameters, it is possible to equalize the time on air (ToA) of the packets transmitted by the system's devices, increasing the maximum network throughput. However, our application would not benefit from this solution, since, as better described in the following sections, devices using low bit rates would also transmit their packets very sporadically, leading to poor tracking performance. ADR applied to moving devices is studied in different works, as [79, 57, 146]. In particular, in [79] the authors based the ADR implementation on the estimated RSSI of the device, setting the modulation parameters accordingly. However, they assume to know mobility patterns, which are directly related to the value of the RSSI. Although this approach is useful for devices always moving on the same trajectory (or with only slight variations), it is not applicable to the case of drones exploring an unseen area, or moving towards new objectives. In [146], ADR algorithms employing a Gaussian and an Exponential Moving Average filter are explored with ns-3 simulation, in a scenario of static or slowly moving devices. In this scenario, however, the devices transmit only one packet per hour, which would not be sufficient in a tracking application. Empirical measurements on the performance of the ADR algorithm implemented by The Things Network (TTN) are presented in [57]. In this analysis, devices are placed on trucks moving in the center of a city, which may be different from the UAVs use case for the application requirements, the characteristic of the communication channel and the mobility patterns.

5.2.2 Tracking system model

In this work, we suppose that the UAVs in a swarm periodically transmit their state to a control station using the LoRaWAN communication standard. The aim of the control station is to track the UAVs' positions in different scenarios. To represent the drone motion in a 3D environment, we consider three possible models, i.e., CS, CTRA+, and 3D-CTRA. In the rest of the section, we recall the CS and CTRA models, and we extend them to obtain the system equations for CTRA+ and 3D-CTRA. We also describe the Unscented Kalman Filter (UKF) tracking system used by the control station to track the UAVs' trajectories. In the following, we refer to a UAV's *position* as the coordinates (x, y, z) representing its position in space using a Cartesian coordinate system, while its *orientation* is represented by the three angles (θ, ϕ, γ), which measure the rotation with respect to a horizontal, North-facing position.

The CS, CTRA and CTRA+ models

While conventional systems only track the yaw, i.e., the angle θ between the drone's orientation and the reference direction on the horizontal plane, 3D motion models must also consider the pitch, i.e., the vertical angle ϕ between the drone's orientation and the horizon. Moreover, the target state must include the altitude z as well as the horizontal position (x, y), resulting in the 5-tuple (x, y, z, θ, ϕ). These parameters are common to all the motion models we implement. However, none of our models explicitly considers the orientation on the roll axis: while rotorcraft UAVs typically use motion on the roll axis to turn, we can model the three-dimensional drone as a simple cylinder, which is not affected by rotations on the roll axis, making the representation more compact. Naturally, this does not correctly track the precise orientation of the UAV, but we are only interested in its effects on the future position in space of the drone, for which rotations on the roll axis are irrelevant. Furthermore, the models do not consider the curvature of the Earth, but a perfectly horizontal plane, which results in an error $\Delta h(d)$ when measuring the altitude of an object:

$$\Delta h(d) = R_E \left(1 - \cos\left(\frac{d}{R_E}\right)\right), \tag{5.3}$$

where $R_E = 6.371 \times 10^6$ m is the average radius of the Earth. This error is negligible for distances below 1 km, as $\Delta h(1 \text{ km}) = 7.8$ cm. As we expect the UAVs to cover distances much lower than 1 km between subsequent updates, which come at a rate of 1 Hz, this error can be compensated for by GPS measurements, which do account for the Earth's curvature, and filtered out. If we were to consider much faster UAVs or less frequent sensor updates, the curvature of the Earth would become a significant factor.

The CS model assumes that the target moves with a constant speed v, without any change of direction (see Fig. 5.6). This implies that the turn rate $\omega = \frac{d\theta}{dt}$ and the tilt rate $\psi = \frac{d\phi}{dt}$ are zero, while the model state is simply given by

$$\mathbf{x}_{\text{CS}}(t) = [x(t)\ y(t)\ z(t)\ \theta\ \phi\ v\ ]^T. \tag{5.4}$$

Two-dimensional CTRA revises this hypothesis by assuming the target to have a constant tangential acceleration $a = \frac{dv}{dt}$ and a non-zero turn rate ω on a flat plane. The CTRA+ model makes the same assumption, but the plane on which the UAV moves is rotated by a constant pitch ϕ. As above, the tilt rate ψ is zero:

$$v(t) = v(0) + at; \tag{5.5}$$

$$\theta(t) = \theta(0) + \omega t; \tag{5.6}$$

$$\phi(t) = \phi(0), \tag{5.7}$$

where $v(0)$, $\theta(0)$ and $\phi(0)$ represent the initial velocity and orientation of the target. Practically, CTRA+ can be described by an Archimedean spiral [147] that evolves on a plane tilted by an angle ϕ with respect to the horizon. A sketch of the resulting model motion is given in Fig. 5.7.

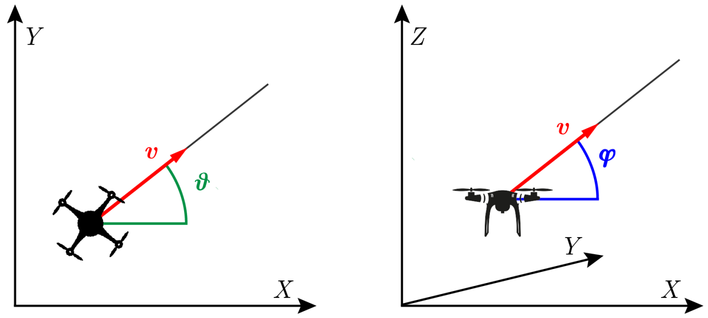


Figure 5.6.: The CS motion model.

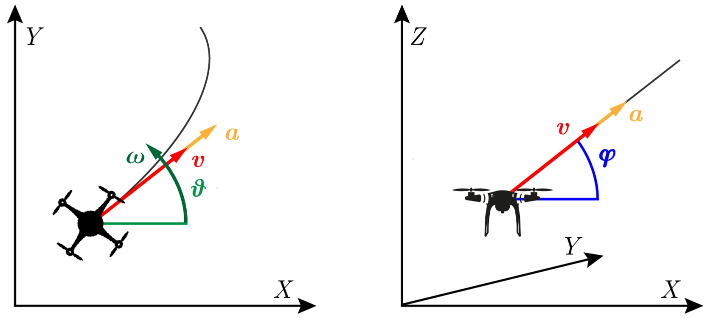


Figure 5.7.: The CTRA+ motion model.

To compute the equations of the CTRA+ model, we first need to separate the components of the target's velocity vector $\mathbf{v}(t)$:

$$v_x(t) = \frac{dx}{dt} = v(t)\cos(\theta(t))\cos(\phi(t)); \tag{5.8}$$

$$v_y(t) = \frac{dy}{dt} = v(t)\sin(\theta(t))\cos(\phi(t)); \tag{5.9}$$

$$v_z(t) = \frac{dz}{dt} = v(t)\sin(\phi(t)). \tag{5.10}$$

Therefore, the velocity $v(t)$ can be computed as $v(t) = \sqrt{(v_x(t))^2 + (v_y(t))^2 + (v_z(t))^2}$, while the target position is given by the integral of the velocity components over time:

$$x(t) = x(0) + \int_0^t v(\tau)\cos(\theta(\tau))\cos(\phi)d\tau; \tag{5.11}$$

$$y(t) = y(0) + \int_0^t v(\tau)\sin(\theta(\tau))\cos(\phi)d\tau; \tag{5.12}$$

$$z(t) = z(0) + \int_0^t v(\tau)\sin(\phi)d\tau. \tag{5.13}$$

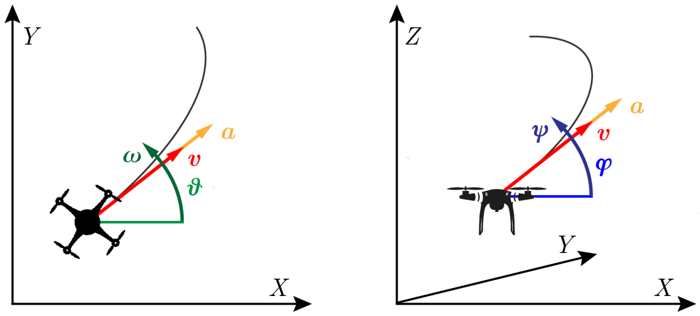


Figure 5.8.: The 3D-CTRA motion model.

After solving the above equations, we get the following results:

$$x(t) = x(0) + \cos(\phi)\Bigg[a\left(\frac{\cos(\theta(t)) - \cos(\theta(0))}{\omega} + t\sin(\theta(t))\right) + v(0)\Big(\sin((\theta(t)) - \sin(\theta(0))\Big)\Bigg]; \tag{5.14}$$

$$y(t) = x(0) + \cos(\phi)\Bigg[a\left(\frac{\sin(\theta(t)) - \sin(\theta(0))}{\omega} - t\cos(\theta(t))\right) - v(0)\Big(\cos(\theta(t)) - \cos(\theta(0))\Big)\Bigg]; \tag{5.15}$$

$$z(t) = z(0) + \sin(\phi)\left(v(t)t - \frac{at^2}{2}\right). \tag{5.16}$$

We note that the procedure is equivalent to 2D CTRA [138] for the x and y components, except for the constant multiplying factor $\cos(\phi)$. Hence, the CTRA+ state is given by:

$$\mathbf{x}_{\text{CTRA+}}(t) = [x(t)\ y(t)\ z(t)\ \theta(t)\ \phi\ v(t)\ a\ \omega]^T\,, \tag{5.17}$$

which corresponds to the tuple representing the current attitude, with the addition of the velocity v, the acceleration a, and the turn rate ω.

The 3D-CTRA model

The 3D-CTRA model extends the above description by adding a constant tilt rate $\psi = \frac{d\phi}{dt}$. Consequently, the target's movement is represented as the combination of two independent spiraling motions on the horizontal and vertical planes, forming a curved helix. We represent the resulting trajectory in Fig. 5.8

While the evolution of $\theta(t)$ still follows (5.6), the pitch is given by:

$$\phi(t) = \phi(0) + \psi t. \tag{5.18}$$

This complicates the derivation of the motion equations considerably, since $\phi(t)$ is now time-dependent. For the sake of simplicity, we report the procedure only for $x(t)$, which is given by the integral of $v_x(t)$ over time. Applying the Werner formula, we obtain

$$x(t) = x(0) + \int_0^t \frac{v(\tau)}{2}(\cos(\theta(\tau)+\phi(\tau)) + \cos(\theta(\tau)-\phi(\tau)))d\tau, \tag{5.19}$$

which can be solved in closed form. The derivations for $y(t)$ and $z(t)$ follow the same steps; the final results are given in (5.20)-(5.22), where we used the compact notation $[F(x)]_a^b = F(b) - F(a)$ to indicate that the primitive function $F(x)$ should be evaluated at the extremes a and b.

$$\begin{aligned} x(t) = x(0) + \Bigg[&\frac{v(\tau)}{2}\left(\frac{\sin(\theta(\tau)+\phi(\tau))}{(\omega+\psi)} + \frac{\sin(\theta(\tau)-\phi(\tau))}{(\omega-\psi)}\right) \\ &+\frac{a}{2}\left(\frac{\cos(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)^2} + \frac{\cos(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)^2}\right)\Bigg]_0^t; \end{aligned} \tag{5.20}$$

$$\begin{aligned} y(t) = y(0) + \Bigg[&-\frac{v(\tau)\cos(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)} - \frac{v(\tau)\cos(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)} \\ &+\frac{a\sin(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)^2} + \frac{a\sin(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)^2}\Bigg]_0^t; \end{aligned} \tag{5.21}$$

$$z(t) = z(0) + \left[-v(\tau)\frac{\cos(\phi(\tau))}{\psi} + a\frac{\sin(\phi(\tau))}{\psi^2}\right]_0^t. \tag{5.22}$$

The equations (5.5), (5.6), (5.18) and (5.20)-(5.22) define the full non-linear version of 3D-CTRA. In the special cases in which $|\omega| = |\phi|$, we need to substitute some terms in the derivation, as the standard equations are undefined: we deal with this case in the Appendix. In general, the 3D-CTRA state is given by:

$$\mathbf{x}_{\text{3D-CTRA}}(t) = [x(t)\ y(t)\ z(t)\ \theta(t)\ \phi(t)\ v(t)\ a\ \omega\ \psi]^T, \tag{5.23}$$

which is equivalent to the CTRA+ state as defined by (5.17), with the addition of the tilt rate ψ.

We remind the reader that 3D-CTRA considers constant values for both ω and ψ. This does not reflect the real behavior of an aircraft, as dives and climbs are usually relatively short. To make the model more realistic, the tracking system reduces the value of ψ by a factor η after every prediction step. In other words, the model implicitly assumes that the drone will gradually reduce its tilt rate and stabilize its pitch until it receives an explicit update from the UAV.

Remote tracking

As in [138], the tracking process is implemented by following a Bayesian approach. Hence, the state $\mathbf{x}$ of each UAV is modeled according to:

$$\begin{cases} \mathbf{x}(t+T) = f(\mathbf{x}(t))) + \zeta(t), \\ \mathbf{o}(t) = h(\mathbf{x}(t))) + \eta(t). \end{cases} \tag{5.24}$$

In (5.24), the first equation describes the evolution of $\mathbf{x}(t)$, while the second describes the relation between $\mathbf{x}(t)$ and the state observation $\mathbf{o}(t)$. Particularly, $f(\cdot)$ depends on the considered motion model $\mathcal{F} \in \{\text{CS}, \text{ CTRA+}, \text{ 3D-CTRA}\}$, while $h(\cdot)$ depends on the measurement system. Besides, $\zeta(t)$ and $\eta(t)$ are the process and measurement noises at time t, and are modeled as independent Gaussian processes with zero mean and covariance matrices Q and R, respectively.

In this work, we exploit an Unscented Kalman Filter (UKF) with the *Van Der Merwe* parameterization [148] to estimate the drone state. The UKF algorithm is an extended version of the original KF able to model the non-linear evolution of the drone state. In particular, the system information is summarized by a set of particles, named *sigma points*, which are processed through functions $f(\cdot)$ and $h(\cdot)$ at each step of the algorithm. Each particle is associated with a weight that decreases as its distance from the state estimation is larger. Hence, the UAV's estimated state can be computed as a weighted average of the sigma points used by the filter. The operation of the filter is described in Appendix B.2, with a detailed mathematical definition.

In our model, we consider that each UAV implements a UKF to estimate its own state, and the control station implements an additional UKF to estimate the state of each UAV in the system. In particular, the UAVs are provided with onboard sensors that provide an update of their state estimation in each timeslot. Instead, the control station's UKF has no input but the information received from the UAV. We adopt a periodic broadcasting strategy [137]: after a randomized time at the beginning of the simulation, each UAV sends the estimate of its own state to the control station with a constant inter-transmission period. A small delay is also added between consecutive transmissions, in order to mitigate the interference suffered by UAVs whose transmissions happen to be synchronized. After it receives an update, the control station updates its UKF with the new information and exploits the predictive step to forecast the UAV's trajectory. Naturally, the errors will compound, causing long-term predictions to become less and less accurate until the next update. In order to enable the UAV to send the UKF parameters even at great distances, we considered the LoRaWAN technology [26], which is better explained in the following section.

5.2.3 Communication system model

In the scenario we consider, we assume that the drone is equipped with a LoRaWAN Class A ED. This class of devices is designed to consume a minimum amount of energy, with nodes staying in sleep mode most of the time, transmitting when necessary, and waking up for reception in two short windows after each transmission. This feature makes the energy consumption of the LoRaWAN system negligible for the UAVs, considering that even the lightest quadcopter requires 50 to 100 W to hover or fly at low speed when carrying no sensor payload. In our case, we can expect a power consumption to fly in the order of hundreds of Watts, 3 to 4 orders of magnitude higher than the communication .

The LoRaWAN protocol also supports the ADR mechanism, through which the NS can control the transmission parameters of the EDs, that we will leverage in the following analysis.

In Europe, LoRaWAN works in the unlicensed 868 MHz sub-band, which is subject to duty cycle regulations. In particular, three 125 kHz channels are allocated to UL transmissions, and must respect a duty cycle limitation of 1%. Another option that we consider in this study is to use the available frequency band to allocate a single 250 kHz channel, which does not bring the benefits of frequency orthogonality but reduces the packet transmission time, and is thus always preferable in case of a system with a single drone.

To maintain a good tracking performance, the packet generation (and transmission) frequency should be very high. However, when combined with multiple transmitting devices, as in the scenario of a swarm of drones, this rapidly increases the traffic injected in the communication channel, and the system performance suffers from interference.

This scenario is highly different from the most common LoRaWAN applications, which involve thousands of sensors transmitting sporadically. The massive number of devices allows network designers to approximate the system as an infinite-user ALOHA system, but this is not possible in our case. In our scenario, a limited number of UAVs is transmitting periodically at the maximum allowed duty cycle of 1%: this case is more similar to the finite-user case, which has been mostly studied for slotted systems [149, 150]. Furthermore, what is interesting in our case is not just the throughput and delay [151], as in most previous studies, but also the Age of Information (AoI) [152], and it is not just the AoI that matters, but its effect on the tracking performance. Whether or not LoRaWAN is suitable to sustain this type of IoT applications is unclear. In this section, we show that, under certain conditions, it can indeed sustain both control and data traffic, but a correct configuration of the system parameters is crucial: a straightforward solution is to leverage orthogonality, both in the frequency domain (i.e., using three channels instead of a single one) and by employing different SFs. However, these choices can also have side effects: the increased packet transmission time due to a narrower channel and/or the use of higher SFs will reduce the number of packets that can be transmitted in compliance with the duty cycle regulations. These trade-offs will be part of the analysis in Sec. 5.2.4.

To partially contain the problem posed by the duty cycle limitation, we compress the system state to reduce the inter-transmission time and improve the tracking performance. In order to minimize the payload size, we can represent the position using 2 bytes, allowing movement in a square box with a size of 13 km while limiting the quantization error to 10 cm, significantly less than the average GPS error. Angles and turn rates can be represented using just 1 byte, with a maximum error of 0.7 degrees. Since velocity and acceleration are limited, they can also be represented with just 1 byte, with a negligible loss of precision. Considering that the CS model requires the knowledge of the attitude 5-tuple and the velocity, its minimum payload size is 9 bytes. The CTRA+ state as given in (5.17) requires 11 bytes, and the 3D-CTRA state as given in (5.23) requires 12 bytes. The different payload formats are reported in Fig. 5.9. The LoRa transmission times for packets with these lengths are reported in Tab. 5.2: to respect the duty cycle limitation, packets can be sent only sporadically, with a transmission period in the order of a few seconds.

5.2.4 Simulation settings and results

In this section we describe the dataset providing the trajectory of a drone, how this has been used to represent the simulation scenario and how simulation parameters have been configured. Then, we present the results obtained, discussing first the scenario of a single drone, and then the scenario where multiple drones are involved, highlighting how the best settings differ for the two cases.

Table 5.2.: Transmission times for packets with different payload size and minimum transmission interval to respect duty cycle regulations. Transmission times are computed as in [28], considering an additional MAC header of 9 B.

SF	B (kHz)	Packet payload (B)	TX time (s)	Min TX interval (s)
7	125	9	0.0515	5.15
		11, 12	0.0566	5.66
	250	9, 11	0.0257	2.57
		12	0.0283	2.83
8	125	9	0.0927	9.26
		11, 12	0.1029	10.29
	250	9	0.0463	4.63
		11, 12	0.0514	5.14

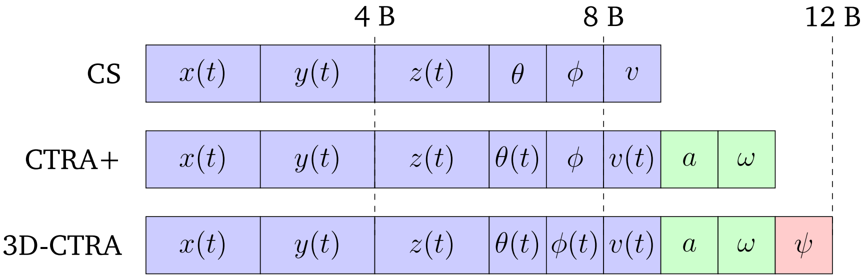


Figure 5.9.: Schematic of the payload format for the three tracking schemes.

Settings

To simulate the UAV mobility, we exploit the Mid-Air dataset [153], which contains the flying records of a quad-copter moving in 3 different virtual environments. Such data are divided into 54 trajectories of equal length, for a total of 79 minutes of flight time. To strengthen our analysis, we randomly combined the different trajectories, thus increasing the duration and the variability of the drone traces. These data are used to represent the ground-truth motion of the UAVs, while we synthetically generated noisy data to represent the information acquired with the drones' sensors. The sensor data included the position, the attitude, and the velocity and acceleration vectors of the UAV, combining GPS, accelerometer and gyroscope.

The process noise of the tracking system is described by the matrix $Q = qI$, where I represents the identity matrix and $q = 0.01$. Instead, the error affecting the drone measurements is given by a diagonal matrix R, whose elements represent the accuracy of the various drone sensors. The noise matrices and the UKF parameters are reported in Tab. 5.3. In particular, values of R were

Table 5.3.: Tracking system parameters.

Parameter	Value	Description
R_x	0.8274 m^2	Position accuracy along x
R_y	0.8274 m^2	Position accuracy along y
R_z	3.7481 m^2	Position accuracy along z
R_v	0.2500 (m/s)^2	Speed accuracy
R_a	0.1521 (m/s^2)^2	Acceleration accuracy
R_θ	0.0085 rad^2	Yaw accuracy
R_ϕ	0.0085 rad^2	Pitch accuracy
R_ω	0.0003 (rad/s)^2	Turn rate accuracy
R_ψ	0.0003 (rad/s)^2	Tilt rate accuracy
q	0.01	Process noise
η	0.9	Tilt reduction parameter

chosen according to [154, 155, 156]. We highlight that the UKF setting, e.g., the state dimension, changes according to the chosen motion model. As already stated, the UKF at the control station is used to estimate the target trajectory by exploiting only the predictive step. This implies that, when a new update is received, the filter state is substituted with the new information, and the estimation process starts again.

The scenario of interest was studied with the network simulator ns-3 using the `lorawan` module described in Chapter 3, with N_d drones moving in the space according to the mobility traces of [153]. The drones are assumed to be equipped with a LoRaWAN interface, which transmits packets at the maximum frequency allowed by the duty cycle. These messages are collected by a GW and forwarded to the NS. Unless otherwise stated, transmitted packets do not require any acknowledgment, and the NS does not control any of the communication parameters. For each packet, we recorded whether or not it was successfully received, according to a log-distance propagation loss model that also considers shadowing and to the interference model that determines the correct reception probability in case of collisions [15].

Packet outcomes are then used to estimate the tracking performance. We also moved the initial position of the GW to see how much the tracking performance is affected by the communication limitations. In the rest of the section, we will analyze the positioning error for different tracking and communication scenarios. In particular, we investigate our tracking scheme for different values of the SF and of the initial distance d between the UAV and the GW. We remark that the UAVs will move from the initial point, so their distance from the

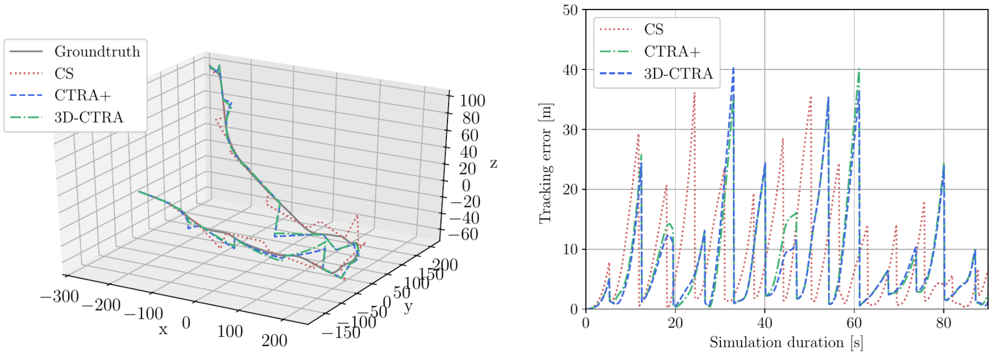
GW will change over a simulated episode, but the initial distance can give a qualitative idea of the range over which the drones will move.

We always initialize EDs to use the lowest SF setting that allows reliable communications within at least a range d from the GW in the absence of interference. However, each UAV may change its SF according to the considered ADR algorithm. We implemented the following ADR strategies:

- In the *NO ADR* setting, EDs do not set the ADR bit in their UL packets, and implement no autonomous data rate adaptation scheme. When a UAV travels out of the coverage zone, its packets are simply not received until it comes back in range.
- In the *NS ADR* setting, all decisions regarding changes in the SF of an UAV are taken by the NS (and conveyed to the device by the GW). In this case, the NS will leverage one of the two receive windows opened by the device after each UL transmission to send a DL packet containing the new SF value to use. The NS will increment the SF when it senses that the UAV is leaving the coverage zone for the current setting (based on the receive power of the last communication), and will decrement the UAV's SF when the received power is above receiver sensitivity by a certain margin (set at 3 dB in this work). Note that the control message will be sent only if a change in the SF is required, otherwise no DL transmission is performed.
- Finally, the *ED+NS ADR* setting has the UAV require confirmation of correct reception for every UL packet it sends. If no ACK is received, the UAV increases the SF on its own. Similarly to what happens in the NS ADR setting, when the UAV comes back in range, the NS will instruct the drone to decrement the SF through a DL communication.

Tracking a single UAV

First, we consider the 90 s drone path shown in Fig. 5.10a. In this scenario, we consider the GW to be positioned in $(0, 1000, 0)$, with an initial distance $d = 1000$ m. The same figure includes the paths estimated by the control station using the CS and 3D-CTRA motion models, considering a communication setup with SF 7 and $B = 250$ kHz. Comparing the different paths, we observe how the CS scheme has difficulty in following the target while 3D-CTRA ensures smaller deviations from the real path. The sharp changes in the estimated paths are due to updates received by the control station: if the model estimates



(a) 3D path.

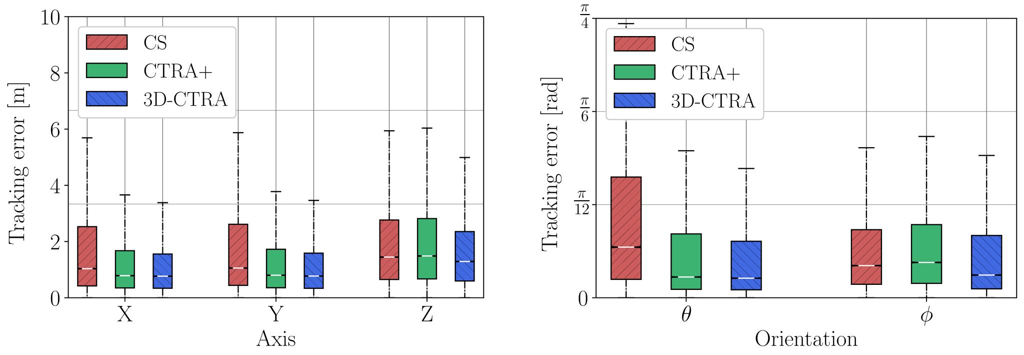
(b) Error over time.

Figure 5.10.: Tracking performance of a single UAV with $d = 1000$ m, SF 7, and $B = 250$ kHz.

the wrong attitude, updating it blindly makes it gradually diverge from the actual path of the UAV. When the control station receives the next packet from the UAV, it resets the state of its filter, resulting in a "jump" in the estimated path. This is confirmed by the results in Fig. 5.10b, which shows the control station tracking error over time for all the considered models: the updates are clearly visible, as the tracking error instantaneously drops. We highlight that the error of the CS model rapidly increases every time the drone performs non-linear movements, as happens at time $t \simeq 20$ s. Instead, the error of CTRA+ and 3D-CTRA presents a smoother trend, with fewer and lower peaks, even though both such models require to transmit more data over the channel. The performance of CTRA+ is almost identical to that of 3D-CTRA, but the latter better addresses vertical drone movements (as occurs at time $t \simeq 40$ s).

In what follows, we evaluate the performance of our system in tracking a single drone starting from a distance $d = 1000$ m from the GW. In particular, we examine the cumulative results over multiple virtual trajectories for a total of 2.5 hours of flight time. Fig. 5.11a shows the distribution of the position error along the three axes with SF 7, and $B = 250$ kHz. In particular, we adopt the boxplot representation, where the white line at the center of the box is the median of the distribution, the box edges are the 25th and the 75th percentile, while the box whiskers represent the 5th and the 95th percentile, respectively.

When considering the X and Y axes, the CS model is outperformed by both CTRA+ and 3D-CTRA, which ensure a richer representation of the drone's movements. In particular, 3D-CTRA shows a slightly lower position error with respect to CTRA+, which uses less information than 3D-CTRA to estimate the drone state. When considering the vertical Z axis, the error of CS is similar



(a) Error on the three axes.

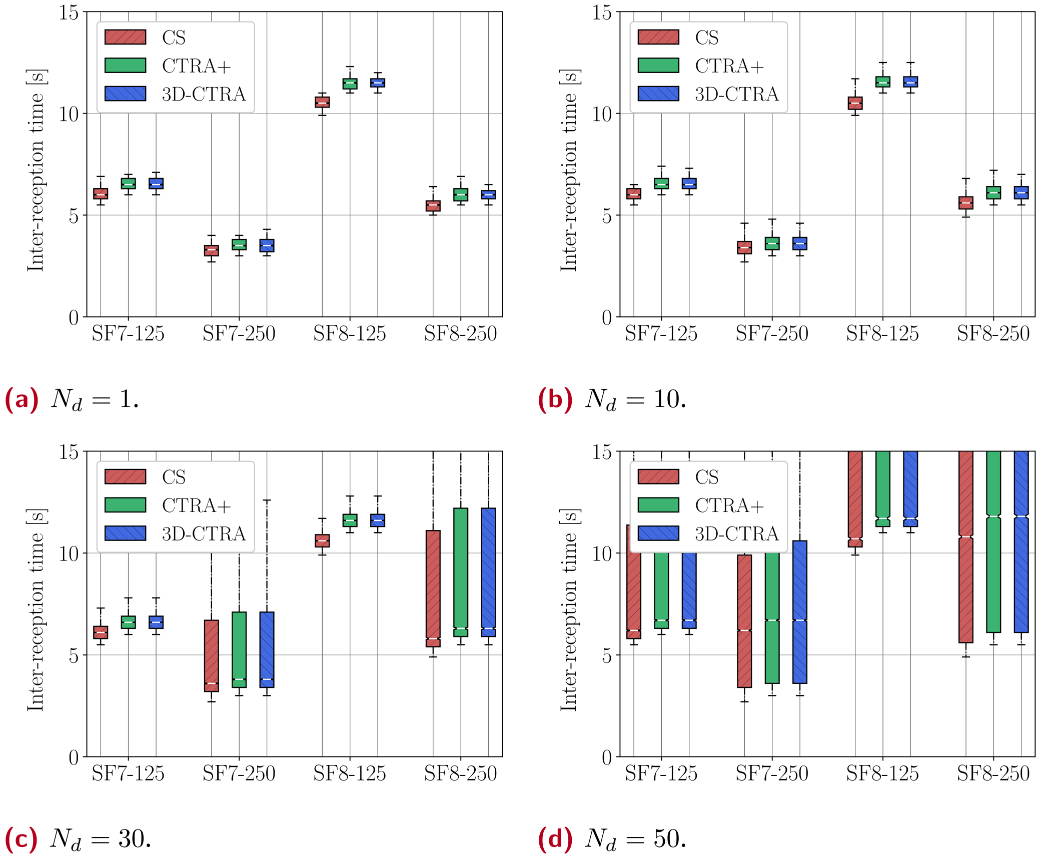
(b) Error on the orientation.

Figure 5.11.: Tracking error with $d = 1000$ m, $N_d = 1$, SF 7, and $B = 250$ kHz.

to that obtained with CTRA+, since both models consider that the target maintains a constant tilt. On this axis 3D-CTRA performs best, because it is the only one that can accurately track the drone maneuvers in all directions. This is confirmed by Fig. 5.11b, which shows the tracking error on the UAV orientation. It is easy to see that the CS model cannot correctly track the orientation of the drone on the horizontal plane, as it assumes it will never change. On the other hand, CTRA+ and 3D-CTRA have similar performance, but 3D-CTRA manages to track the orientation better on the vertical plane, as it can represent the movement of the UAV more accurately.

Swarm tracking performance

Fig. 5.12 shows the distribution of the inter-reception time for a swarm of drones starting 1 km away from the control station, with different tracking systems and communication settings. This parameter almost corresponds to the Peak Age of Information (PAoI), i.e, the maximum value reached by the AoI before a new update is received, as the transmission delay is much smaller than the inter-reception time. We can observe that the CS model ensures the lowest inter-reception time since its state is constituted by only 6 variables. On the other hand, CTRA+ and 3D-CTRA need to transmit more data over the channel and are characterized by similar communication statistics. Furthermore, the size of the swarm is critical to determining the inter-reception time, as collisions can multiply the time between successful packet receptions. We can easily see that increasing the size of the swarm from $N_d = 1$ to $N_d = 10$ has almost no effect, while a system with $N_d = 30$ is significantly different. The choice of the bandwidth is also extremely important, as using $B = 250$ kHz allows UAVs to transmit data faster, but also reduces the



(a) $N_d = 1$.

(b) $N_d = 10$.

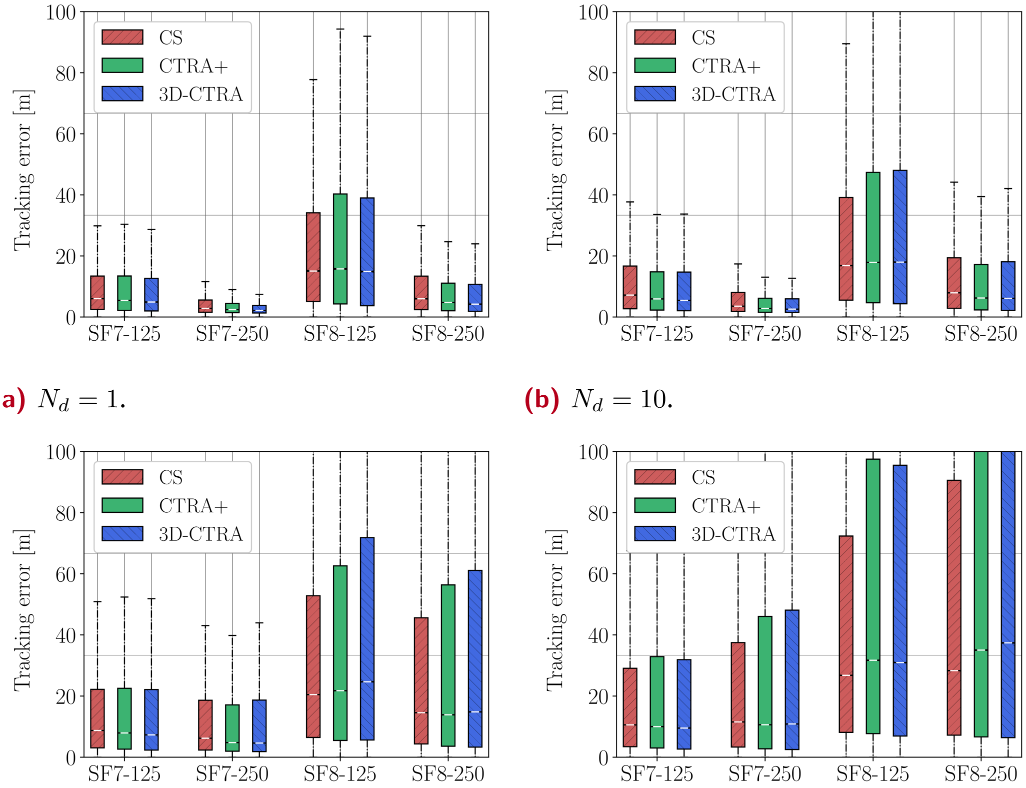
(c) $N_d = 30$.

(d) $N_d = 50$.

Figure 5.12.: Inter-reception time when using different communication settings with $d = 1000$ m.

number of orthogonal channels from 3 to 1, increasing the number of drones competing for the same channel. In fact, the increase in the inter-reception time for systems with $N_d = 30$ and $B = 125$ kHz is almost imperceptible. In general, the average inter-reception times are often still lower for $B = 250$ kHz, as the packet frequency is doubled, but the worst-case performance is much worse for $N_d = 30$ and $N_d = 50$, as shown by the upper halves of the boxes.

In general, reducing the PAoI corresponds to more frequent UKF updates at the control station, thus improving the tracking performance, but the relationship between PAoI and tracking error is not linear. Fig. 5.13 shows the tracking error distributions in the same scenario: in this case, choosing $B = 250$ kHz provides a better performance even for $N_d = 30$, but not for $N_d = 50$. Interestingly, we can see that 3D-CTRA outperforms the other models when the inter-reception time is low, but the simple CS model becomes the best whenever the average inter-reception time goes over 10 seconds. This is due to the non-linear nature of CTRA+ and 3D-CTRA, which can accurately track the UAVs' movements over a short timespan but suffer from the accumulation of errors over longer



(a) $N_d = 1$.

(b) $N_d = 10$.

(c) $N_d = 30$.

(d) $N_d = 50$.

Figure 5.13.: Tracking error when using different communication settings with $d = 1000$ m.

intervals: even without considering turns and maneuvers, a small error in the estimated acceleration can lead to a quadratic increase in the tracking error over time, while a constant speed model has no acceleration and thus will have a somewhat bounded error. However, in these conditions, the error for all the tracking models increases significantly, and might be too high for some applications.

Spreading factor optimization

As we discussed above, interference and packet collisions can be a significant issue in large swarms. However, LoRaWAN has another parameter that we can tune to maximize performance, namely, the SF. Packets sent with different SFs are orthogonal and can be received at the same time, providing additional protection from interference. At the same time, higher SFs can increase the packet transmission time significantly, leading to far longer inter-reception times to respect the duty cycle constraint.

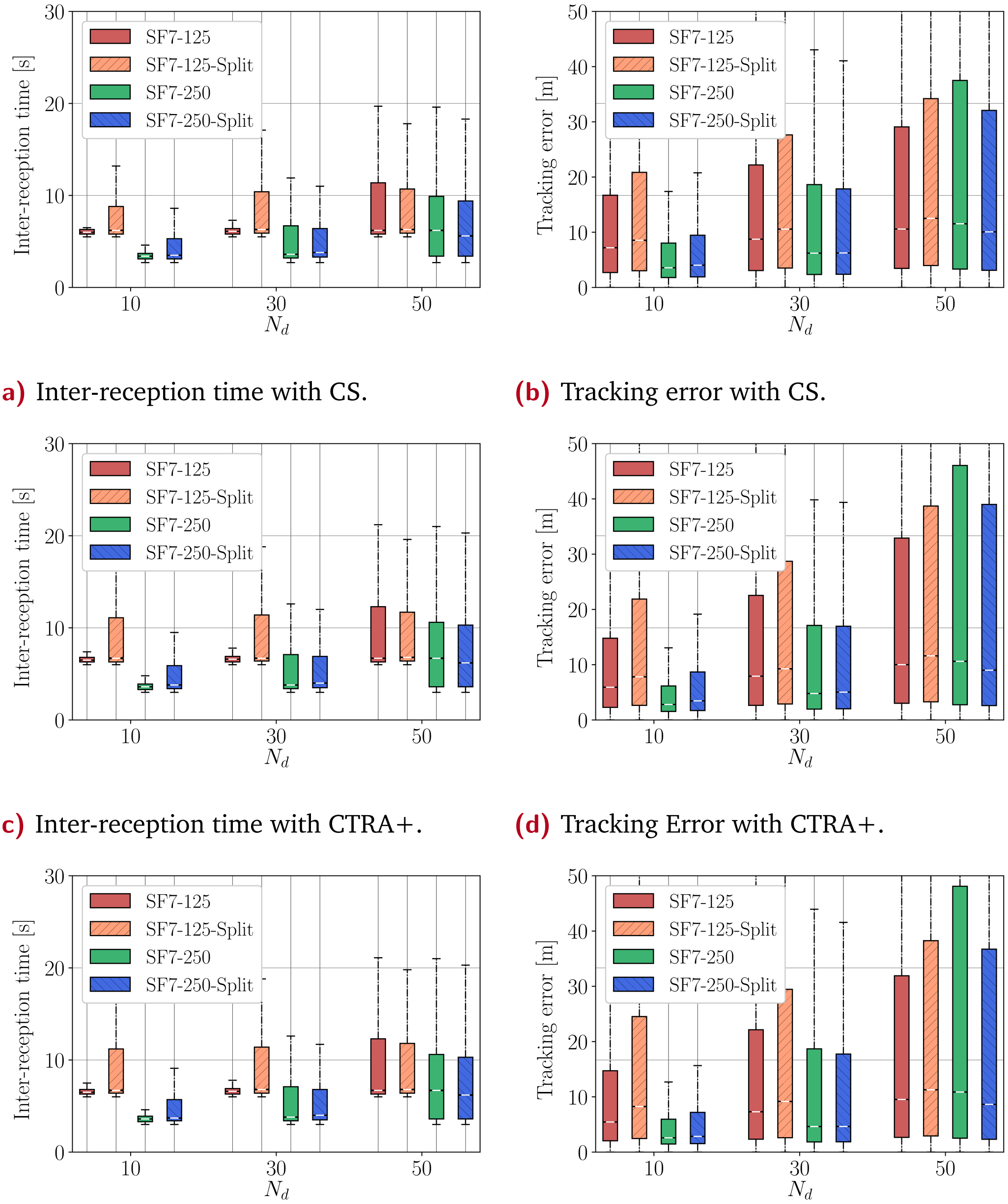
However, inter-reception times of over 10 s severely degrade the tracking performance. As such, we can limit the examined SFs to 7 and 8, i.e., the lowest two. In this case, the optimal choice to reduce congestion would be to have half of the UAVs use SF 7, while the other half use SF 8, but this would lead the UAVs that use the lower setting to have a significant advantage, as they have a shorter inter-reception time, and any collision for SF 8 packets would lead to a very long inter-reception time for the affected drones. In order to balance the two sets, we define a *split* system in which two thirds of the drones use SF 7, while the others use SF 8.

Fig. 5.14 shows that, for $N_d = 10$, splitting the UAVs between the two SFs is suboptimal, while it can provide a small performance bonus if $N_d = 30$ and $B = 250$ kHz. Interestingly, splitting the swarm is not a good choice for the system with $B = 125$ kHz even with a swarm of $N_d = 50$ drones: as reducing the bandwidth increases the number of orthogonal channels, but also the time required for a transmission, avoiding collisions is not worth the cost of further increasing the inter-reception times for the UAVs with SF 8. We also note that the SF optimization cannot solve the fundamental issue of the more complex models, as the updates are still not frequent enough to avoid divergence, and the CS model is still the optimal one for $N_d = 50$, maintaining the 75th percentile of the tracking error below 30 m.

However, SF adaptation algorithms are not just useful to reduce collisions, but also for their original purpose: extending the range of the LoRaWAN network. In Fig. 5.15, we analyze the results obtained using the ADR system we described in Sec. 5.2.3, for different values of N_d. If the ADR is not active (NO ADR), drones do not vary their communication settings regardless of the scenario conditions. Instead, the NS ADR and the ED+NS ADR systems allow UAVs to vary the SF according to the policies described in Sec. 5.2.4.

The figure clearly shows that using the ED+NS ADR scheme results in much higher inter-reception times and, consequently, a higher error. This is caused by multiple factors: firstly, UAVs using ED+NS ADR will increase the SF as soon as the GW misses an ACK to the UL message. The fact that ACKs need to employ the same SF as UL packets further exacerbates this problem, by increasing the duty cycle consumption by the GW and further limiting the amount of ACKs that can be sent.

As shown in Fig. 5.15, the good inter-reception time performance achieved by the NS ADR scheme results in a better tracking error, which makes the NS ADR scheme the best choice for the specific scenario we considered. However, we remark that different ADR policies might be implemented to also take



(a) Inter-reception time with CS. **(b)** Tracking error with CS.

(c) Inter-reception time with CTRA+. **(d)** Tracking Error with CTRA+.

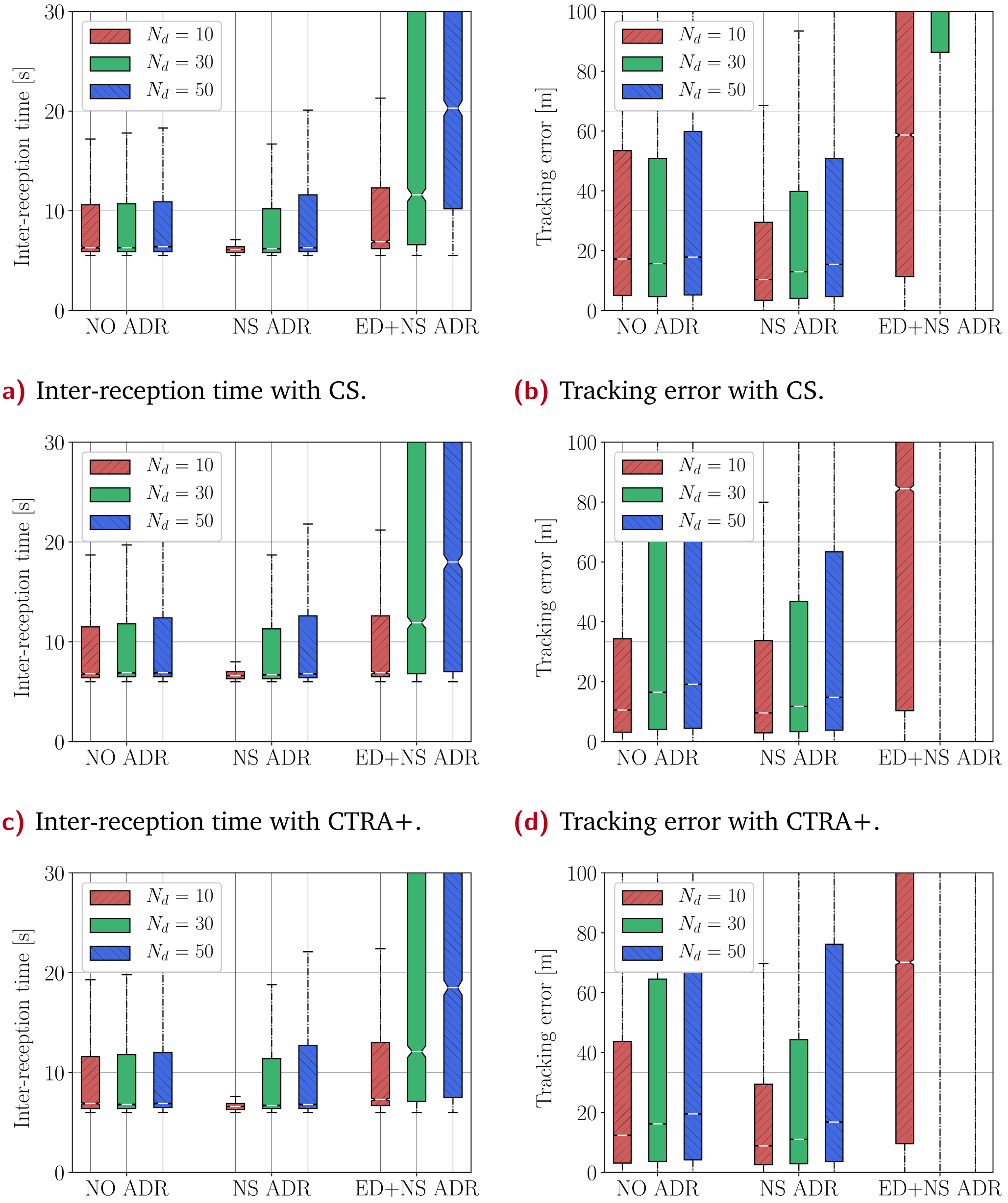
(e) Inter-reception time with 3D-CTRA. **(f)** Tracking error with 3D-CTRA.

Figure 5.14.: Performance when using different SF settings and $d = 1000$ m.

into account available knowledge of the UAV's mobility patterns for a smarter configuration of the SF parameter.

Joint tracking and sensing

We can now look at the consequences of piggybacking the tracking update on sensor communications. Naturally, LoRaWAN is a low-bitrate technology, so the sensor updates will themselves have a limited size. We analyze sensor payloads of 16, 32, and 64 bytes, combined with the three tracking systems



(a) Inter-reception time with CS.

(b) Tracking error with CS.

(c) Inter-reception time with CTRA+.

(d) Tracking error with CTRA+.

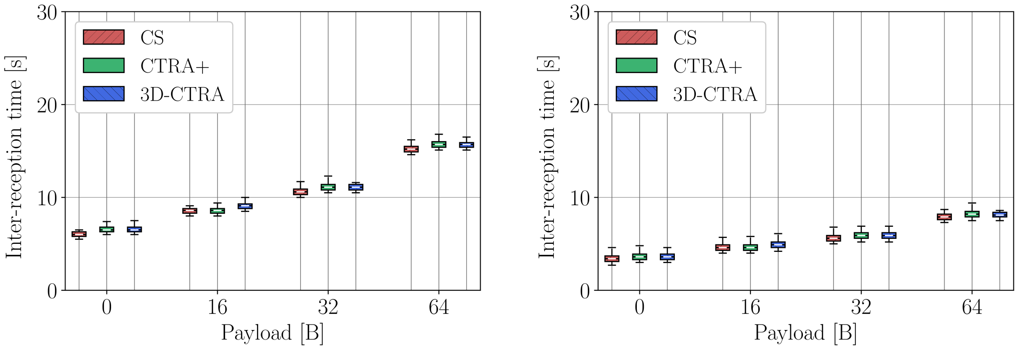
(e) Inter-reception time with 3D-CTRA.

(f) Tracking error with 3D-CTRA.

Figure 5.15.: Performance when using different ADR settings with $d = 3000$ m.

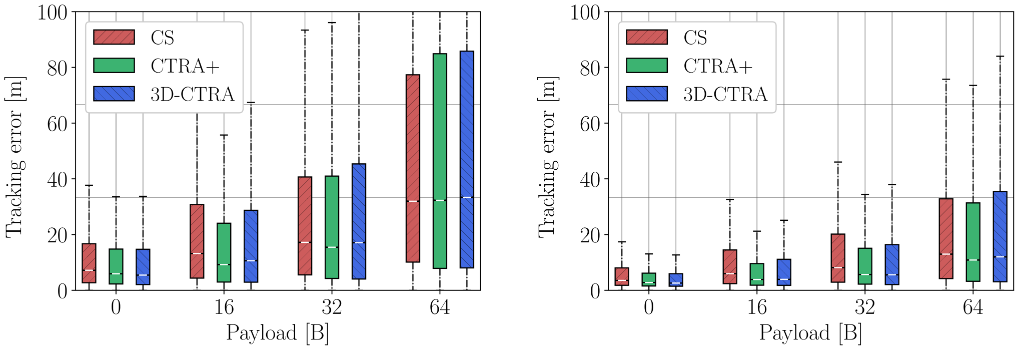
we have presented. We first look at the inter-reception time, which roughly corresponds to the PAoI for the sensor data, as the transmission takes up less than 1% of the total value. Fig. 5.16 shows a boxplot for a swarm with $N_d = 10$, using two different bandwidths, as above. Naturally, inter-reception time grows with the payload, but it is still below 10 s even with the 64 B payload when using $B = 250$ kHz.

As we discussed above, this has an obvious effect on the tracking error: Fig. 5.17 shows the tracking error as a function of the payload size. If we



(a) $B = 125$ kHz. **(b)** $B = 250$ kHz.

Figure 5.16.: Inter-rx time when using different communication settings with $d = 1000$ m, $N_d = 10$, and SF 7.

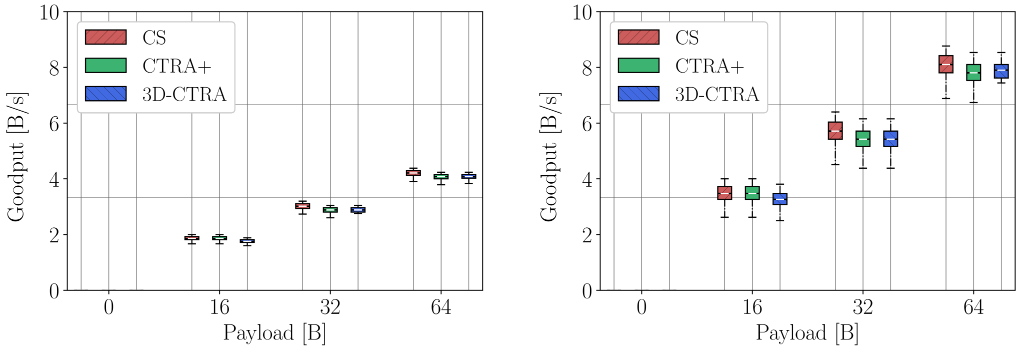


(a) $B = 125$ kHz. **(b)** $B = 250$ kHz.

Figure 5.17.: Tracking error when using different communication settings with $d = 1000$ m, $N_d = 10$, and SF 7.

consider the system with $B = 250$ kHz, the tracking error grows as the payload size increases, but is still often within acceptable parameters most of the time. It is interesting to note that, while CS is always the best choice with a non-zero payload and $B = 125$ kHz, CTRA+ is the best choice for small payloads with $B = 250$ kHz, as the inter-reception times are still within acceptable bounds to benefit from a more accurate model.

We can also look at the goodput generated for each payload size, which also increases as the payload size grows: Fig. 5.18 shows the received bytes per second for the sensor application, considering only the payload bits. There is a trade-off between goodput on one side, and PAoI and tracking performance on the other: smaller payloads can be delivered more frequently, correspondingly improving the tracking, but also reduce the overall goodput. We also analyzed the packet erasure probability, which was similar for all payload sizes and tracking schemes, around 5% for $B = 125$ kHz and 15% for $B = 250$ kHz.



(a) $B = 125$ kHz.

(b) $B = 250$ kHz.

Figure 5.18.: Goodput when using different communication settings with $d = 1000$ m, $N_d = 10$, and SF 7.

This difference is due to the far larger number of collisions that the drones experience with a single, higher-throughput channel, with respect to having 3 orthogonal ones with a lower capacity.

5.2.5 Conclusions

In this section, we presented a tracking system for UAVs, based on a novel 3D-CTRA mobility model and on periodic transmissions over LoRaWAN. Our system can track a drone's trajectory with high accuracy even when the drone is at 3 km from the LoRaWAN GW, and the mobility models we propose significantly outperform standard CS. In particular, the choices of SF allocation and channel bandwidth were found to significantly impact tracking performance when drone swarms are considered.

Moreover, despite LoRaWAN's duty cycle limitations, simulations conducted in this work show that the technology is suited to manage swarms of dozens of drones, provided that an appropriate ADR scheme is implemented to handle scenarios in which UAVs move out of the coverage area. As we discussed in the introduction, the accuracy reached by our system can enable several applications, from urban pollution monitoring to precision agriculture, which are relatively tolerant to imprecisions in the positioning.

Several extensions of this work are possible. The movement model could be refined by including maneuver and mission-level information to reduce the tracking error. Moreover, it would be interesting to explore features that are not part of the LoRaWAN standard up to now, like the use of a different frequency plan or of listen-before-talk instead of applying the duty cycle. Finally, the study of the behavior of swarms, and strategies to avoid packet collision, are interesting options that would enable new applications by improving the tracking accuracy at low cost.

6 Enabling Green IoT: evaluation of battery-less LoRaWAN nodes

A problem that arises when considering large IoT deployments is the powering of devices that, nowadays, is mainly faced by adopting battery-based solutions. However, this approach is not sustainable from both an ecological perspective, due to the dangerous chemicals still present in batteries, and an economical standpoint, as battery replacement is a costly operation. Therefore, a possible alternative to realize Green IoT solutions is to make devices self-sustainable in terms of energy. This can be done by employing a battery-less approach, where the energy is harvested from renewable sources, such as solar light or wind, and is stored into capacitors. However, the limited and inconstant energy supply and the limited energy storage capacity of such devices require special care in the design of communication and computational processes, which have a major impact on the energy consumption of the devices.

In this chapter, we continue the study of the LoRaWAN technology by evaluating the feasibility of a Green IoT solution with energy harvesting, with the aim of reducing the environmental footprint of IoT systems [157, 158].

In this perspective, it is fundamental to limit the use of batteries (both disposable or rechargeable) to power sensor devices, since their replacement is costly from a time, economic and environmental perspective, and motivates the migration towards greener solutions. An eco-friendly alternative to batteries is using energy harvesting techniques, where energy is derived from renewable sources (e.g., solar power, thermal/wind energy), and is stored in (super) capacitors to power the devices. Unfortunately, the variability of harvested energy and the small energy density of capacitors can potentially cause an intermittent behavior of the device, affecting its performance and capabilities, including communication.

The remainder of this chapter is structured as follows. Sec. 6.2 describes our ns-3 implementation of a battery-less IoT node, which is combined to the `lorawan` module presented in Chapter 3 while the specific ns-3 implementation and integration with existing modules are discussed in Sec. 6.3. In Sec. 6.4 we validate the capacitor's implementation applying it to LoRaWAN nodes, and discuss the obtained results. Then, we leverage such ns-3 implementation to investigate the performance of a battery-less LoRaWAN node with energy harvesting. In particular, we highlight the need for energy aware approaches, where packet transmission is conditional on the energy level of the device, making the best use of the available energy resources. We investigate the impact of various parameters, such as packet length and capacitor size under different environmental conditions. In Sec. 6.5, we propose and compare different energy-aware packet scheduling algorithms, that make it possible to better use the energy resources and to prevent the device from switching off. We test them in scenarios with various energy harvesting capacities, also using measurements of harvested power collected from real testbeds. Furthermore, in Sec. 6.6 we present some preliminary experimental results obtained by evaluating the performance of a real setup, where the power is harvested by a solar panel. Finally, Sec. 6.7 presents final considerations on the feasibility of battery-less LoRaWAN for Green IoT implementations.

6.1 State of the art

The adoption of battery-less approaches and energy-harvesting techniques have started gaining interest in the last few years, but have not been deeply evaluated yet. Therefore, works that deal with IoT networks with battery-less devices and energy harvesting [159, 160, 161, 162], mainly address theoretical analysis (i.e., mathematical modeling) or empirical evaluations. Instead, the use of simulations can provide a precious help in the evaluation of the interplay between the network state, the system configuration and the device's energy capabilities. For example, the communication could benefit from the robustness provided by message repetitions or the use of a higher transmission power, but this will impact the energy autonomy of the device; conversely, low energy levels may prevent the correct transmission of some packets. These aspects are further complicated when considering the variability of the energy source in nodes with harvested power and the interference of many communicating devices.

The inclusion of energy harvesting techniques in IoT devices has started gaining interest in the last years. As reported in [163], many contributions present LPWAN architectures with integrated energy harvesting, and LoRaWAN is one of the most tested technologies. In [164], Mabon et al. propose the architecture of an energy harvesting sensor node powered with solar panels, not related to a specific application. Instead, other contributions contextualize the system deployment in specific scenarios and with different energy harvesting sources: in [165, 166] the authors integrate solar panels for a structural health monitoring system for buildings, and for earthquake detection, respectively. In [167] the authors investigate the usage of Kinetic Energy Harvesting to convert motion/vibration energy into electrical energy to power IoT devices, and, through prototypes evaluation, show the efficiency of the proposed solution. In [168] Finnegan et al. explore the feasibility of a LoRaWAN sensor powered with radio-frequency (RF) ambient resources, and identify design and environmental constraints such as the hardware implementation, the availability of wireless energy (which depends on the specific scenario the system is deployed) and the distance between transmitter and receiver. Similarly, in [169], Loubet et al. describe a prototype implementation of a LoRaWAN monitoring network where devices are powered with energy harvested from RF sources, and propose to control the periodicity of the measurement and data transmission by tuning the availability of RF resources. In [170] Orfei et al. describe a system to monitor the asphalt of a bridge through LoRaWAN sensors powered by the bridge vibrations.

Differently from the aforementioned works, which face the problem using an empirical/hardware evaluation, other contributions leverage more theoretical approaches, drawing more general considerations. In [160] Delgado et al. employ Markov chains to model the intermittent behavior of a single LoRaWAN device with energy harvesting, including the effects of parasitic resistances of the capacitor, evaluating feasibility and performance when using UL and DL traffic, and validating it with a C++ simulator. The work presented in [161] considers battery-less capacitor-based LoRaWAN devices, and investigates a proper scheduling of sensing and transmission tasks, comparing a turn-off and a sleep-based approach. Furthermore, it studies the optimal turn-on voltage threshold that allows the node to complete such tasks for a given setting (capacitor size, energy harvesting rate), and validate the results using an environment emulator connected to a real device.

With respect to state-of-the art, in the analysis presented in this section, we consider energy-aware approaches that can work (and exploit) variable energy harvesting rates. In particular we focus on a single-node scenario and analyze the impact of different parameters (i.e., capacitor size and packet size) on the device's performance. We compare different energy aware scheduling approaches and employ various energy-harvesting traces obtained from real measurements.

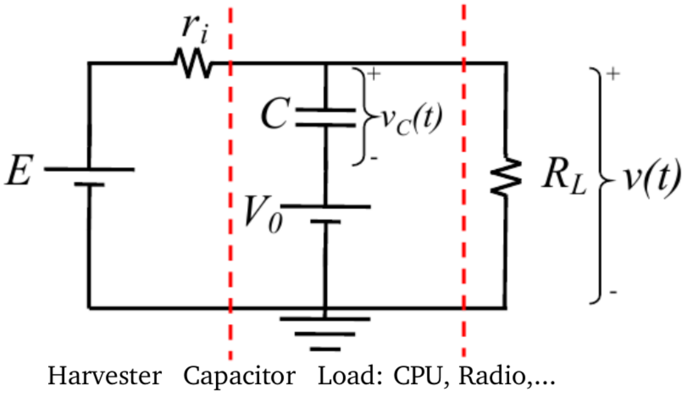


Figure 6.1.: Electrical circuit model of a battery-less IoT device [160].

6.2 Battery-less IoT devices with energy harvesting

To model a battery-less IoT device, we consider the approach used in [160]. The device consists of several components: a Micro Controller Unit (MCU), a radio unit, some peripherals (e.g., for sensing purposes), a capacitor to store energy, and a harvester mechanism to recharge the capacitor. The overall device can be modeled as an equivalent electrical circuit with three parts: (i) the harvester, (ii) the capacitor, and (iii) the load, as represented in Fig. 6.1, and better described next.

The harvester it is the only energy source in the system. The harvester is modeled as an ideal constant voltage source (denoted by E) with a series resistance ($r_i(t)$) that determines the maximum power that can be produced by the harvester, which is given by

$$P_{harvester}(t) = \frac{E^2}{r_i(t)} . \tag{6.1}$$

In general, the resistance $r_i(t)$ can change in time, according to the fluctuations on the energy harvesting process. By coupling the harvester with a voltage regulator, however, the output voltage E can be stabilized. Our model makes it possible to either generate the harvested power values as independent random samples taken from a given distribution, or to read them from a pre-loaded trace file. In the following, indeed, we will consider both an ideal harvesting source, characterized by a constant harvesting rate, and realistic energy sources, whose harvested power is given by the input traces obtained from solar panels, better described in Sec. 6.5.2.

The load it models all the components of the system that consume energy. According to the activity performed by each component, it is possible to define different *states* of the load, which are characterized by a specific power consumption. For each state, we can therefore define a load resistance $R_L(s)$, which is computed considering the total current $I_{load}(s)$ absorbed by the load in the specific state s:

$$R_L(s) = \frac{E}{I_{load}(s)}. \tag{6.2}$$

The capacitor it stores the energy generated by the harvester and releases it to the load when required. The behavior of the system can be represented by a series of intervals corresponding to different events/activities (e.g., MCU active and radio transmitting), corresponding to the different states of the load. For each state s, the voltage of the capacitor can be represented by $(V_0, v_C(t))$, where V_0 is the capacitor voltage when entering the state, and $v_C(t)$ is the voltage of the capacitor after t seconds spent in state s. As depicted in Fig. 6.1, V_0 is included in the circuit as an ideal voltage source, while $v_C(t)$ is the voltage over time of an ideal capacitor.

The voltage provided by the capacitor to the load after t seconds in state s can thus be computed as

$$v(t,s) = E\frac{R_{eq}(s,t)}{r_i(t)}\left(1 - e^{-\frac{t}{R_{eq}(s,t)C}}\right) + V_0 e^{-\frac{t}{R_{eq}(s,t)C}}, \tag{6.3}$$

where C is the capacitance of the capacitor [in Farads], and

$$R_{eq}(s,t) = \frac{R_L(s) r_i(t)}{R_L(s) + r_i(t)}. \tag{6.4}$$

To model real devices, we consider that they may switch off at anytime because of an energy level too low to continue their functioning. Therefore, we define two voltage thresholds for $v(t,s)$: V_{th_low}, and V_{th_high}. When the capacitor's voltage drops below the V_{th_low} value, the device switches off, and it cannot perform any operation. In this state, the current consumption is minimal (5.5 μA, due to the circuitry [160]), and most of the harvested energy will hence be used to recharge the capacitor. When the stored energy exceeds the voltage threshold $V_{th\_high} > V_{th\_low}$, the device goes back to the active state.

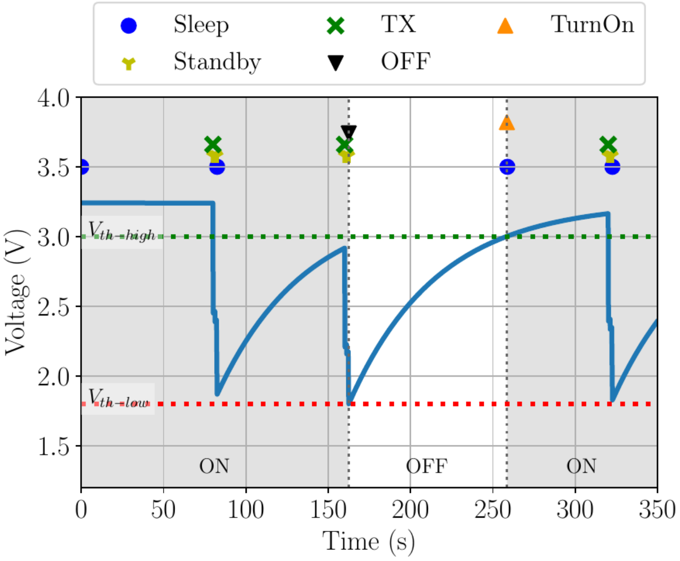


Figure 6.2.: Example of the device's voltage when it enters different phases.

As an example, Fig. 6.2 shows the voltage level of a device over time, highlighting the on/off phases with different background colors. The figure has been obtained by setting C = 5 mF, $P_{harvester} = 0.001$ W, packet generation period of 60 s, and LoRa DR 3. The voltage thresholds determining the switch to on/off states have been set to $V_{th\_low} = 1.8\ V$ and $V_{th\_high} = 3\ V$. Markers correspond to the beginning of the state, which will be better explained in Sec. 6.4.

6.3 Code implementation in ns-3

In this section, we present our ns-3 implementation of a battery-less node with a (super) capacitor coupled with an energy harvester, which enables the performance evaluation of IoT networks with battery-less devices. This is built as an extension of the native ns-3 energy model, and can be easily integrated with existing ns-3 modules, supporting the intermittent behavior of devices that can turn off or on according to their energy level. Also, a supplementary class makes it possible to import values for the harvested power from external files, which can be obtained from real measurements. The implemented code is available at [171]. In Figure 6.3, a scheme depicts the relation between the different components. We point out that the presented framework can be extended and/or tuned to consider also different behaviors of the device (e.g., states), or values for current consumption. Indeed, the application considered in this work is based on the `LoRaRadioEnergyModel` class and inherits its specificities.

Table 6.1.: Relevant attributes of the `CapacitorEnergySource` class.

Attribute	Description
Capacitance	capacitance [F]
CapacitorEnergySourceInitialVoltage	initial voltage of the capacitor [V]
CapacitorMaxSupplyVoltage	Maximum supply voltage for the capacitor energy source [V]
CapacitorLowVoltageThreshold	V_{th_low}, as fraction of the maximum supply voltage
CapacitorHighVoltageThreshold	V_{th_high}, as fraction of the maximum supply voltage
PeriodicVoltageUpdateInterval	Time interval between periodic voltage updates [s]

6.3.1 Capacitor

The ns-3 class implementing the storage of the energy in a capacitor is called `CapacitorEnergySource`, as it extends the `EnergySource` class available in ns-3.[1] Thus, it is used as done for the classes implementing the Lithium Ion Battery or the non-linear battery model. As such, it can be easily connected to energy harvester components (`EnergyHarvester`) and to the class modeling the energy consumption behavior of the device (`DeviceEnergyModel`). [2] The capacitor's features can be set using the class attributes and methods; the most relevant attributes are reported in Table 6.1.

The value of the energy stored in the capacitor and the corresponding voltage can be updated periodically by calling the appropriate function: `UpdateEnergySource`. This function computes $v(s,t)$ as for Eq. (6.3) according to the current state s, with V_0 being the voltage computed at its previous call. It is recommended to call the function also before switching the device to a new state, in order to keep the voltage up-to-date with the correct value of $R_L(s)$ and guarantee that the device's energy is not depleted, preventing the correct switching to the new state.

Besides the common "setters" and "getters" methods, and the auxiliary functions to compute the voltage level, the class also provides the following:

- `IsDepleted`, a function returning `true` if the current voltage value is below V_{th_low};

[1] `https://www.nsnam.org/doxygen/classns3_1_1_energy_source.html`

[2] To avoid compilation errors, in `ns3/src/energy/model/energy-source.h`, the *private* variables should be moved to *protected*.

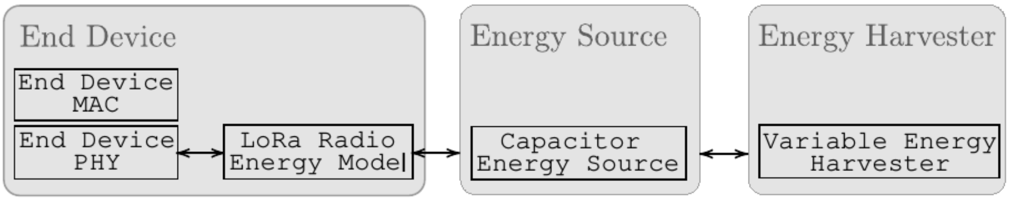


Figure 6.3.: Diagram illustrating the relation between different code components.

- `ComputeLoadEnergyConsumption`, a function computing the energy dissipated only by the load where a given current flows for a given time interval, and the initial voltage level of the capacitor is given as input;

- `TrackVoltage`, a function producing a file with the value of the voltage. The use of a trace source connected to the variable indicating the remaining voltage may instead have incorrect behavior because of the small difference between consecutive updates, which may not be detected by ns-3 native implementation;

- Traced values, to inspect their evolution through the simulation (e.g., the voltage level).

6.3.2 Variable energy harvester

The class `VariableEnergyHarvester` extends the `EnergyHarvester` class provided by ns-3, taking harvested power values from a .csv file given as input. The harvested power is periodically updated, and the energy source object(s) (e.g., the capacitor) connected to the harvester are updated accordingly.
The implementation of the function reading the .csv file is specific to the file we considered as input to our scripts. The code can be easily modified to consider data saved in a different format, since it only needs a pair (timestamp, $P_{harvester}$). Note also that, when running simulations, the length of the trace provided to the energy harvester should be as long as the simulated time. In case of mismatch, an error is raised during the simulation.

6.3.3 Integration with an existing module: `lorawan`

To validate the capacitor model, we employ the `lorawan` ns-3 module [40, 39]. In particular, we extended the class `LoraRadioEnergyModel`, a child class of the `DeviceEnergyModel`, to work with the capacitor's implementation, and added some states and variables to improve its compliance to real devices. Furthermore, the module's classes representing the MAC and PHY layers of the devices have been modified to update and verify the energy level before switching to a new state. The on/off behavior is implemented as follows: if the stored voltage is below V_{th_low}, the capacitor enters in the "depleted" state, the `LoraRadioEnergyModel` is notified, and the device enters into the Off state, interrupting ongoing transmissions, if any. In this case, the transmitted signal is considered only as interference by all the receivers. Conversely, when enough energy is harvested, and the voltage is above V_{th_high}, the capacitor switches from the "depleted" to the "recharged" state, trigging some events from the `LoraRadioEnergyModel` class (usually, the switch to the Sleep state) and enabling again packet transmission.

Furthermore, we also implemented the behavior of a smarter device, which is able to predict the energy cost of a packet transmission: if the predicted voltage after the operation is below V_{th_low}, the transmission at the MAC layer is not performed since, being incomplete, it would be unsuccessful.

6.4 Application and validation

In this section, we consider the LoRaWAN technology to validate the `CapacitorEnergySource` class presented in Sec. 6.3, and consider the analysis as first results evaluating the feasibility of a battery-less approach applied to LoRaWAN devices. In the following, we first introduce the states that characterize the LoRaWAN device operations. Then, we show preliminary results where the proposed framework is applied to LoRaWAN nodes, and validate the approach with a comparison with state-of-the-art approaches. Then, we expand the analysis to observe mutual relations between capacitor's properties and the configuration of the technology, and how they impact on the success of the communication.

In the following discussion, we assume only UL traffic. Furthermore, to measure the quality of the communication, we will indicate as *UL cycle* the interval between the beginning of the ED's packet transmission till the moment when it is successfully delivered to the GW, and *UL + DL cycle* the interval from the moment when the ED starts the UL packet transmission till the successful reception of the corresponding ACK.

6.4.1 LoRaWAN device states

As discussed in Sec. 6.3, the different states the device goes through are important to determine the energy consumption of the device. According to the LoRaWAN protocol described above, the following states are identified.

Table 6.2.: Current consumption in the different states [160].

State	MCU	Radio current	Total current
Off	Standby	0	5.5 μA
Turn On	Active	not considered	15 mA
Sleep	Standby	0.1 μA	5.6 μA
Tx	Active	28 mA	28.011 mA
Idle	Standby	1.5 μA	7 μA
Standby	Standby	10.5 mA	10.5055 mA
Rx	Active	11 mA	11.011 mA

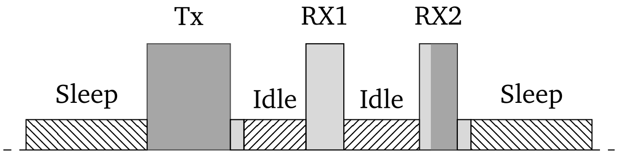


Figure 6.4.: Example of ED's state transitions.

The respective current consumption is reported in Tab. 6.2, considering the load composed by the MCU and radio units. The contribution of the MCU is considered only in terms of current consumption, and corresponds to 11μA in active state, 5.5 μA in standby state.

- *Off* (*): the ED's radio is switched off, and the MCU is in standby, maintaining only the clock synchronization;
- *TurnOn* (*): the device wakes up from the Off state, with a certain energy expenditure. In our implementation we consider a current consumption of 15 mA and a state duration of 300 ms, but the values can be tuned according to specific devices considered;
- *Sleep*: the radio is in sleep state, saving power, without performing any activity, and the MCU is in standby mode;
- *Tx*: the device is transmitting data;
- *Idle*: "waiting" period before the opening of the receive window;
- *Standby*: listening to idle channel when the receive windows are open. Also, the standard defines the ED to switch to Standby (for a very short time) after transmission and reception operations;
- *Rx*: the device is receiving data.

The Off and Turn On states, marked with (*), are not part of the standard, but are present in real devices. In Fig. 6.4 a diagram depicts the operational states of the device. In this case, the device wakes up from the Sleep state to transmit data requiring an ACK, which is successfully received in RX2, after a short time spent in Standby mode. The light-grey-colored regions represent the Standby phase. Note that the device will be able to complete an UL cycle if the voltage level is enough to complete the transmission procedure (no impairments from the channel/network are considered), while an UL + DL cycle is successfully executed if the energy stored in the device is enough to complete all the operations from transmission to reception, which can happen in either RX1 or RX2, including also the intermediate Idle and Standby states.

6.4.2 Results

To test our ns-3 implementation, we consider a simple LoRaWAN network composed of a NS, a GW and a single ED provided with a capacitor with variable size. Furthermore, we consider V_{th_low}=1.8 V and V_{th_high}=3 V, and different values for $P_{harvester}$, from 0 W to 0.01 W, which are constant in time. The device periodically generates packets with a payload length of 10 bytes, and can use either unconfirmed or confirmed messages, with m=1. The smart option preventing a packet transmission if the energy cost is not supported by the device is used. The LoRa settings are as considered in [160].

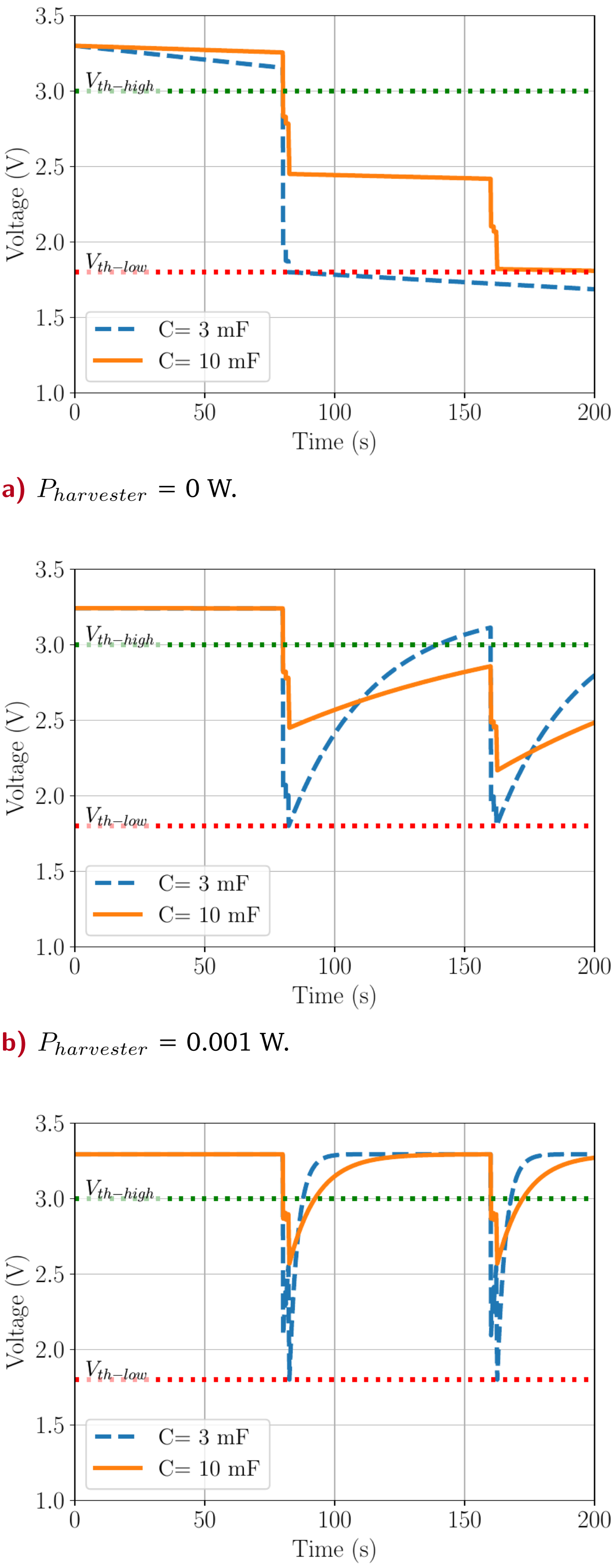
Fig. 6.2 shows the capacitor's voltage together with the states of the device.[3] The initial voltage of the capacitor is 3.3 V, which is almost constant for the first part of the simulation, when the device is in Sleep mode. At t =80 s, the ED performs the first transmission, entering the Tx state, which causes a first drop in the capacitor's voltage, bringing it to 2.44 V. The traffic type is unconfirmed, therefore, no DL transmission is expected. Nonetheless, as dictated by the standard, the two reception windows are opened, which cause the voltage drop around time 81 s. Note that, as it can be seen also in Fig. 6.5a, the energy drop during RX1 is smaller than that experienced during RX2, whose duration is $2^6 = 64$ times longer than that of RX1. In the simulated scenario, the harvesting rate is EH=0.001 W, which allows the capacitor to recharge during the sleeping period from 81 s to 160 s, reaching almost 3 V. Note that this is not visible during the initial sleeping period, because the voltage level is very close to the maximum voltage supported by the device. At 160 s a second transmission occurs, followed by the opening of the two reception windows. In this case, the voltage at the beginning of the cycle was lower than in the previous case, and the long duration of RX2 makes

[3] Markers signaling when entering in Standby and Idle states are not plotted for clarity.

the voltage drops below V_{th_low}, so that the ED enters the Off state. Then, a recharging phase follows, and when $v(t, s)$ reached V_{th_high} =3 V, the ED starts the TurnOn phase, entering into the Sleep state. This enables it to successfully perform the next transmission, at time t=240 s.

Fig. 6.5 shows the voltage level for different capacitor sizes and $P_{harvester}$ values, which have a determinant impact on the communication capabilities. As a benchmark, in Fig. 6.5a we report the case with no harvesting. From this, we can appreciate the impact of different capacitors' sizes: a smaller capacitor discharges much faster than a bigger one, and can rapidly make the device switch off. However, when using smaller capacitors, also the recharging phases are faster, as it can be observed in Fig. 6.5b and Fig. 6.5c: this behavior may cause the device to swap between On and Off states, preventing proper communication. Conversely, a larger capacitance will charge and discharge more slowly, allowing better communication performance (in terms of successful transmissions) also in the case of lower harvesting rates, since it will reduce the number of times the node enters in Off state and, consequently, the energy cost for taking it back to the active state. The downside is that, whenever the capacitor voltage drops below the lower threshold, it will take a longer time to accumulate enough energy to pass again the high threshold and bring the device back to an operational state. Therefore, while the V_{th_low} is typically hardware-dependant, a proper tuning of the V_{th_high} threshold should also consider the capacitor's size.

The minimum capacity size that makes it possible to complete an UL (resp. UL + DL) cycle for different UL packet sizes, harvesting rates and DRs is presented in Fig. 6.6, and compared with mathematical results obtained from the model proposed in [160], which are represented with lines, while markers represent simulation outcomes. The results of the model had been confirmed by comparison with real devices in [161]. The DL packet size is fixed to 39 bytes (at APP layer). Also, the initial capacitor voltage is computed taking into account the current in the Off state. Since there is a single ED in the network, the GW can always use RX1, sparing the device the energy consumption due to additional states. In both plots, we can observe that the minimum required capacitance increases for bigger payloads of the UL packet, as expected. Moreover, the lower the DR, the larger the required capacitance, because of the longer transmission time. For lower harvesting rates, a larger capacitor should be employed to successfully complete a cycle, as discussed previously. Similar trends can be observed for the minimum capacitance needed to accomplish an UL + DL cycle: in this case, the values are higher

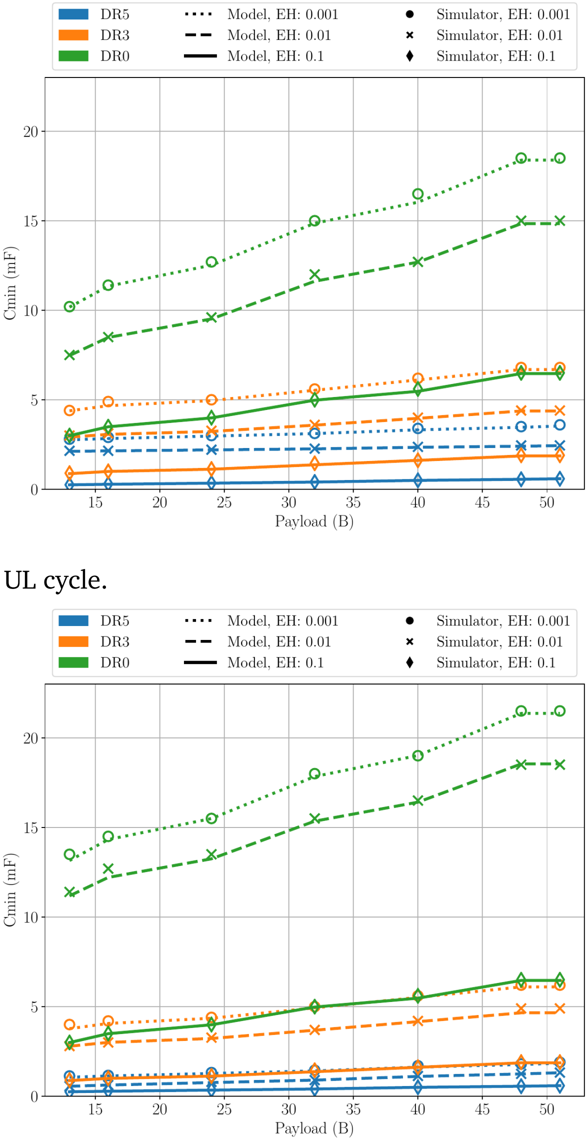


(a) $P_{harvester}$ = 0 W.

(b) $P_{harvester}$ = 0.001 W.

(c) $P_{harvester}$ = 0.01 W.

Figure 6.5.: Voltage for different values of $P_{harvester}$ and capacitor's size.



(a) UL cycle.

(b) UL and DL cycle.

Figure 6.6.: Minimum capacity to complete a cycle compared to mathematical results, computed as in [160].

than for the UL cycle only, since additional actions must be performed by the device, including the reception of a DL packet. From these plots we can finally observe that there is a strong agreement between model and simulation results, with the small discrepancy between the two only due to the quantized step used in the simulator.

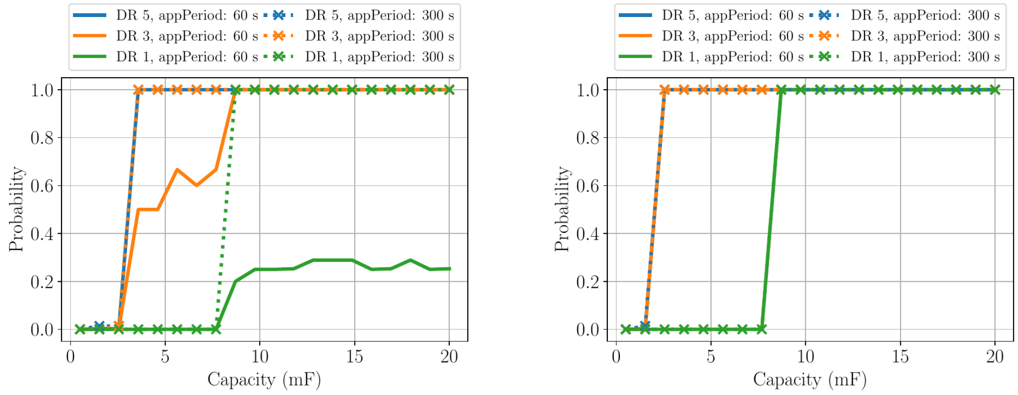
A final batch of simulations was run to reproduce a realistic scenario where an IoT application periodically generates packets of fixed size. In particular, we evaluated the success probability (in terms of delivered packets) when varying the capacitor size, for an application sending confirmed/unconfirmed traffic for 6 hours, with different packet generation periods. Note that the success probability is computed as the ratio between the number of delivered packets and the total number of packets generated at the application level.

Fig. 6.7 and Fig. 6.8 show the results for different values of the packet generation period and harvesting rate. Also in this case, we can observe a strong dependence on the DR: while lower DR values improve the transmission robustness to possible channel impairments, they are more costly in terms of energy, strongly affecting the number of packets that are successfully received by the GW. This can be mitigated by storing more energy, as happens for bigger capacitors charged with higher harvested power (Fig. 6.7b). Instead, using higher DR values require smaller capacitors, in the order of a few mF. Furthermore, transmitting packets more sporadically leaves enough time to recharge the capacitor, obtaining a higher success probability for a given capacitor's size.

For example, increasing the interval between consecutive packet transmissions from 60 s to 300 s makes it possible to halve the minimum capacitance when using DR 3 and $P_{harvester} = 0.001$ W (Fig. 6.7a). Fig. 6.8 reports similar results for the probability of also receiving the ACK: in this case, similar considerations on the relation between minimum capacity, DR and $P_{harvester}$ values can be drawn. However, it is interesting to note that, for low values of $P_{harvester}$ (Fig. 6.8a), the capacitor's size that maximizes the success probability is lower in the case of confirmed traffic than when using unconfirmed traffic, despite the reception of DL packets occurs. This confirms that using an ACK (with no payload) to prevent the opening of RX2 brings some benefit on the ED's energy consumption and communication performance, specially when the harvested power is low. These aspect could be taken into account for a proper network configuration that targets energy efficiency: using shorter RX2 by employing higher DRs, or even preventing their use, could have a significant effect on the device's energy performance.

6.5 Energy-aware scheduling approaches and simulation settings

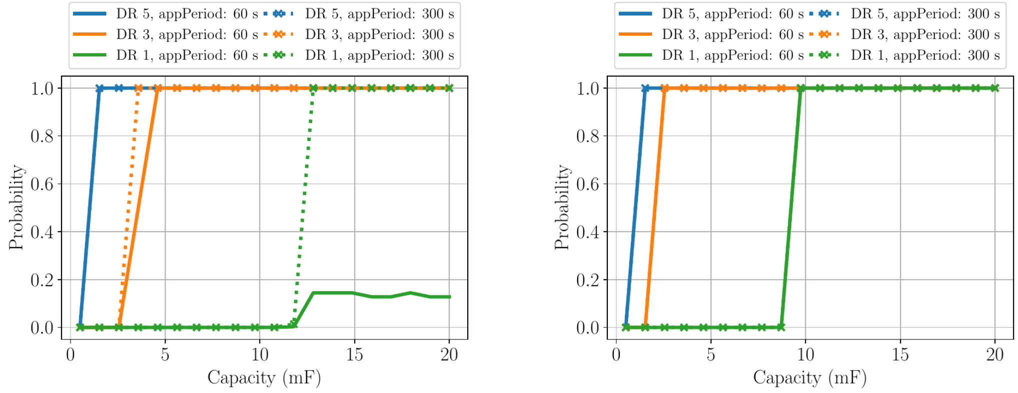
In this section, we introduce some scheduling algorithms that implement different energy-aware algorithms to determine when to transmit sensor data. Then, we describe the settings employed in our simulations and the perfor-



(a) $P_{harvester}$ = 0.001 W.

(b) $P_{harvester}$ = 0.01 W.

Figure 6.7.: Success probability for unconfirmed traffic.



(a) $P_{harvester}$ = 0.001 W.

(b) $P_{harvester}$ = 0.01 W.

Figure 6.8.: Success probability for confirmed traffic.

mance metrics that we leverage to evaluate the different packet scheduling approaches.

6.5.1 Packet scheduling approaches

To evaluate the feasibility of battery-less LoRaWAN devices powered with energy harvesting systems, we investigate the design of different packet scheduling approaches that take the device's energy level into account, and make the best use of the available energy resources. To this aim, we compare the following approaches, generating UL data packets in different ways.

- **Unaware Sender (US)**: it generates packets with a period I, independently of the device's energy level. This energy unaware approach will be used as baseline for comparison with energy aware approaches.

- **Energy Aware Sender with Fixed Threshold (FS)**: it generates packets when the voltage level of the capacitor is above a fixed threshold and at least I seconds have passed from the previous transmission.

We also consider other energy aware senders where the threshold voltage for generating packets corresponds to the minimum voltage needed to successfully complete the transmission cycle, preventing the device from switching off. This threshold is computed dynamically, taking into account the communication parameters: packet size, SF, and whether or not an ACK is required. Furthermore, when computing the expected energy cost of the cycle, recharging of the capacitor during the cycle is also considered. In particular, the expected harvested power can be computed according to different algorithms that, in turn, determine the following sending algorithms:

- **Conservative Sender (CS)**: it computes the threshold considering no energy harvesting, i.e., conservatively assuming a worst case scenario.

- **Simple moving Average Sender (AS):** it considers that the harvesting during the cycle equals the mean harvested power in the last x seconds. As discussed in Sec. 6.5.5, we tested different values for x, whose impact was not very significant. In following, we will show the results for $x = 5$ s. constant for the duration of the whole cycle, and $x = 5$.

- **Optimal Sender (OS):** in this case, perfect knowledge of the harvested power during the whole cycle is considered. While this assumption is not realistic, results obtained with this algorithm represent a performance upper bound for the considered harvesting scenario.

constraints are immediately dropped. This assumption results in different behaviors between US and the energy aware approaches. When setting I to be lower than the minimum silent time imposed by the DC, an ED whose transmission has been prevented by the DC will wait I more seconds before the generation of the next packet under US algorithm, while with the energy aware algorithm, once the silent time has passed, the packet will be generated (and transmitted) as soon as the energy level permits.

6.5.2 Energy harvesting traces

In the following analysis, we test the device's performance with different packet scheduling approaches, considering different kinds of harvesting sources, namely: (i) an ideal energy harvesting source providing constant harvested

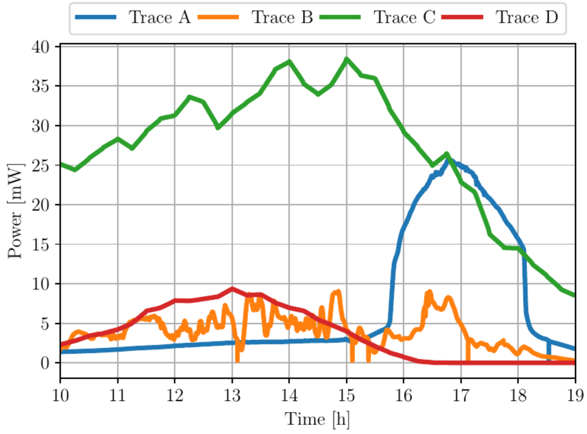


Figure 6.9.: Harvested power P_h for different traces.

Table 6.3.: Main features of harvested power traces.

Trace	Day of collection	Position	Power (avg±std) [mW]	Reference
Trace A	sunny day in September	West-facing windowsill	7.2±8.2	self-collected in Antwerp (BE)
Trace B	cloudy day in September	West-facing windowsill	4.0±2.0	self-collected in Antwerp (BE)
Trace C	12th June, 2010	North-facing windowsill	27.5±8.1	[172], Setup C
Trace D	12th December, 2009	North-facing windowsill	3.8±3.3	[172], Setup C

power, (ii) several empirical traces of solar power harvested in indoor environments. For the second case, we consider 4 power traces collected from 10:00 to 19:00 in different periods of the year, different locations and with different hardware. As it can be seen in Fig. 6.9, the harvested power is quite different, providing us with various scenarios that make it possible to appreciate the system performance. Further details about these traces are reported in Tab. 6.3. All measurements have been collected by employing 6-cells mono-crystalline (4x2 cm^2) solar panels.

6.5.3 Simulation settings

For our simulations, we leverage the `lorawan` ns-3 module and the capacitor implementation described in Chapter 3 and Sec. 6.3, extended to evaluate different packet schedulers. We simulate a single-gateway single-ED network, with the ED transmitting packets with different data payloads (PL) using SF 7. Note that, since a single device is employed, the effect of using different

SFs would only be on the transmission and RX1 durations. Since in case of unconfirmed traffic the duration of RX1 is assumed to be negligible compared to the time on air of UL packets, we decided to investigate the impact of variable PL size with fixed SF. Indeed, in the specific scenario considered in this study, varying the SF would only yield a rescaling of the results, without changing the considerations that can be drawn from the analysis.

The ED is provided with an empty capacitor, which is charged by the harvesting process. The duration of the simulations was set to 9 hours, based on the available energy harvesting traces. The device switches off if the voltage falls below $V_{th\_low} = 1.8$ V, value that are in line with off-the-shelf LoRa devices [160], while the turn on threshold is set to $V_{th\_high} = 3$ V. The minimum interval between the generation of consecutive packets is set to $I = 4$ s, that yields a maximum transmission rate larger than what is actually allowed by the DC constraint for PL≥ 5 B, which hence sets an upper bound to the achievable throughput. In this way, we can better appreciate the effect of the different scheduling algorithms. In the results shown below, the voltage threshold for generating packets with the FS scheduling algorithm is set to 1.82 V, slightly above the V_{th_low} threshold.

6.5.4 Performance metrics

To compare the different scheduling algorithms, we will employ the following metrics:

- Number of UL packets successfully transmitted by the ED. Note that it may be possible that a packet is successfully transmitted, but the ED is not able to complete the cycle because of a low voltage value. In this case, the packet transmission is successful, but the device will switch off, possibly preventing future transmission if not able to recharge on time.

- Since the capacitor is initially empty, during the simulation the device can be in operational state (ON) or not (OFF), or in the initial charging phase (Charging). We measure percentage of simulation time the ED spends in each of these states.

- Mean time between consecutive transmissions, neglecting the initial charging phase. Note that, in the best case (i.e., no energy constraints), this is lower bounded by the DC limitation.

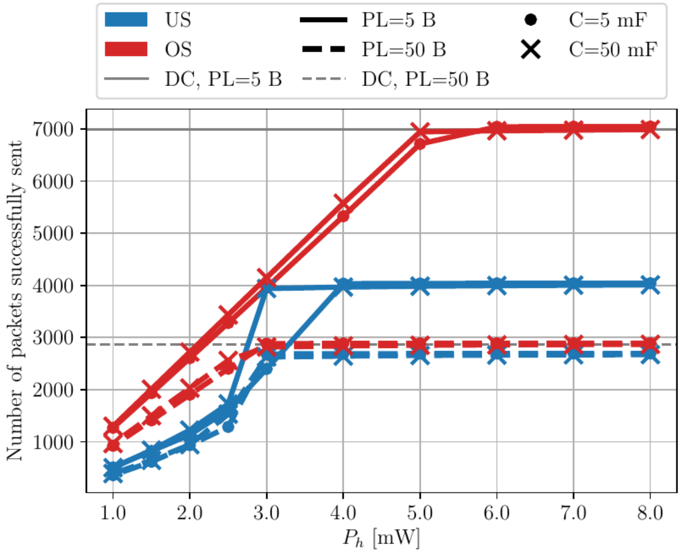


Figure 6.10.: Performance for different configurations, with constant P_h.

6.5.5 Results

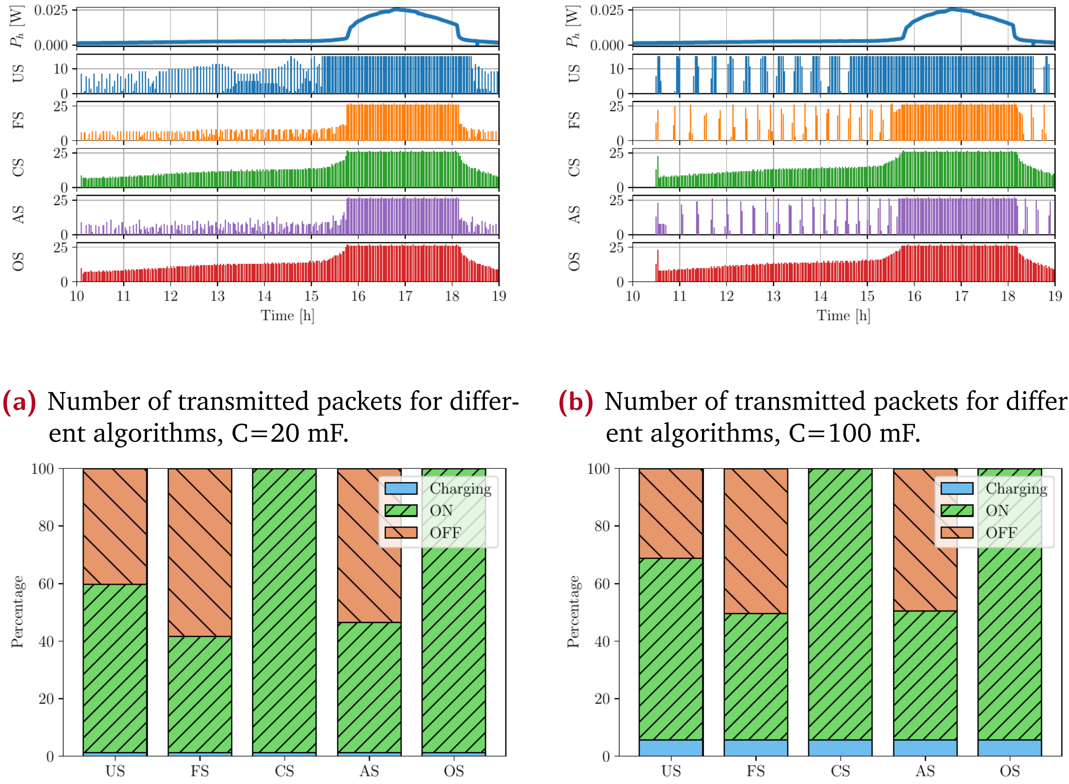
In the following, we present the simulation results that showcase the importance of considering energy aware algrithms for packet generation, and compare the performance of the algorithms previously proposed.

Importance of energy awareness

We begin our discussion by assessing the maximum gain in number of sent packets that can be obtained moving from an energy unaware algorithm (US) to the best possible energy aware scheduling approach (OS). Also, we show the impact of multiple parameters, with the assumption of constant harvesting power P_h. From Fig. 6.10 we first notice that, as expected, the ED is able to transmit more packets for higher values of harvested power, converging to a maximum value for high values of P_h. A second element that impacts the performance is the amount of data generated by the application: indeed, when PL=50 B (dashed lines), the number of sent packets is smaller, because of both the higher amount of energy required to successfully complete the cycle and the DC limit, as better explained later. To be noted that the maximum number of packets that can be transmitted in the considered period (9 hours), within the DC constraint is 6997 when PL=5 B, and 2870 when PL=50 B (see Tab. 2.8), which explains the convergence to these values for high harvested energy. Finally, a third factor to consider is the dimension of the capacitance, which however has a more limited impact. Bigger capacitors take longer to charge/discharge, but are then able to complete more transmission cycles.

Comparison of scheduling approaches

Assessed the potential of energy-aware policies over energy unaware ones, we now look for the best energy aware scheduling approach, evaluating different settings. In Fig. 6.11 we report the number of packets transmitted by the ED over time (upper plots), and the fraction of time the device spends in the ON/OFF states (lower plots). The transitory time needed to charge the capacitor, before the node becomes operational is considered separately and indicated as "Charging". More in detail, the upper plots show the evolution of the harvested power for trace A over time, and the number of packets sent by the device in this scenario, with each bar representing the aggregated number of packets sent during two-minute intervals for different schedulers and capacitor sizes. First, we can notice a correlation between the number of transmitted packets and the harvested power, with both of them increasing between 15:30 and 18:30, when the device was lighted by direct sunlight. Indeed, higher harvested power allows the ED to charge the capacitor faster, thus maintaining a voltage above the threshold set for transmissions (energy-aware approaches), or preventing the switch off. Secondly, we compare the performance for two capacitance values, i.e., C=20 mF and C=100 mF. As expected, the larger the capacitor, the longer it takes to charge, as it can be seen comparing Figs. 6.11c, 6.11d and observing the number of packets sent over time. Indeed, in Fig. 6.11b, for each scheduling algorithm, there is an initial part of the simulation where no packets are sent. Then, when the ED reaches V_{th_high}, there is a spike of sent packets, since the high voltage (3 V) reached is above the threshold set by the sender, and makes it possible to transmit more packets in a short time. After this initial transitory phase, from all the graphs in Fig. 6.11, we can see that CS and the optimal OS algorithms transmit packets rather homogeneously in time, while the other scheduling algorithms make the device switch off rather often, particularly when the energy harvesting rate is low. In particular, US yields periodic switching off of the device because it transmits regardless of the energy level; however, when the capacitor is large, it can operate it can keep the device operational for a longer period, during which it can transmit more than what allowed by a small capacitance. However, also OFF periods are longer, which results in the larger gaps between transmissions for C=100 mF than for C=20 mF. Similar considerations hold for FS and AS. In particular, given the high variability of the energy harvested in time, the moving average estimate of the harvested power over a window of few seconds is not a good predictor of near future harvested power. As such,



(a) Number of transmitted packets for different algorithms, C=20 mF.

(b) Number of transmitted packets for different algorithms, C=100 mF.

(c) Fraction of time spent in different states, C=20 mF.

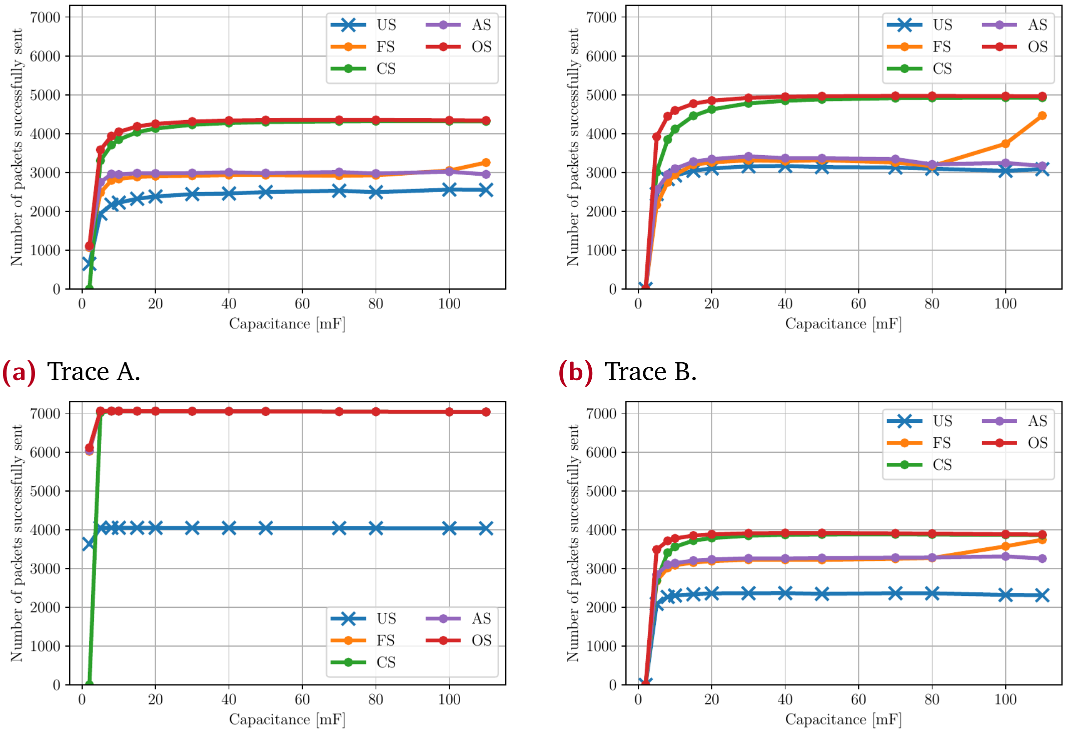
(d) Fraction of time spent in different states, C=100 mF.

Figure 6.11.: Comparison of scheduling behaviors for different capacitances, using trace A.

AS frequently overestimates the harvested power, causing the capacitor voltage to drop below the lower voltage threshold and thus making the device turn off. It is worth noting that, a more accurate prediction of harvesting power would provide some gain, but in these scenarios the CS approach is already so close to optimal that the possible gain when using other prediction techniques would be minimal.

Performance evaluation

In Fig. 6.12 we explore the joint impact of capacitor size and harvested power (different traces) on the number of packets successfully sent by the different packet scheduling approaches. First, we can notice that, for all energy-harvesting traces, the number of sent packets increases with the capacitance up to values around 40 mF, though the absolute number of transmitted packets depends on the harvesting rate. Indeed, for trace C, which has the highest mean harvested power (cf. Tab. 6.3), the number of successfully sent packets



(a) Trace A.

(b) Trace B.

(c) Trace C.

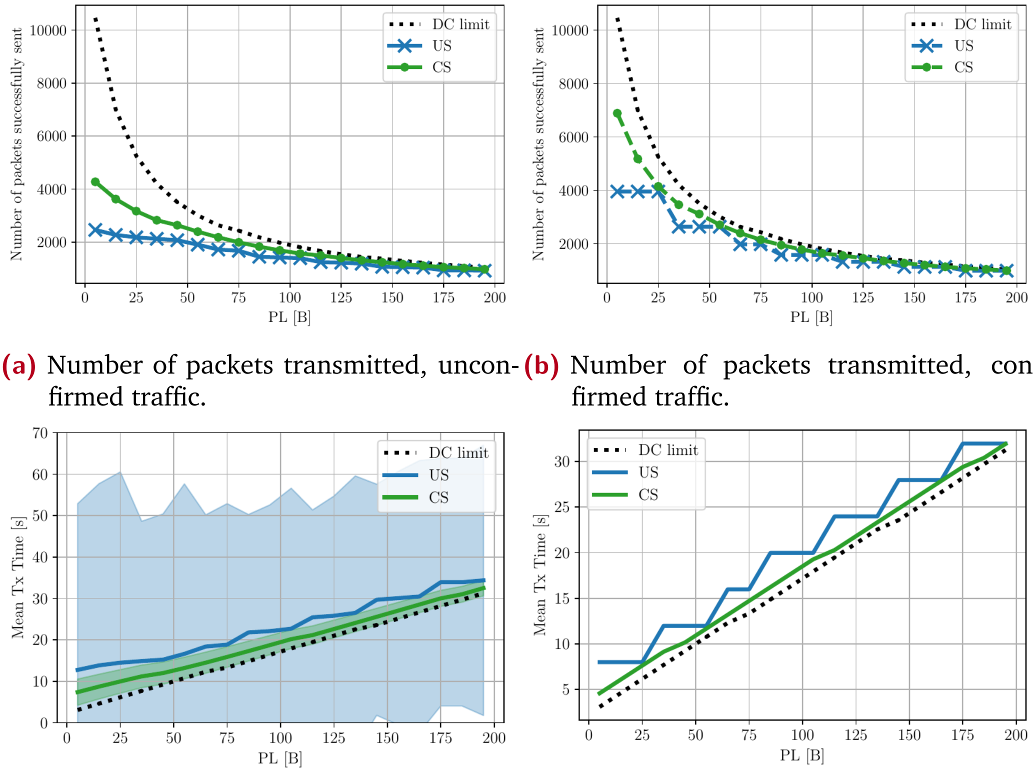
(d) Trace D.

Figure 6.12.: Comparison of scheduling behavior in terms of number of successfully sent packets, for different capacitances and power harvesting traces, PL=5 B.

reaches 7000, while in the other scenarios it does not exceed 5000. The P_h variability plays a role, too. Trace A has a higher mean P_h, but also higher variability, as it was observed in Fig. 6.11. Indeed, periods with low P_h correspond to a low amount of transmitted packets, so that the performance obtained with trace A is lower than that obtained with trace B.

From Fig. 6.12 we can also appreciate the difference between the scheduling approaches. US always performs the worst, transmitting about 40% less packets than the OS. The reason is twofold: on the one hand, the ED turns off more often; on the other hand, the scheduler drops packets that cannot be transmitted because of DC constraints, waiting I more seconds before generating a new packet, as will be better investigated below. CS, instead, is able to achieve a performance very close to the optimal because, due to the higher threshold value, the ED never turns off. Notice also that, for very small capacitors, no packets are transmitted: because CS sets a higher voltage threshold, that is not achievable by a device equipped with only a small 2 mF capacitor assuming no harvesting during the cycle (as conservatively done by CS). The performance of FS and AS algorithms, instead, are similar, both transmitting between 70 and 80% of the packets with respect to OS. FS also transmits more packets for high capacitance values, due to the fact that, in that case, the capacitor will discharge slowly, staying above the 1.82 V level for a longer time and without falling below V_{th_low}. Other results, not reported here due to space constraints, also compared different versions of the AS approach, which considered different time-averaging intervals for the harvesting power prediction, i.e., 1, 5, 30 and 300 seconds. The performance difference, however, turned out to be negligible, except for trace C, where $x = 300$ s reached slightly better performance, thanks to the slower variations of the energy harvesting process in of this trace, which yields better moving-average estimates with longer averaging intervals.

We now explore the impact of PL and, implicitly, of the DC limitations, on the device capabilities, for unconfirmed (Figs. 6.13a, 6.13c) and confirmed traffic (Figs. 6.13b, 6.13d). To obtain these results, we selected the CS approach, and a capacitance of 40 mF, which, according to the previous discussion, was the minimum value obtaining good performance for all 4 traces. In Fig. 6.13 we compare the benchmark approach (US), with the CS solution, plotting also the performance bounds imposed by DC limitations. From Fig. 6.13a, we can see that the number of successfully sent packets decreases for increasing values of PL, with US and CS both converging to the limit imposed by the DC. It is interesting to notice that, from Fig. 6.13b, the best performance is obtained when transmitting confirmed packet. This is somehow counter intuitive, since the ACK reception consumes energy. However, a successful ACK reception in RX1 prevents the opening of RX2, thus reducing the energy expenditure and allowing more packets to be transmitted. We can also notice that, when employing confirmed traffic and US (Fig. 6.13b), curves have a step shape. This happens because US blindly transmits packets with a fixed period. Therefore, if the packet transmission is prevented because of the DC, the following transmission opportunity will occur I seconds later. This is visible, in particular, by inspecting the mean transmission time (see Fig. 6.13d), and it yields in a step-shaped curve also in the number of sent packets. Note that this behavior is not clearly visible for US with unconfirmed traffic because of the higher mean transmission time. Also, in Fig. 6.13c the colored regions represent the variability of the mean transmission time: it is apparent that, while for CS the packets are transmitted quite regularly, with US there is no a clear transmission pattern, and intervals between transmissions can be sometimes very long due to the device switching off, as already observed in Fig. 6.11. Instead, confirmed traffic yields better results for both metrics, and



(a) Number of packets transmitted, unconfirmed traffic.

(b) Number of packets transmitted, confirmed traffic.

(c) Mean transmission time, unconfirmed traffic.

(d) Mean transmission time, confirmed traffic.

Figure 6.13.: Performance comparison for different PL sizes, using trace A, C=40 mF.

makes it possible to have a mean inter-transmission time closer to the DC limit, with negligible variance, in the order of tens of milliseconds, with no switching off of the device even for the US approach. Finally, we should also consider that, although the DC regulation limits the throughput performance, it provides advantages for energy-harvesting approaches, since by spacing transmissions apart, it allows the device to stay in sleep mode, charging the capacitor with minimum current consumption.

These results confirm the benefits of using confirmed traffic to reduce the energy consumption, as already observed in other literature works, such as [160]. However, for high traffic loads, ACK transmission may be prevented by the limited GW capabilities, as observed in [84], and the use of confirmed traffic may only produce packet re-transmissions, causing a dramatic drop on the devices' energy level, and in the system performance. Therefore, we tested a solution where, instead of using confirmed traffic, the sender transmits unconfirmed packets, but RX2 is set to use the same SF as RX1, i.e., SF 7 in our simulations. Results (not reported here for space constraints) showed that this approach achieves basically the same performance as using

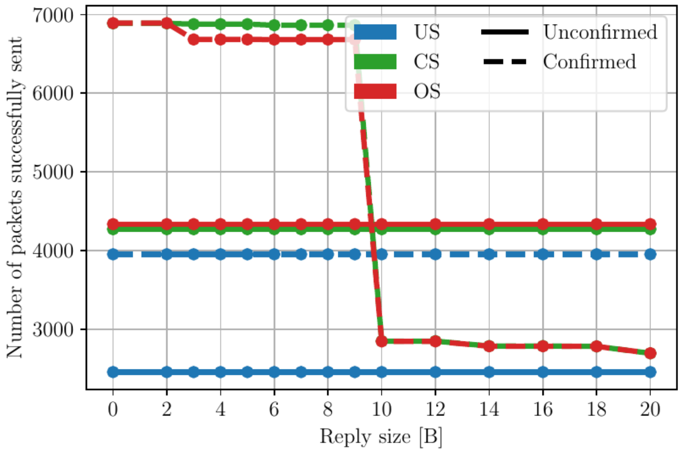


Figure 6.14.: Number of successfully sent packets for different sizes of the DL packet, using trace A, C=40 mF.

confirmed traffic, pointing out that the long duration of RX2 is a major cause of energy consumption. This solution can be easily implemented in real systems by modifying the expected value of RX2's SF through appropriate NS commands. Note that, in general, the ED will use the lower SF that makes the communication with the GW possible, and the SF for RX2 should be consistent with that. When using SFs different from SF 7, although the energy improvement is not maximized, the usage of a lower SF for RX2 still represents an improvement compared to the standard behavior.

Finally, we inspected the impact of the DL packet size. Notice that, in general, the ED can not be aware of it, and in the threshold computation, when a DL message is expected, the ACK is assumed to carry no payload. As observed in Fig. 6.14, for some payload lengths of the reply packet (2-9 B), CS outperforms OS, since the conservative assumption on no harvested power makes it possible to store additional energy, which is then employed for packet reception. However, from the figure, it can be noted that using confirmed traffic is beneficial only for DL packet sizes below 10 B, after which the assumption of a DL packet with no payload has a significant impact on the estimate of the energy consumption, and the usage of unconfirmed traffic is preferable.

6.6 Experiments with real devices

During the collaboration with the IDLab research group of the University of Antwerp, BE, it has been possible to test the feasibility of a battery-less approach for LoRaWAN nodes with real hardware. Some first experiments had the objective of evaluating the energy-unaware approach. We recall that,

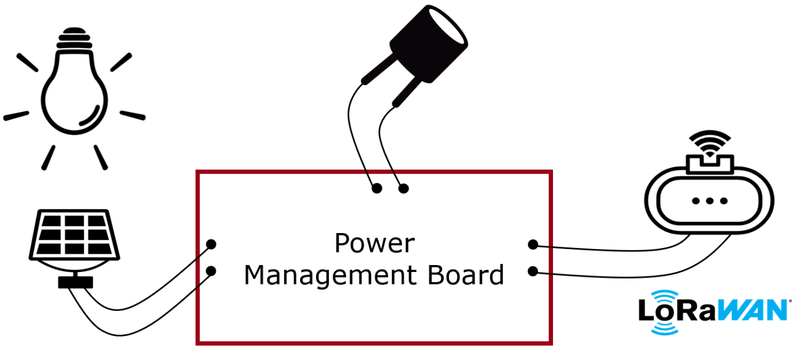


Figure 6.15.: Scheme of the experimental setup.

in this case, the device switches on and starts to transmit UL packets as soon as enough energy is stored in the capacitor, If the energy is not sufficient to maintain it operational, the device switches off. Note that, differently from the setup modeled and described in Sec. 6.2, in these real experiments we needed a power management board to connect the different components of the circuit and provide the device with a constant voltage. Indeed, the device would not work properly if powered with variable voltage.

Preliminary investigations showed that the natural sunlight was an unreliable energy source due to its inconsistent behavior, and the corresponding harvesting power was usually not high enough even to guarantee the switching on of the device, also due to the inconsistent weather conditions. Therefore, we opted for a solution where the solar panel was powered by artificial light: in this way we could get a constant harvesting power, which allowed us to perform experiments in a more controllable manner. In the following, we describe the experimental setup and employed hardware, and present the obtained results.

6.6.1 Experimental setup

To perform experiments with battery-less LoRaWAN nodes we used the experimental setup depicted in Fig. 6.15 and Fig. 6.16, placing it in a dark room to prevent other light sources from affecting the correctness of the experiments. We employed the following hardware:

Power Management Board We employed the integrated energy management circuit e-peas AEM1091, which extracts DC power from a solar panel, stores the energy in a rechargeable energy storage element, and supplies the system with a regulated voltage, whose values can be configured. Note also that the employed solar panel performs the tracking

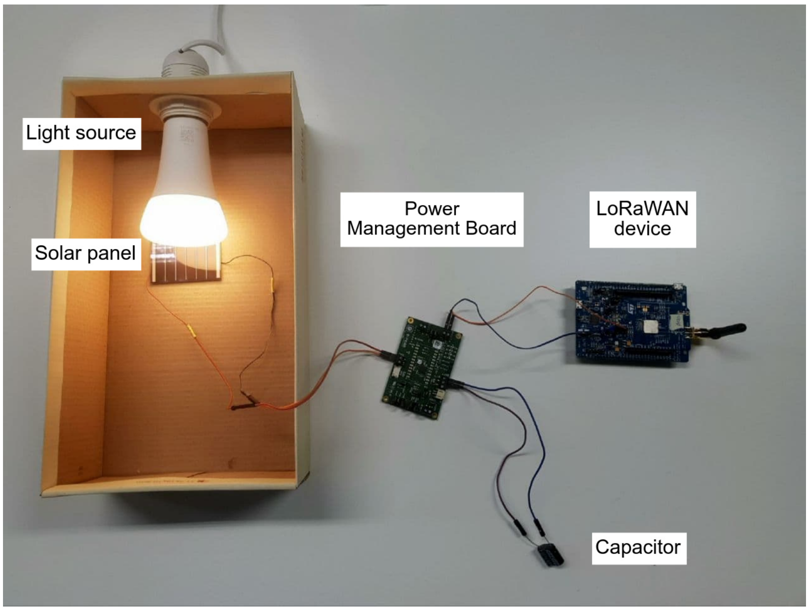


Figure 6.16.: Picture of the experimental setup.

of the maximum power point, maximizing energy extraction. For our setup, we configured this circuit to supply constant 3.3 V. Among the possible configurations of the working intervals provided by the board, we selected that where the output power is provided only when the voltage stored in the capacitor is in the interval [3.60 V, 4.12 V]. Additionaly, the minimum voltage required on the storage element before enabling the output is 4.04 V.

Solar panel We employed a Panasonic AM-5608 solar panel, having size of 60.1 x 40.3 mm, and consisting of six amorphous silicon solar cells.

Storage element We employ capacitors of different sizes: 0.1 F, 0.5 F, 1.0 F, 2.5 F.

LoRaWAN device The LoRaWAN node we used is the LoRa Discovery kit B-L072Z-LRWAN1 [173], which employes the ultra-low-power STM32L072CZ microcontroller and the SX1276 LoRa transceiver [29]. The board has been configured to work at 3.3 V, desoldering the bridges specified in the manual. In the board, we flashed an application transmitting messages according to predefined configurations. The measured values of the current consumption of the board with this setup, powered at 3.3 V, are reported in Tab. 6.4. Note that it may be possible to further reduce the current consumption in Standby state: in this way, also lower values of the harvested power may be sufficient to support the device functioning.

Table 6.4.: Measured current consumption in the different states.

State	Current consumption [mA] average	max
Off	0	0
Standby	3.90	18.65
Tx	40.42	53.64
Rx	14.86	30.37

Table 6.5.: Voltage and current provided by the solar panel for different light intensities and corresponding power.

Light intensity	Voltage [V]	Current [mA]	Power [mW]
25%	4.44	8.9	4.80
50%	4.72	9.5	5.13
100%	5.05	10.2	5.51

When powered, the device is connected to The Things Network [36] with the ABP joining procedure. Data collected from the GW are recorded using a Python script that retrieves data from The Things Network server through the Paho Message Queue Telemetry Transport (MQTT) protocol. Also note that the ED was placed at a distance of some meters from the closest GW, preventing packet losses due to interference and channel impairments.

Energy source To obtain a constant harvested power we employed a dimmable Philips Hue light bulb A27 providing warm white light (2700 K) and 1600 lm. In Tab. 6.5 we report the voltage and the corresponding power provided by the solar panel placed below the light for different values of light intensity.

6.6.2 Results

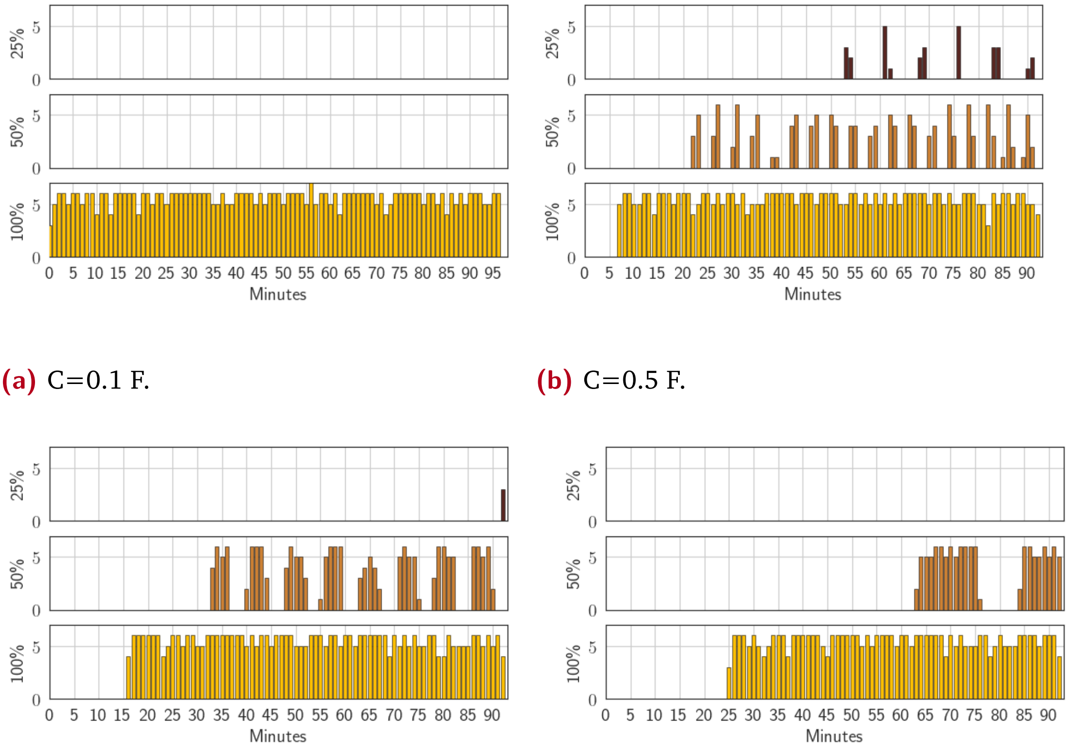
The experiments aimed to evaluate the feasibility of the battery-less solution with LoRaWAN devices, investigating the parameters affecting the performance, as previously evaluated through simulation studies (see Sec. 6.4). The explored configurations are reported in Tab. 6.6. In particular, the application flashed into the device started functioning as soon as the LoRaWAN node was powered up, transmitting unconfirmed data with a fixed SF for the whole duration of the experiment. The interval between packet generation is called *appPeriod* and can be 10 s or 20 s according to the experiment of interest. The packets had a MAC payload of 2 B, corresponding to a packet time on air duration

Table 6.6.: Parameters configuration.

Parameter	Value
Experiment duration	1:33 h / 1:00 h
appPeriod	10 s, 20 s
Packet time on air	46.336 - 61.696 ms
Spreading factor	7, 8
Capacitance	0.1, 0.5, 1.0, 2.5 F
Light intensity	25, 50, 100%

of 51.456 ms when transmitting with SF 7. However, it was observed that, in some cases, the packet time on air reported in the data collected from the GW varied during the same experiment, also taking values of 46.336 ms and 61.696 ms, although settings and configurations remained the same. In this first test campaign, we opted for a short packet duration (small payload and SF 7) to start with low requirements for the energy required for the transmission. Future work can also evaluate the impact of other configurations of these values.

In Fig. 6.17 we report the number of packets that have been received at the GW for different light intensities and capacitor sizes, starting from empty capacitors ($V_0 = 0$ V), with appPeriod=10 s. Each bar represents the number of packets sent during one minutes, which, given the duration of appPeriod can be 6 at most. Note that the behavior is generally consistent with that observed in Sec. 6.5.5, with the duration of the initial charging phase depending on the capacitor size, and the intermittent behavior for lower light intensities (corresponding to low harvesting powers). Interestingy, when the harvesting power is sufficiently high, also smaller capacitors are able to support an almost continuous functioning of the device (this was experimentaly observed also with capacitors of even smaller size). However, when the harvesting power decreases, there are no packets transmitted during the duration of the whole experiment (see Fig. 6.17, light intensity equal to 25%, 50%). Indeed, in this case, when the voltage level required to turn on the device is achieved, the starting of the turning on phase quickly depletes it, due to the small capacitor's size, preventing the node to correctly switch to the operational mode. As expected, for lower values of the harvesting power (50% of light intensity) the ON phase lasts longer when the capacitor size is larger. At the same time, also the charging time is longer, and this is particularly visible when the light intensity is equal to 25%: in the case of C=2.5 F the device is not even switching on during the duration of the whole experiment.



(a) C=0.1 F.

(b) C=0.5 F.

(c) C=1 F.

(d) C=2.5 F.

Figure 6.17.: Number of received packets for different values of light intensity and capacitor sizes, appPeriod = 10 s, SF 7, V_0 = 0 V.

In Fig. 6.18 we see the cumulative number of received packets for the different configurations. As expected, a major role is played by the light intensity: the higher its value, the more packets are transmitted for any capacitor size. However, we can notice that the charging phase really impacts the performance, and the larger the capacitor, the less packets are transmitted. However, when the light intensity is equal to 50%, increasing the capacity from 0.5 to 1 F, brings some benefits because of the longer duration of the ON phase; instead, with C=2.5 F, the effect of the initial charging phase becomes dominant.

To remove the impact of the initial charging phase, the next results focus on the activity of capacitor and device from the first transmission i.e., the first time the capacitor's voltage reaches the 4.04 V threshold required to activate the power managment board, and considered the performance over a 60 minutes period.

As we can observe from Fig. 6.19 and Fig. 6.20, when the light intensity is 100% and a high amount of power is harvested, similar performance are obtained for any capacitance value, since the device is always operational. Conversly, with lower values of the harvested power, the bigger the capacitor,

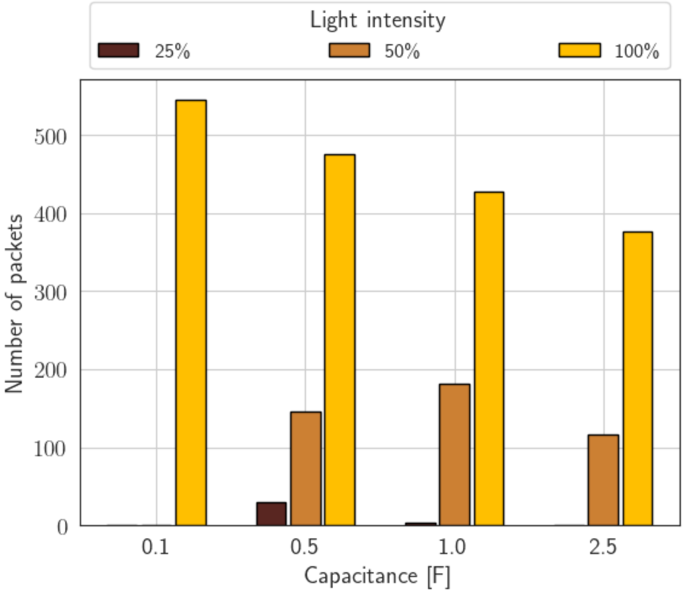


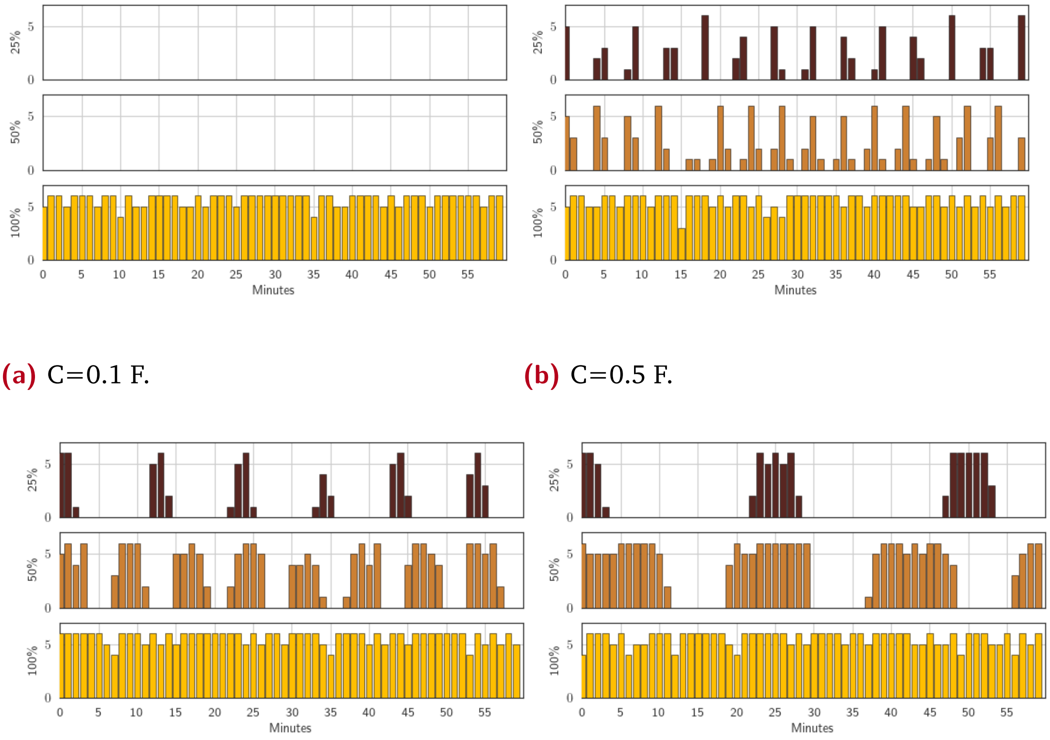
Figure 6.18.: Total number of packets received, when using SF 7 and appPeriod=10 s.

the higher the number of packets that can be sent, because of the slower depletion of the stored energy.

Then, we explored the impact of the SF. Since by increasing the SF, the packet time on air duration doubles, to have a fair comparison, we increased the appPeriod to 20 s, and run experiments for SF 7 and 8, with capacitance equal to 0.5 F. In this case the maximum number of packets that can be sent in one minute is 3: thus, with maximum harvesting power (light intensity of 100%), the number of total packets that is sent is approximately half that in the case with appPeriod=10 s (e.g., Fig. 6.20 and Fig. 6.22a). Interestingly, instead, for lower values of the light intensity, the number of packets sent for appPeriod=20 s is higher than half of that obtained when appPeriod=10 s. This is probably because the capacitor can recharge more during the longer transmission interval, preventing some switching off events of the device.

Then, we can compare the results for SF 7 and SF 8 when appPeriod=20 s (Fig. 6.21 and Fig. 6.22). As it can be seen from the plots, the increase of the SF in this scenario does not have an impact on the performance, and the number of packets that is sent is approximately the same. In this case, therefore, doubling the packet duration does not affect much the energy consumption of the device, and the tested harvesting powers are sufficient to recover from the increased energy drain during the 20 s time interval between consecutive packets, with no additional switching off of the device.

The limited effect of the SF is probably due to the small packet size, which limits the impact of the energy drain during the transmission. However, it is



(a) C=0.1 F.

(b) C=0.5 F.

(c) C=1 F.

(d) C=2.5 F.

Figure 6.19.: Number of received packets, since the first transmission (V_0 = 4.04 V) for different values of light intensity and capacitor sizes, SF 7, appPeriod=10 s.

likely that if the current consumption in the standby phase was a few orders of magnitude lower, the transmission phase would have a larger impact on the total energy consumption, and the choice of the SF, as well as the packet length, would be the determining factors for the device performance.

The preliminary experimental results showed in this section represent a proof of concept of the battery-less solution for LoRaWAN devices, demonstrating the feasibility of this approach. However, these evaluations can be further expanded with a deeper comprehension of the different components of the evaluation board, to minimize the energy consumption, by testing other configurations and implementing an energy-aware packet scheduler to make the best use of the available energy sources, as described in Sec. 6.5.

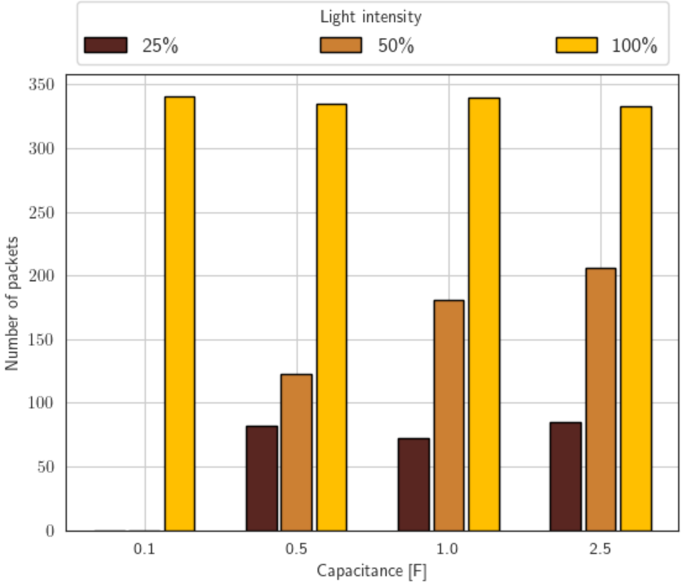
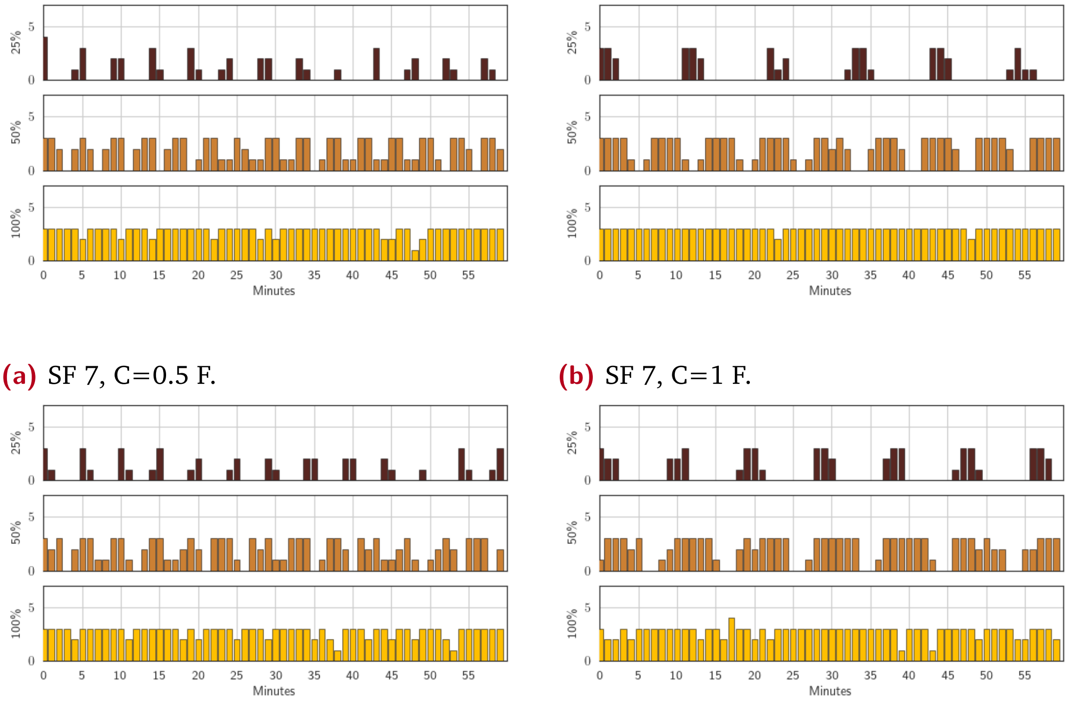


Figure 6.20.: Total number of packets received since the first transmission, for different values of light intensity and capacitor sizes, SF 7 and appPeriod=10 s.



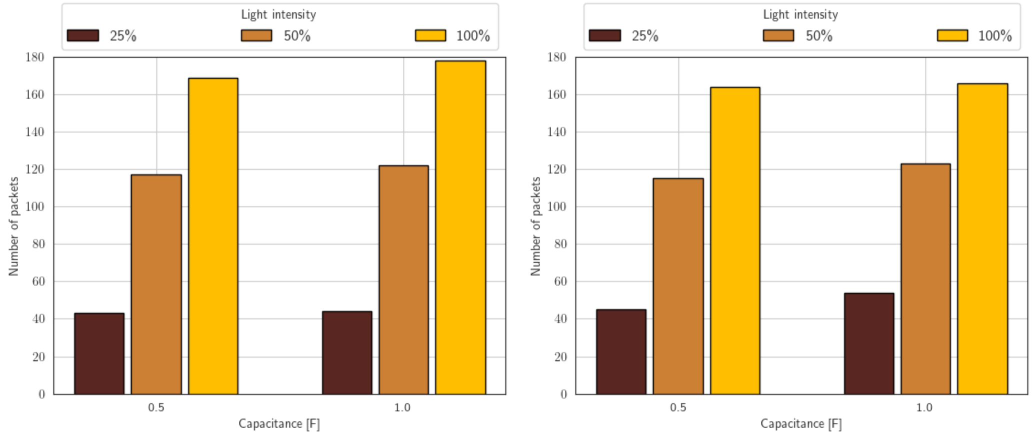
(a) SF 7, C=0.5 F.

(b) SF 7, C=1 F.

(c) SF 7, C=0.5 F.

(d) SF 8, C=1 F.

Figure 6.21.: Number of received packets since the first transmission, for different values of light intensity and capacitor sizes, appPeriod=20 s.



(a) SF 7. **(b)** SF 8.

Figure 6.22.: Total number of packets received since the first transmission, for different values of light intensity and capacitor sizes, appPeriod=20 s.

6.7 Conclusions

In this chapter, we have considered the problem of powering IoT devices with batteries, and contemplated the feasibility of a battery-less, capacitor-based approach. To this aim, we described an ns-3 implementation of a (super-) capacitor IoT node that considers the presence of a generic harvesting source to power battery-less devices. We applied it to LoRaWAN EDs, evaluating the feasibility of this solution. From the results obtained, it is apparent the impact of the harvested power and capacitor size on the communication performance, as well as the effect of the technology settings on the energy requirements.

Furthermore, we proposed some energy-aware algorithms for packets scheduling, and provided simulative results assessing the impact of the environmental scenario (available energy), and design choices (capacitor size, message payload, sender application) on the capability of the node to send packets. Although the availability of harvested power has a major impact on the performance, we identified that a good behavior could be obtained by employing a capacitor of C=40 mF size with the CS approach, which, thanks to a conservative assumption on the harvested power, prevents the ED from switching off, reaching performance close to the optimal. We showed that, although larger PL sizes increase the transmission time, the effect on the performance is mostly limited by the DC constraint which, on one hand bounds the amount of packets that can be transmitted, but, on the other hand, helps charging the capacitor by imposing silent periods. Finally, energy and communication performance can be increased with the usage of confirmed traffic or by controlling the SF employed in RX2.

Finally, the battery-less solution has also been investigated through empirical measurements, for which we showed here preliminary results. In particular when using real devices, it has been noted the impact of the environmental conditions (available solar energy), and device settings (excessive current consumption due to unused components of the device, capacitor size), which can be determining factors for the proper functioning of the system.

The resuts presented in this chapter can be extended following different directions. First, simulations can help in assessing the performance of LoRaWAN networks with battery-less devices. When considering a network with multiple battery-less devices with harvesting capabilities, it can be noted that EDs located close enough will likely have similar power availability, which may lead to correlated transmission patterns, possibly increasing the collision probability. One solution could be the usage of different SFs which, however, will impact on the device's energy (and, thus, communication) performance. Our future work will therefore consider the analysis of this trade-off in large LoRaWAN networks, and investigate possible improvements and solutions. Simulations couls also be leveraged to test smart ADR algorithms which allocate SFs considering both the network performance and also the stored voltage and the harvesting capabilities. Furthermore, more accurate preditction algorithms for energy availability could be investigated in other scenarios, possibly showin the efficancy of other solutions for packet generation.

Secondly, the experimental evaluation can embrace the evaluation of further communication and device settings, as well as implementing smarter approaches for packet transmission and tasks scheduling accorfding to the energy availabilty.

7 Dissecting energy consumption of NB-IoT devices empirically

Among the main requirements, IoT devices feature low power capabilities, that allow the minimization of maintenance costs and the extension of devices' lifetime. Therefore, although many communication technologies and device manufacturers claim to provide power efficient solutions, it is recommended to test their capabilities before actual installation, to identify possible flaws and find ameliorations and/or parameter configurations that enhance energy performance. In this section, we analyze the energy performance for two of the most common IoT technologies, namely LoRaWAN and NB-IoT.

In this chapter we focus on NB-IoT, a communication standard recently introduced by 3GPP, which offers a robust and energy efficient connectivity option to the rapidly expanding market of IoT devices. To unleash the full potential of the technology, devices are expected to work in a plug and play fashion, with zero or minimal configuration of parameters, still exhibiting excellent energy efficiency. In the following, we present the results of an extensive experimental campaign comparing the energy consumption of two commercial IoT devices and different operators, and quantifying the impact of several parameters. Furthermore, the analysis also provides insights on the effects of the parameters affecting the current consumption, proving that a proper configuration is actually necessary, and also sheds light on the relation between the channel quality and energy consumption.

The remainder of this chapter is structured as follows: in Sec. 7.1 we give a general overview, providing the context and explaining the main concerns that motivated this study. Sec. 7.2 describes our experiment workflow, while Secs. 7.5 and 7.6 break down the parameters that affect energy consumption. Sec. 7.7 discusses network KPIs. Sec. 7.8 looks into how the above affect typical NB-IoT use cases, Sec. 7.9 condenses the existing literature and Sec. 7.10 draws the final conclusions.

7.1 Introduction

The recent explosion in the number of IoT devices has been supported by a few proprietary low power wide area systems, which rely on unlicensed spectrum. Their popularity caused 3GPP to investigate cellular IoT technologies, resulting in the development of LTE-M and NB-IoT standards. A main focus area for these technologies is high energy efficiency, enabling devices operated by tiny batteries to operate for prolonged periods of time. The main advantages of the 3GPP standards are the use of licensed spectrum and the fact that they build upon existing 3GPP technologies, allowing for more stable and predictable performance, and reuse of infrastructure. LTE-M and NB-IoT are critical in enabling future 5G networks to support the density and latency requirements of massive machine type communications [174]. They can also seamlessly coexist with the upcoming New Radio (NR) access technology, since the latest standards allow the reservation of NR time-frequency resources for LTE-M and NB-IoT transmissions. During 2019, the number of LTE-M and NB-IoT connections increased by a factor of 3, reaching close to 100 million connections. These technologies are projected to account for 52% of all cellular IoT connections, by the end of 2025 [175].

In particular, we focus on NB-IoT, which provides lower throughput but more robust connectivity than LTE-M, and is hence geared towards massive deployments of IoT devices. As of January 2020 there are at least 94 NB-IoT Networks deployed commercially in over 45 markets [176, 177]. Early NB-IoT deployments have been used for smart parking, water metering and smart lighting [178]. NB-IoT modules are already integrated in devices offering solutions in areas as diverse as smart locks, intelligent road management, asset tracking, animal monitoring and smart home appliances [179].

Energy efficiency is certainly a major concern for typical IoT deployment scenarios, since batteries of IoT devices are not meant to be recharged or replaced, tying the lifetime of the battery to the lifetime of the device itself. Our analysis aims to quantify the impact of several parameters to energy consumption in NB-IoT nodes, and reveals that network configurations may greatly affect the device lifetime, without offering performance gains. For example, we show that setting a flag may reduce the energy needed to transmit an UL packet and receive a response under good signal conditions from 0.82 J to 0.12 J, with no performance penalty. In a scenario where 6 messages per day are transmitted, the device's lifetime is extended from 8.5 years to 30. NB-IoT users though, are naturally inclined to believe that, in analogy with broadband cellular services, NB-IoT services can also be accessed in a plug–and–play fashion, without or with minimal set-up of the end devices. In the same fashion, application developers should not rely on default settings but instead carefully pick parameter values that best match the tradeoff between delay and device lifetime of their use case. These findings can be further extended and used as a guideline to design algorithms or mathematical models to define the optimal parameter configuration with the correct metadata metrics (i.e., ECL, SNR, RSRP) as input. These algorithms could be applied by networks operators to tune the UE/network settings to fit a specific type of traffic, transmission frequency and requirements in an IoT application, increasing both user experience and network efficiency.

To the best of the authors' knowledge, literature research is more focused on optimizing energy efficiency through improved scheduling strategies or predicting traffic needs, which have general applicability [180, 181, 182]. The purpose of this analysis is to go beyond these and other early studies of empirical NB-IoT performance characterization, most notably [183, 184, 185], whose findings and limitations are discussed in detail in Sec. 7.8. Comparatively, the experiments presented here are more comprehensive: we 1) test more operators and / or more modules, thus revealing inefficiencies of specific module-operator combinations 2) use the latest NB-IoT features and 3) study a bigger variety of scenarios. In particular, we analyze the intricacies of operator configuration and strategies that greatly affect key metrics and battery life, while also deep diving into the performance of Release-13 enhancements. We conduct the first exhaustive experimental study of its kind, exploring the NB-IoT ecosystem, under various power management configurations. The experiments involved two different NB-IoT boards and two main telecommunication operators in a western European country, enabling us to appreciate the impact of implementation choices on the system energy efficiency. Since we focused on parameter tuning the main findings can be generalized to other networks and devices. Our measurements are spread across several months in the period between October 2018 and October 2019.

Our main contributions are:

1. A thorough presentation of the power-saving mechanisms supported by the latest commercially available NB-IoT release;

2. Experimental study of the different configurations and operator strategies, where we quantify their impact on energy consumption and network

KPIs at the Radio Resource Control (RRC) state level and, when applicable, within a state;

3. Analysis of which metadata metrics better reflect the device behavior and
4. An algorithm for extracting the state of a device directly from current time series logs.

In the sequel, we elaborate on some rather surprising findings. The energy consumption of NB-IoT devices does not seem to be strongly affected by channel conditions, except under extremely harsh conditions. Furthermore, the packet size's impact on the overall device power consumption is negligible when parameters are set to default values but becomes more significant when energy saving mechanisms are in use.

Based on such empirical observations, we recommend device-side and network-side parameter configurations that yield similar application-level performance, while preserving the device battery. We have communicated the findings to the measured operators, and they reconfigured their networks accordingly, resulting in a boost in energy efficiency for end users.

7.2 Methodology

In the following, we first discuss our experimental setup, motivating our choices with respect to: (i) experimental boards; (ii) tools for measuring energy consumption; (iii) measurement setup and (iv) collection of metadata for contextualizing the measured performance. Then, we describe the experiment scenarios.

7.2.1 Experimental setup

Experimental boards (UE) During the measurement period, both operators deployed NB-IoT using 15 KHz single-tone over band B20 (800 MHz) in Guardband. We have used two off-the-shelf NB-IoT modules, compatible with this configuration, namely u-blox SARA-N211-02B [186] and Quectel BC95-G [187]. These modules are among the first commercially available LTE Cat NB1 UEs and they have been certified by a number of mobile operators. The first module supports data rates up to 27.2 kbps in DL and 31.25 kbps in UL. Quectel BC95, when operating in single-tone, supports up to 25.2 kbps in DL and 15.625 kbps in UL. Since the form factor of these modules does not lend itself to experimentation, they

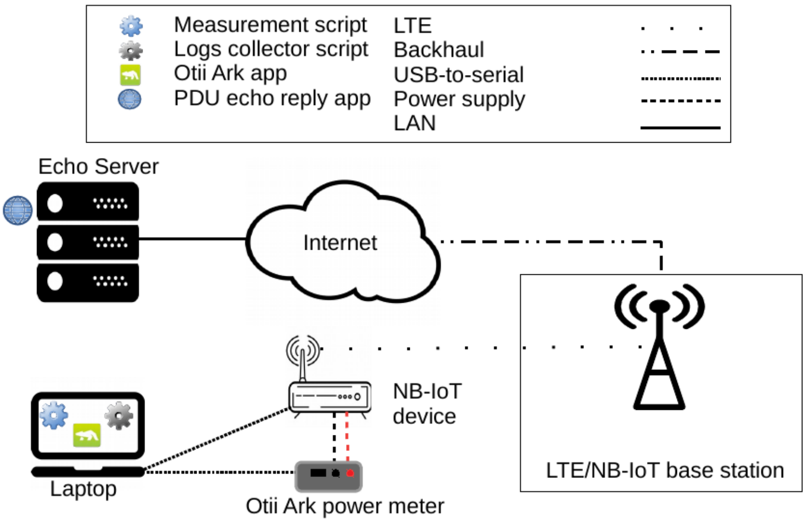


Figure 7.1.: Experiment setup.

are sold as a part of a development board (i.e., dev-kit) that facilitates powering the module and interfacing with it via USB.

Measuring energy consumption We have employed the Otii Arc power measurement device for tracking energy consumption.[1] This device can be used as both a power supply unit for the tested IoT device and a current and voltage measurement unit. It provides up to 5 V with a high resolution current measurement with a sampling rate up to 4000 samples per second in the range from 1 μA to 5 A. To characterize the energy consumption associated with different NB-IoT operations, we need to ensure that the meter measurements correspond to the current drawn by the module only, and not to the entire dev-kit. When using SARA-N211-02B, this can be obtained by powering the module directly with the Otii Arc power measurement device. Quectel BC95 does not readily allow for a similar setup. In this case, we had to remove three resistors from the dev-kit and solder a zero-ohm resistor on the power path to isolate the module power supply from the dev-kit.[2]

Measurements We have connected each dev-kit to a laptop, where we run a set of scripts to manage the NB-IoT UE's authentication, registration to the network, and RRC configuration. The NB-IoT modules use commercial subscriptions to connect to two major mobile operators in a European country. Both operators deploy NB-IoT in guard band, which reduces the

[1] https://www.qoitech.com/products/standard

[2] A zero-ohm resistor acts simply as a jumper or a wire.

likelihood of interference with LTE. To measure power consumption, we send UDP packets of various sizes (12, 20, 128, 256 and 512 bytes) to a well-provisioned server that echoes them back. The packets are sent at different frequencies depending on the experiment. Fig. 7.1 shows our experiment setup. Our goal is measuring a baseline performance, thus we avoid generating traffic through applications (e.g., Message Queuing Telemetry Transport [MQTT], Constrained Application Protocol [CoAP]), as this would add the complexity of the application on top of an already complex setup. We have repeated the measurements under various power management configurations, which we describe below. Further, we have run the experiments at various locations, which we then group based on their coverage condition into "Good coverage" and "Bad coverage". We create poor coverage conditions in two ways: 1) by using signal attenuators and 2) by placing the modules in a specially designed metallic box. This setup allows for repeatability of experiments. For some of the experiments we used a different method to simulate poor coverage in a real-life scenario: we placed the modules in a deep basement, in a similar fashion to a metering device use case. The performance at the basement is similar to the performance when using the attenuators and the special box. In these bad conditions, normal LTE mobile devices are out of coverage. Fig. 7.2b presents the RSRP values of each group of locations, which can be used as a guide to reproduce our experiments.

Data collection We use the same laptop to control both the dev-kit and the Otii Arc power measurement device. Besides measuring power consumption, we also track RTT, packet loss and throughput. We used a set of `AT` commands for collecting connection metadata. These include RRC Connection and Release events, SNR, TX Power, ECL, Physical Cell Identity (PCI), RSRP and Reference Signal Received Quality (RSRQ). In addition, we use a software called UEMonitor, developed by Quectel, to collect and decode debug messages generated by the UEs, as well as NB-IoT control plane messages such as the Downlink Control Indicator (DCI) messages.

Our measurements are spread over several months between October 2018 and October 2019, which gives us the opportunity to track the maturing of the measured deployments. Overall, we have sent about 13000 packets, which corresponds to 9 days at the rate of one packet per minute. 70% of these experiments were run using the default settings and 30% using the RAI flag. Furthermore, 75% of the experiments were conducted in good coverage conditions. Also, one third of the experiments involved sending a 20-byte UDP packet, the remaining two thirds were split among packet sizes of 12, 128, 256 and 512 bytes.

7.3 Experiments

We focus on three operation modes / scenarios corresponding to the possible setting of the RAI flag (see Tab. 2.3):

TX/RX default timers (RAI-000) The UE sends an UL packet to a remote server, which echoes it back. During this operation it remains in the Connected state, monitoring the channel for paging messages until the inactivity timer expires. Then, after the RRC Release, it enters the PSM. This scenario corresponds to applications that require two-way communication, e.g., reliable monitoring or alarm services.

TX/RX and release (RAI-400) Here, the RRC connection is released once the response from the server is received. The application scenario is again a two-way communication service. The immediate release is intended for optimizing energy consumption.

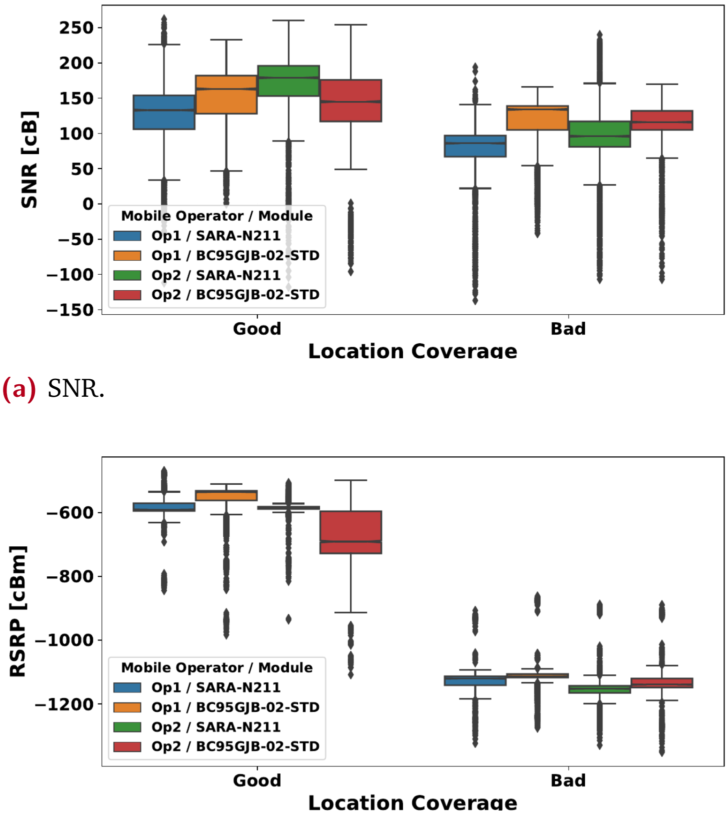
TX and release (RAI-200) In this case, the RRC connection is released after sending the UL packet, without waiting for a server response. This corresponds to services without strict reliability requirements.

As described in Sec. 2.2, each of these scenarios comprises two distinct states: Connected and Idle. We examine the energy consumption during the Connected and Idle state separately. To do this, we need to identify which state and phase the device is in at any particular time point. We present our algorithm for automatically identifying the device state from the experiment logs in Appendix C.

7.4 Metadata as a proxy for performance

It is important to collect accurate and frequent metadata, as they are an indication of performance and help diagnose problems. Both devices report metadata through AT command requests. These requests consume energy (around 15 mJ in our measurements) and may take several seconds to fulfill. The response time increases with worsening signal conditions. For instance, at locations with very bad coverage the request may time out and some of the metadata might not be reported or have obviously wrong values (e.g., SNR value of -30000). In this section, we examine the metadata reporting accuracy and investigate which metadata metrics better reflect network and energy performance, so that users can get the most value out of this costly operation. Both devices report power ratios in cB ($1dB = 10cB$) and power in cBm ($1dBm = 10cBm$). We start by comparing the behaviour of the most commonly used metadata metrics: SNR and RSRP. Fig. 7.2, presents the distributions of SNR and RSRP when we group the measurements based on the expected signal quality of the measurement location. As we will present in the sequel, the biggest effect on performance is caused by the choice of operator and module, thus in this figure and the rest of the section, we control our measurements for these two variables. The SNR distributions are very wide, with a significant overlap between the good and the bad locations. Further, the median values between the two locations show a small difference between 30 and 80 cB. In contrast, the RSRP distributions better reflect the signal quality at each location, there is significantly less overlap in the distributions and the distributions are also narrower.

SINR and RSRP are connected by $SINR = \frac{S}{I+N} = \frac{12*RSRP}{I_{tot}+N_{tot}}$ [188], where we assume that RSRP is free from noise and interference and includes only useful (reference signal) power. I_{tot} and N_{tot} are the interference and noise computed over the whole 180 kHz bandwith, and since RSRP is the power of a single 15 kHz subcarrier we multiply it by 12, which is the number of subcarriers. Both operators deploy NB-IoT in the guard band, thus there should be no interference from normal LTE traffic. Also, the measurements were performed soon after the NB-IoT was deployed, so the number of other users is very small, minimizing interference from neighboring NB-IoT cells. Thus, the main component of the denominator is noise, which is affected by temperature and the noise figure of the receiver, so we expect the noise to not fluctuate much. Under these assumptions, RSRP and SNR should have a linear relationship when expressed in cB. Fig. 7.3 shows the connection between RSRP and SNR. Assuming no interference, the red line is the ideal mapping of RSRP values to SNR for typical values of thermal noise density and receiver noise figure, $N_{thermal} = -1740cBm/Hz$ and $NF_{receiver} = 70cB$, respectively: $SNR_{cB} = RSRP_{cBm} + 1252$ (proof in Appendix C.2). This relationship is verified for the bad coverage measurements, but not for the good coverage measurements.



(a) SNR.

(b) RSRP.

Figure 7.2.: Distributions of metadata in locations with weak signal (bad coverage) and good signal (good coverage).

We briefly report our observations for the rest of the metadata. Both modules log the following metadata: RSSI, SNR, RSRP, RSRQ, ECL and TX Power. RSSI values have similar distributions to the RSRP values and are typically between -470 and -600 cBm in good locations and around -1030 cBm in bad. Thus, they are typically 60 cBm higher than RSRP in good and around 100 cBm higher than RSRP in bad conditions. The RSRQ values are around -108 cBm for good conditions and slightly worse between -108 and -113 cBm for bad conditions. RSRQ has very small variation across different conditions, making it poorly correlated with performance. Since RSSI does not provide further information over RSRP, we can safely disregard it.

Only the RSRP and RSRQ are reported to the eNodeB and from these the eNodeB can estimate the RSSI [189]. The RSRQ measurement provides additional information when RSRP is not sufficient to make a reliable handover or cell reselection decision. In contrast, RSRP is the most important metadata metric. During the Random Access Procedure (RACH), the UE sets its ECL and TX Power based on the RSRP thresholds it receives from the eNodeB. If the

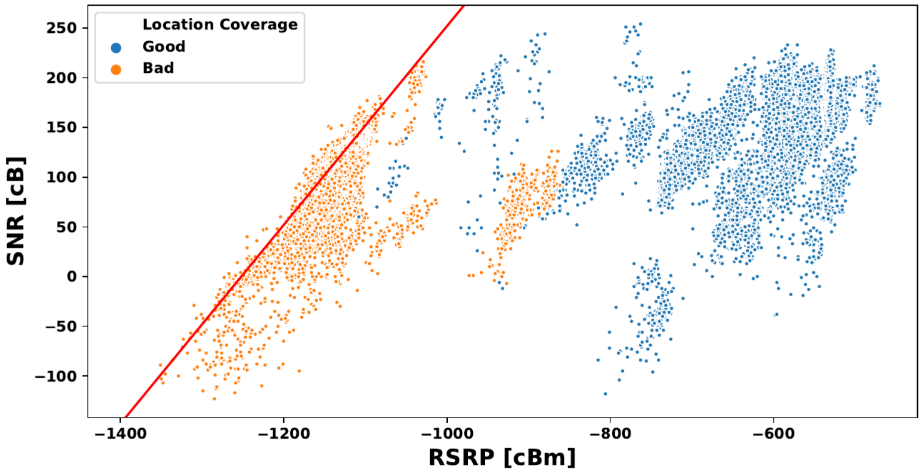


Figure 7.3.: Mapping of SNR and RSRP. The red line represents the ideal mapping assuming no interference (applies only to bad coverage data).

UE is unable to connect, it increases its TX Power by 2 dB increments, until it achieves connectivity or until it reaches a predefined number of preamble transmission attempts per ECL supported in the serving cell. Then, it increases its ECL by 1 and sets the TX Power to maximum and repeats the process. The RSRP thresholds and the number of transmission attempts per ECL are set by the operator [17]. For the above reasons, we will focus mostly on RSRP in the sequel, as it is the metric that best reflects the performance and energy consumption.

Fig. 7.4 presents the RSRP values observed for every ECL level. From this figure, we can empirically estimate the RSRP thresholds per operator. Op 2 is switching faster to higher ECL. Even though the difference between the thresholds among the operators is small, as we will present in the next sections, it has a big effect on all the KPIs. Energy consumption and other KPIs increase marginally between `ECL 0` and `ECL 1`, but deteriorate sharply between `ECL 1` and `ECL 2`, due to the huge number of repetitions and use of maximum TX Power. Some of the metrics that are affected are: device lifetime, throughput, RTT and packet loss. Using a higher ECL when not necessary, has a big impact on battery lifetime, without affecting robustness. As we will discuss in the next chapters, Op 2 performs poorly in locations with bad coverage. This is due to its more aggressive `ECL 2` threshold.

Finally, we study how TX Power is connected to ECL and RSRP in Fig. 7.5 and 7.6, respectively. In Fig. 7.5, we observe the range of possible TX Power per ECL. Empirical measurements show the minimum TX Power of a normal LTE device to be -22 dBm [190], in contrast the NB-IoT modules may transmit with as low as -290 cBm ($\approx$ -29 dBm) and the transmit values have a granularity

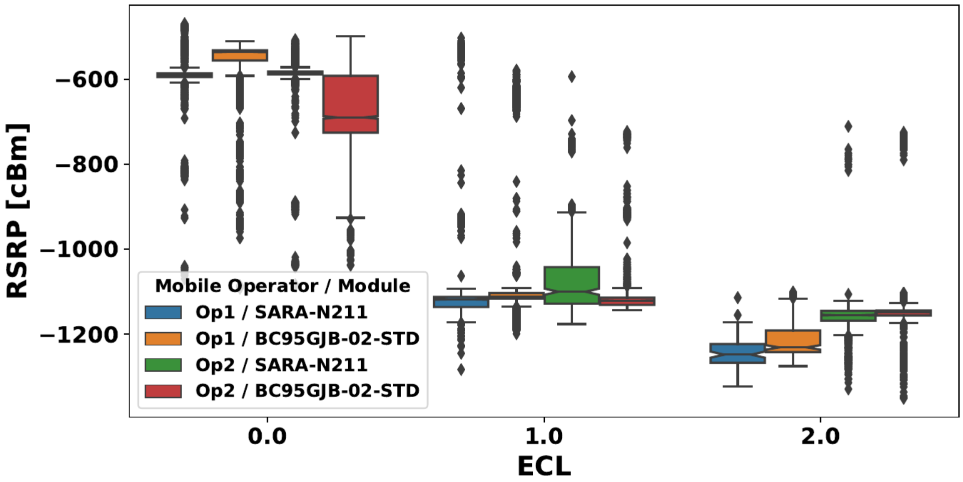


Figure 7.4.: Distributions of RSRP based on ECL value.

of 10 cBm. NB-IoT utilizes less bandwidth thus, needs less TX Power to reach similar SNR values to LTE. `ECL 0` uses the full range of values and rarely the maximum value of Cat NB1: 230 cBm. `ECL 1`uses the maximum value for 79.3% of the samples. This is due to the RACH algorithm discussed above: if the initial value is `ECL 0` and the RACH procedure fails, the UE will attempt again with `ECL 1` and maximum TX Power. As expected, `ECL 2` uses maximum power in 98.4% of the samples. Fig. 7.6 reveals a linear relationship between RSRP and TX Power and also shows the more aggressive TX Power choices of Op 2, since it usually uses higher TX Power for the same RSRP value. The linear relationship holds for the RSRP range typically associated with `ECL 0`, between -1000 and -500 cBm. Worse RSRP values, mostly related with higher ECLs, use almost exclusively maximum power.

Takeaways. We conclude that of the available metadata metrics, the most useful ones are ECL and RSRP, which are directly related. Other metrics either correlate weakly with performance or do not involve enough variability to be useful. Operators should carefully choose the mapping between ECL and RSRP.

7.5 Energy consumption in Connected state

We now turn to examining whether the actual energy consumption by NB-IoT UEs, while in Connected state, in the real world conforms with the standard behavior outlined in Sec. 2.2.

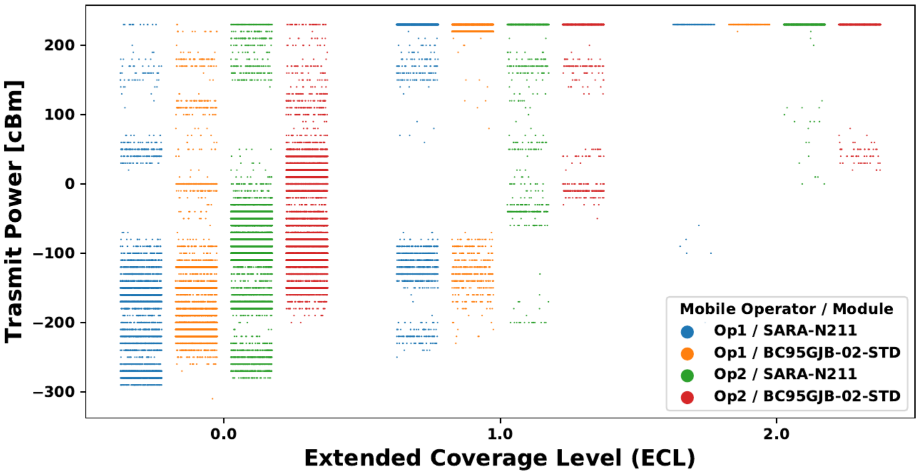


Figure 7.5.: Relationship between ECL and TX Power.

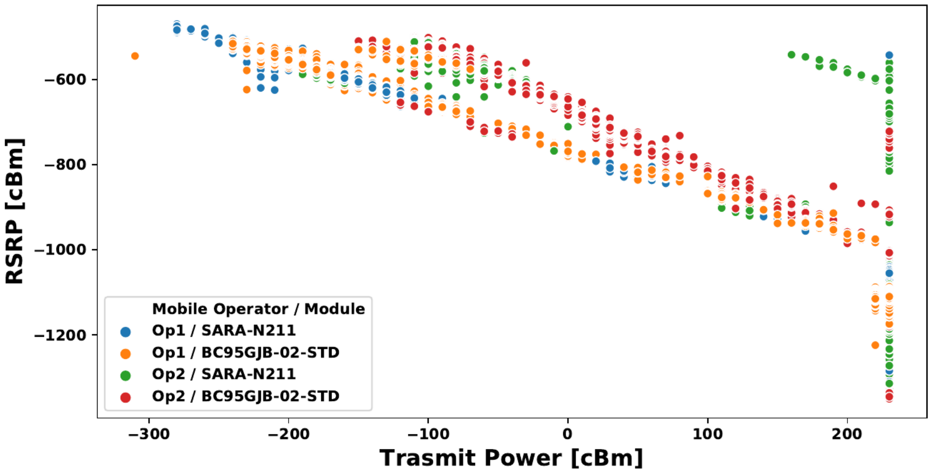


Figure 7.6.: Relationship between RSRP and TX Power.

7.5.1 Connected state with default settings

Fig. 7.7 shows the distribution of energy consumption for the first experiment scenario with no RAI (i.e., UL and DL activity with default timers, see Sec. A.2) for the different combinations of operator and module in the Connected state. In this scenario, we send a UDP packet, which is echoed back by the server. As we will show in the sequel, packet size has minimal impact in this configuration, so we include in the default settings analysis all the packet sizes. We split the dataset into two groups, depending on the coverage conditions at the location of the measurements, as discussed in Sec. 7.2.1.

Good coverage We record a clear difference between both operators and modules. Op 1's energy consumption is 3x or more than Op 2's. Also, SARA-N211 consumes more power than Quectel-BC95, though the difference depends on the operator! Tab. 7.1 presents the median energy

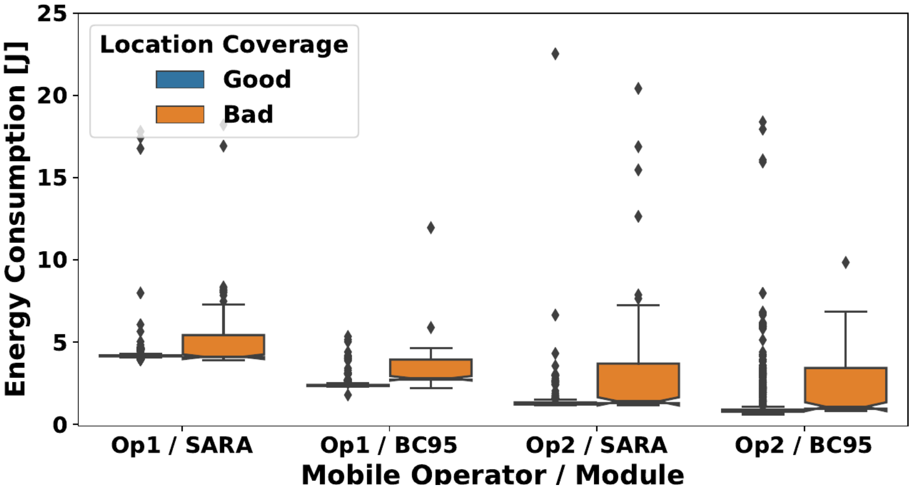


Figure 7.7.: Energy consumption distribution of Connected state, grouped by coverage conditions. All packet sizes, no RAI.

Table 7.1.: Median energy consumption of Connected state under good coverage. Includes samples of all packet sizes.

Module	Operator	Energy [J]
Quectel - BC95	Op1	2.39
	Op2	0.82
SARA - N211	Op1	4.17
	Op2	1.27

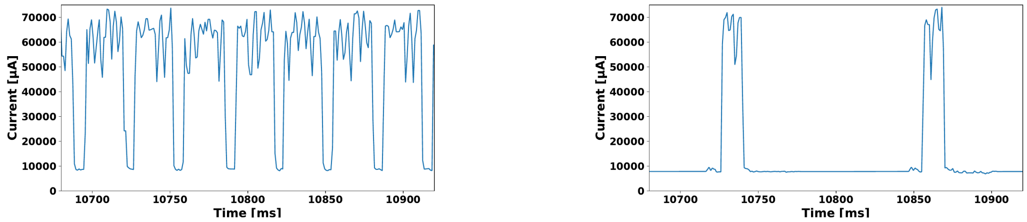
consumption for all operator module combinations. Digging deeper into our data, we find that the difference between the operators stems from the fact that Op 1 does not enforce any quiet period while paging during the inactivity timer period, like Op 2. Instead, Op 1 is mostly in a high energy paging state. Fig. 7.8a and 7.8b, illustrate the behavior of Op 1 and Op 2 respectively, during the inactivity timer.

Fig. 7.9 shows the median values of the consumed energy for every substate of the Connected state. Recall that the Connected state comprises three substates: synchronization with the network (sync), data plane transmission and reception (TX) and inactivity timer. In the inactivity timer, the UE performs paging and, ideally, enforces cDRX.

The inactivity timer substate dominates the energy consumption. Thus, it is critical to consider whether it is necessary, and if so, cDRX should be used. This also hints at that the size of the transmitted packet becoming irrelevant, since the increase in energy consumption for the extra bytes is minuscule compared to the total energy consumption of the Connected

Table 7.2.: Median energy consumption of Connected state under poor coverage. Includes samples of all packet sizes.

Module	Operator	ECL	Energy [J]
Quectel - BC95	Op1	0	2.71
		1	2.80
		2	4.04
	Op2	0	0.88
		1	1.03
		2	3.44
SARA - N211	Op1	0	4.15
		1	4.10
		2	5.50
	Op2	0	1.28
		1	1.40
		2	3.77



(a) Op 1 constantly monitors paging during the inactivity timer.

(b) Op 2 performs cDRX during the inactivity timer.

Figure 7.8.: Current draw during the inactivity timer period.

state. Fig. 7.10 illustrates the energy consumption for different packet sizes. The cost increases sub-linearly with packet size – the bigger the packet, the less energy is spent per byte. Increasing the packet size from 20 to 512 bytes results in an increase in energy consumption by a few tens of mJ, negligible when compared to the energy consumed in the inactivity timer substate, which is in the order of Joules.

Poor coverage Fig. 7.7 shows no clear differences in the median power consumption between locations with good or poor coverage. However, the latter ones are characterized by stronger variability with the inter-quartile difference several times the median. The difference in coverage results in picking different ECL levels. A higher ECL means extra repetitions when sending data, to increase the likelihood of successful delivery, which translates into a higher energy consumption. In brief, there is a slight increase in energy consumption between `ECL 0` and `ECL 1` and a big increase between `ECL 1` and `ECL 2`. The devices at good signal locations

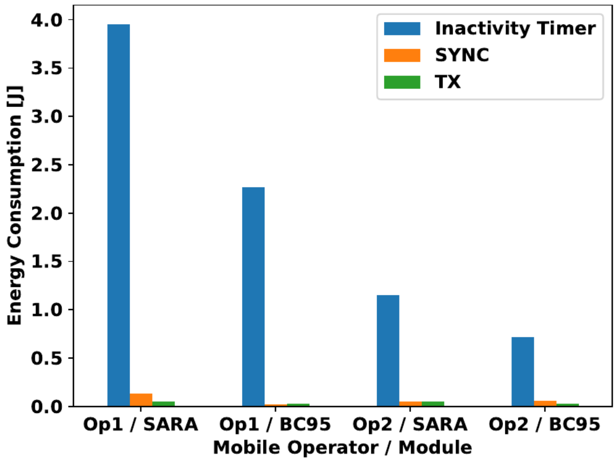


Figure 7.9.: Break down of energy consumption per Connected state substate for locations with good coverage. We used the median value of each substate. All packet sizes.

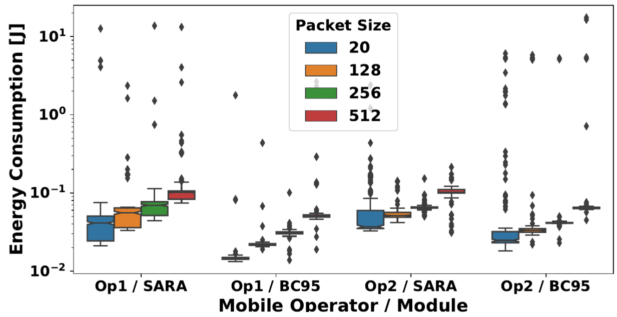


Figure 7.10.: Distributions of the consumed energy of the TX substate, grouped by the packet size of the transmission, for locations with good coverage. Y axis is in logarithmic scale.

are almost always using ECL 0 and in about 1% of the cases they use ECL 1. This explains the compactness of the energy consumption distribution. At bad signal locations the devices are on ECL 2 for about 30% of the samples, with the rest of the samples being split between ECL 0 and ECL 1. These proportions vary according to the device and operator combinations. This 70/30 split explains why the median energy consumption in areas with poor coverage is close to the median energy consumption in areas with good coverage. Tab. 7.2 presents the median energy consumption for all operator module combinations further split according to the ECL level. The values in the table confirm that the difference between poor and good coverage is chiefly evident for ECL 2.

There are small variations in the consumed energy, even between subsequent transmissions with the same ECL level, which can not be fully explained. Both operators use a fixed number of repetitions per channel across all their base stations. For example, for Uplink data transmissions the repetitions are 2, 8 and 32 for `ECL 0`, `ECL 1` and `ECL 2` respectively. Thus, the number of repetitions is not a factor in the energy consumption variation within the same ECL. We have also shown that `ECL 1` and `ECL 2` typically use the maximum TX Power and the variability increases in poor coverage, so TX Power can not be the only factor contributing in the variability within an ECL level. Upon closer inspection of the power traces, we observe that especially in bad conditions the power consumption patterns (e.g., peak current, staying in a high current draw level for longer, etc.) within substates, show big variations, even if we control for TX Power and ECL. Further, they become more pronounced with worsening signal conditions. We are not able to find the route cause of these patterns, but we assume that it is related to the module implementation. These patterns are also observed during the inactivity timer period, which is the dominant contributor to energy consumption, under default settings. Thus, under default settings, we observe higher variability in our results compared to when we use energy saving features, as we will present in the sequel.

7.5.2 Connected state with RAI

Now we move to discuss the energy consumption when the RAI flag is set.

Is the RAI flag respected? We observe that both operators may ignore the flag and proceed to perform an inactivity timer procedure. For Op 1, this is a rare occurrence, it just happened for 3 packets in our dataset. A plausible cause could be corrupt signaling packets. For Op 2, however, the RAI-200 flag was ignored for 50% of all packets of all measurements performed before April 2019, regardless of the module. More specifically, one in every two packets would consistently utilize the inactivity timer after transmission, instead of dropping to Idle state. Fig. 7.11 illustrates this behavior. The short spikes are transmissions where the flag was respected, while the periods with intense activity (e.g., the one starting around $t = 200000$) are instances where the flag was ignored. Following the discovery of this anomaly, we informed Op 2 who then corrected the misconfiguration that caused it. Fig. 7.12 shows the energy consumed when transmitting 20 bytes, while setting RAI-200, before and after our feedback to the operator. In the "before" case, the distributions are

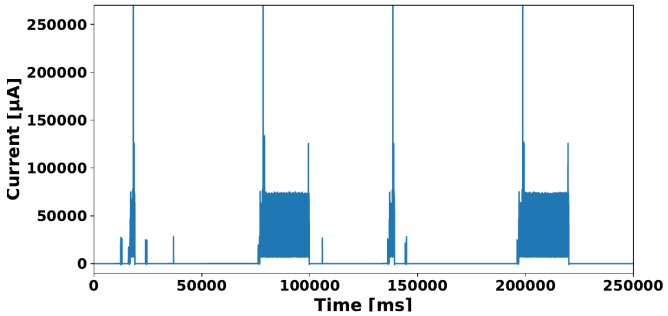


Figure 7.11.: Current draw of Op 2 over SARA-N211 when using the RAI-200 flag in early 2019. 20 bytes packets. Good coverage.

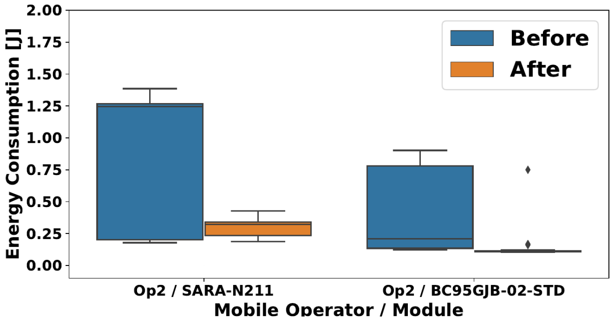


Figure 7.12.: Energy consumption of Op 2 when sending 20 bytes using RAI-200 in a location with good signal before and after our feedback.

broader, exhibiting values similar to those measured when the RAI is not in use (see Fig. 7.7). Fixing this bug has led to a reduction in the median value by 80%. The gains are even higher for larger transmissions and/or challenging signal conditions. In the rest of the section, we only present measurements collected after the correction.

Energy consumption Fig. 7.13 and 7.14 present the energy consumption for several combinations of packet sizes and coverage locations while setting the flags RAI-200 and RAI-400, respectively. Using RAI leads to great savings in energy. Tab. 7.3 presents the median energy consumption, for all operator, module and RAI combinations. The median energy consumption without RAI can be 20 times higher for Op 1 and 15 times higher for Op 2. Furthermore, the choice of the module influences the energy consumption greatly. SARA-N211 consumes about twice as much as Quectel-BC95.

Interestingly, in most cases, using RAI-400 results in only a slight extra energy consumption compared to RAI-200. This may partially be attributed to the locality of the echo server, offering short round trip

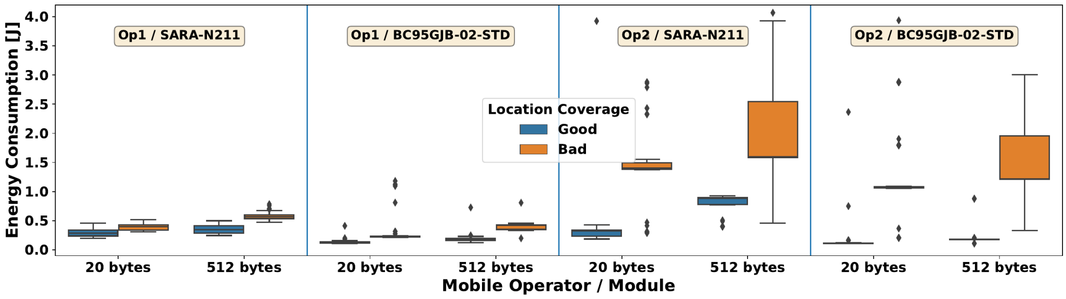


Figure 7.13.: Distribution of energy consumption for RAI-200, grouped by coverage conditions and packet size.

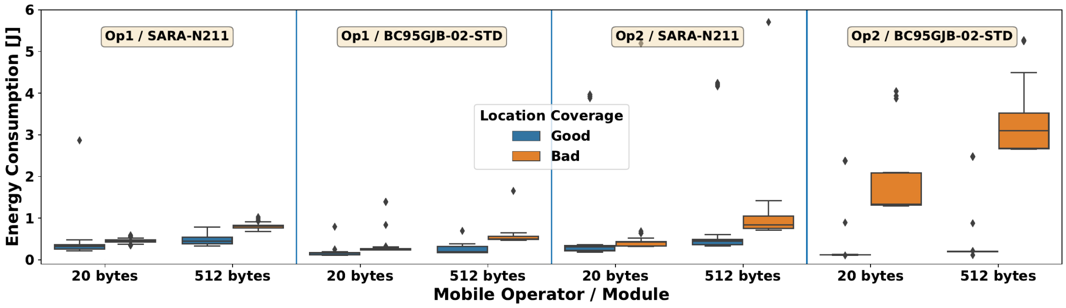


Figure 7.14.: Distribution of energy consumption for RAI-400, grouped by coverage conditions and packet size.

times, and by extension reducing the total duration of the Connected state. For example, let us consider the best performing pair of operator and module which is Op 1 and Quectel-BC95, under good conditions. When sending a 20-byte packet, the median Connected state duration (i.e., time between two Idle states), is 3.13 seconds for RAI-200 and 3.23 seconds for RAI-400. The median duration when transmitting 512 bytes remains unaffected at 3.12 seconds with RAI-200, but reaches 4.06 seconds with RAI-400. Under poor coverage, transmitting 512 bytes requires 3.94 (RAI-200) or 4.95 seconds (RAI-400).

With the inactivity timer substate being removed, the impact of the payload size increases. We have tested two very different packet sizes: 20 and 512 bytes (see Fig. 7.13 and 7.14). The larger packets result in a larger energy consumption. This increase hovers around 60% and never exceeds 100%. Hence, although payload size plays an important role, the choice of the module has more impact. For example, if we focus on the median energy consumption of Op 1 over RAI-400 (i.e., the two left quadrants of Fig. 7.14), keeping the packet size constant and changing the module from Quectel-BC95 to SARA-N211, it will result in an 115% increase in good locations and a 70% increase in bad locations. Finally,

Table 7.3.: Median energy consumption when sending 20 bytes with RAI under good coverage (Joules).

module	operator	RAI-200	RAI-400
Quectel-BC95	Op 1	0.12	0.17
	Op 2	0.11	0.12
SARA-N211	Op 1	0.27	0.31
	Op 2	0.31	0.33

we observe that Op 2 draws significantly more power at places with poor coverage, compared to Op 1. Digging deeper into this, we find that Op 2 uses ECL 2 more frequently than Op 1, as was expected based on Op 2's more aggressive ECL thresholds we detected in Fig. 7.4 of Sec. 7.4. This results in repeating each transmission several times, causing an up to tenfold increase of the overall energy cost compared to Op 1 under similar conditions.

Takeaways. The use of RAI flag leads to significant savings in energy consumption. The choice of the UE is key to energy consumption, which suggests the need for a UE certification process. Operators must thoroughly test and confirm that their implementation and configuration conform to the expected standard behavior. We have highlighted a few cases of misconfiguration that translate into excessive energy consumption. Interestingly, the measured NB-IoT deployments seem to fare well under poor coverage conditions except for the extremes. Payload size becomes important only when the RAI flag is set and the network is correctly configured.

7.6 Power consumption in Idle state

The majority of the NB-IoT devices' lifetime is spent on Idle state and mostly on PSM and the sleep phase of Idle state DRX (iDRX), if available. This section quantifies power consumption in the PSM and eDRX modes.[3] Note that these modes do not have a specific time duration, thus we present the power consumption rather than the energy.

[3] iDRX can be either DRX or eDRX (i.e., DRX with PTW and prolonged sleep periods). Our analysis applies to both, but we use eDRX in this section, since this is expected to be more popular in NB-IoT.

7.6.1 PSM

During PSM, the radio is OFF and the device is in a “deep sleep” mode. Thus, the only parameter affecting power consumption is the module itself (i.e., the combination of the hardware and firmware). Both modules consume around 10 μW, with the median values being 10.61 μW for Quectel-BC95 and 9.35 μW for SARA-N211. In rare occasions ($< 2\%$ of the PSM samples in the dataset), the modules fail to reach the typical PSM current levels of 2-5 μA, resulting in an elevated power consumption that may exceed 30 μW. Hence, the power distributions (not shown) are fairly compact, with 98% of all samples centered around the median.

7.6.2 eDRX

eDRX consists of listening and sleep phases (see Sec. 2.2.1). The *eDRXCycle* parameter determines the overall duration of an eDRX cycle, which is the time between the starting points of two consecutive listening phases. However, the total duration of the sleep phase is not standardized, because the listening phase may vary in length due to channel conditions. To estimate the energy consumption, while on eDRX, we measure the time spent listening t_{eDRX-L} as well as the consumed power P_{eDRX-L}. Multiplying these two gives the energy consumed while listening: E_{eDRX-L}. The time spent sleeping equals the total time spent in eDRX minus the time spent listening ($t_{eDRX-S} = t_{eDRX-total} - t_{eDRX-L}$). The power consumed while sleeping (P_{eDRX-S}), is in the same range as PSM. Hence, the overall energy consumption in eDRX is given by

$$E_{eDRX} = (E_{eDRX-L} + P_{eDRX-S} \cdot t_{eDRX-S}) \cdot N_{cycles}, \tag{7.1}$$

where N_{cycles} is the number of listening-sleep cycles in the eDRX mode, which can be derived by the configuration. Next, we examine each of the two phases.

Listening The duration of the listening phase depends chiefly on coverage conditions, with listening phases in bad coverage lasting significantly longer. More specifically, it is affected by the ECL, which is in turn determines the number of control channel repetitions the UE should listen. Tab. 7.4 presents the median values of t_{eDRX-L} and E_{eDRX-L} for different operator, module and coverage combinations.

Listening starts with a low power synchronization period and ends with a more power demanding period of listening to paging occasions (PO).

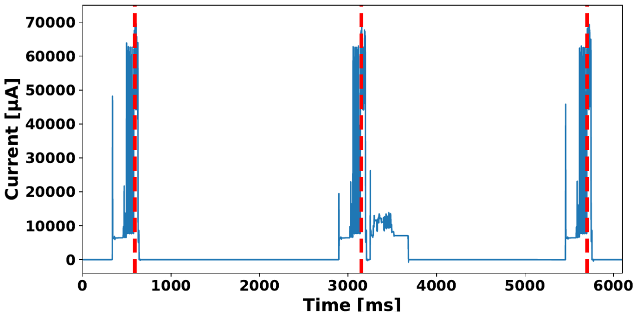


Figure 7.15.: Current draw of the buggy eDRX listening phase implementation. The listening phases should end at the dashed vertical lines.

Table 7.4.: Median values of energy consumption and duration of eDRX listening phase.

Coverage	Module	Operator	Energy [mJ]	Duration [ms]
Bad	Quectel - BC95	Op1	21.4	470.2
		Op2	24.6	476.7
	SARA - N211	Op1	33.7	536.5
		Op2	39.1	552.2
Good	Quectel - BC95	Op1	6.4	215.0
		Op2	6.3	215.2
	SARA - N211	Op1	10.3 (20.0)	224.5 (300.0)
		Op2	10.1	222.8

There is a bug observed in both devices, where they may remain at an elevated power level after the PO period ends, shown in Fig. 7.15, increasing the phase's duration and energy. The proper ending points of the PO periods are marked with red lines in the figure. This bug appeared mostly when using SARA-N211 over Op 1 and in later measurements appears much less frequently. Tab. 7.4 uses only recent measurements, where the bug is rarely observed (in the parenthesis we present the values while the bug was still frequent).

Under good conditions we do not observe any difference between the operators for the same module. The device though has a major effect on energy consumption, with SARA-N211 consuming a median 10 mJ and Quectel-BC95 a median 6 mJ. Under bad conditions, the power consumption mostly depends on the ECL. Op 2 has a tendency switching to ECL 2 faster than Op 1, as conditions get worse, and this is reflected in the energy consumption. As in good conditions, the most important factor in the energy consumption is the module, with Quectel-BC95 showing better efficiency. Tab. 7.4 also shows that listening time duration evidently increases under poor coverage.

Sleeping As with PSM, the deciding factor of the energy consumption in the sleeping phase is the module. The median of P_{eDRX-S} is 10.01 μW for SARA-N211 and 10.36 for Quectel-BC95.

Takeaways. In deep sleep, energy consumption is only affected by the choice of the module. Energy consumption while listening is determined by coverage and the choice of the module under good conditions. Under bad conditions, operator choice becomes important as well.

7.7 Network Performance: RTT, Throughput and Packet Loss

Finally, we examine the network KPIs: packet loss, RTT and throughput. Tab. 7.5 presents a summary of these metrics, as well as some of the metrics discussed in the previous sections, allowing for a complete overview of the performance.

Packet Loss In our experiments we transmit a single UDP packet to a well provisioned server and, if applicable, echo it back to the device. We embed each packet with a unique ID. If the packet never reaches the server we assume it was lost in the UL direction. In the experiments where the UE is expecting a response, if a packet reaches the server but the corresponding reply is never received by the UE, we assume a loss in the DL direction. LTE UEs (e.g., smartphones) experience almost null packet loss when they are immobile / stationary and connected to uncongested LTE networks ([191], Fig. 3.1 and Tab. 3.1). In contrast, we observe that packet loss rates in commercial NB-IoT deployments are between 0.5% and 1%. The majority of the losses happen in the UL, and worsening signal conditions cause a slight increase, as expected. Surprisingly, the more aggressive use of robust ECL levels by Op 2, does not translate into better packet delivery, compared to Op 1.

Table 7.5.: Summary of KPIs. Packet Loss is calculated among the whole dataset. Throughput and RTT are calculated for 20 bytes packets.

Mobile Operator / Module	Location Coverage	Packet Loss: Both Directions [%]	Packet Loss: Uplink [%]	Packet Loss: Downlink [%]	RTT Median 20 bytes [s]	Throughput Median 20 bytes [bps]	Energy Consumption default Median all packet sizes [J]	Energy Consumption RAI-400 Median 20 bytes [J]
Op1 / BC95GJB-02-STD	Bad	0.621	0.573	0.048	3.333	386.708	2.801	0.251
Op1 / BC95GJB-02-STD	Good	0.468	0.401	0.067	2.894	1574.426	2.388	0.162
Op1 / SARA-N211	Bad	0.427	0.047	0.379	4.045	385.890	4.114	0.447
Op1 / SARA-N211	Good	0.124	0.093	0.031	3.102	1235.531	4.174	0.326
Op2 / BC95GJB-02-STD	Bad	1.130	0.963	0.167	5.391	203.304	1.048	1.331
Op2 / BC95GJB-02-STD	Good	0.947	0.726	0.221	2.753	400.751	0.819	0.121
Op2 / SARA-N211	Bad	0.855	0.617	0.237	6.220	203.110	1.396	0.427
Op2 / SARA-N211	Good	0.772	0.386	0.386	2.478	399.750	1.270	0.311

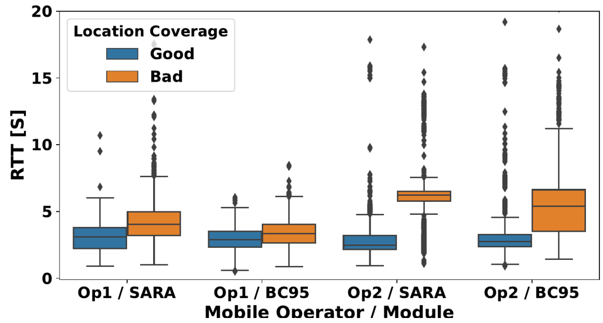


Figure 7.16.: RTT per location coverage when sending a 20-byte packet. All packet sizes.

This might indicate that the losses are not happening in the Radio Access Network. If guaranteed delivery is important, the use of a higher layer protocol such as MQTT or CoAP is needed.

RTT We measure RTT through the device logs. Fig. 7.16 and 7.17 present how signal quality affects RTT. Under good conditions, despite the differences in energy consumption presented in Sec. 7.5, round trip delays show small variability among the combinations examined. On the other hand, higher ECL values increase the duration of both transmission and reception, due to the big number of repetitions, consequently increasing RTT. In Fig. 7.16, the higher values of delay under bad signal conditions for Op 2 compared to Op 1 are attributed to the much more frequent use of ECL 2. Fig. 7.18 presents the effect of packet size on RTT, where we observe Op 2 having a bit longer delay than Op 1. As expected, longer packets require more transmission and reception time, increasing RTT.

Throughput Fig. 7.19 and 7.20 break down the parameters that affect throughput in the UL. In our calculations, transmission starts from the scheduling time and ends when the packet is transmitted. Due to the signaling overhead, larger packets tend to have a higher average transmission speed, as shown in Fig. 7.19. Both operators are using 15 KHz single-tone mode, which has a theoretical maximum UL peak rate for Cat-NB1 devices of 16.9 Kbps. We observe Op 2 being significantly slower than Op 1, even in good locations, indicating inefficiencies in the signaling procedures and only Op 1 consistently gets measurements close to the theoretical maximum. Signal quality has a great effect in measured speed, with experiments in bad coverage locations resulting in less than half the speed.

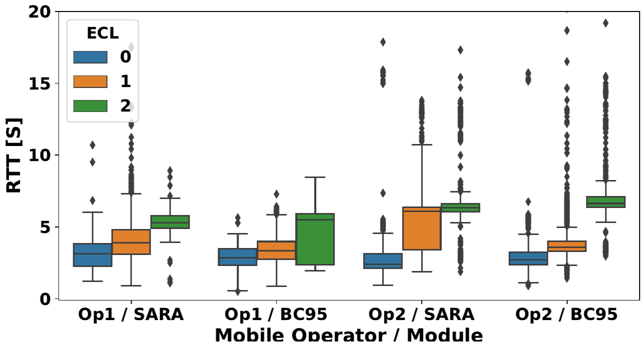


Figure 7.17.: RTT per ECL level when sending a 20-byte packet.

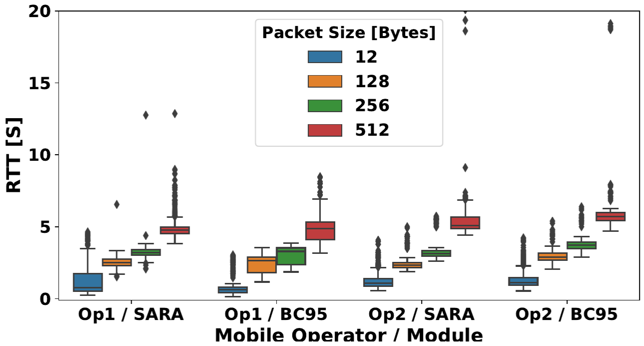


Figure 7.18.: RTT per packet size under good signal.

Takeaways. The NB-IoT networks we measure have higher packet loss rates than ordinary LTE networks. ECL and packet size are the main factors affecting RTT, since they increase the time of all the RAN procedures. Throughput is affected primarily by the operator and the packet size.

The previous sections have presented our comprehensive measurement campaign, over real commercial networks in a variety of conditions, devices and configurations. We have shown that NB-IoT offers greater configuration flexibility compared to LTE, due to the recent NB-IoT specific enhancements and that there are clear differences between operators and modules. Most of the energy per transmission is consumed during the Active Timer phase, thus it is critical for application developers to consider using the appropriate RAI flag. Operator misconfiguration may waste significant amounts of energy and it is hard to detect without low level study of the energy traces. The devices consume disproportionate energy, while meeting their expected network KPIs (i.e., RTT, packet loss), therefore there is no indication of a problematic behavior. For example, Op 1 did not enforce quiet periods during the Active Timer and Op 2 consumed more power under RAI because of a bug that frequently ignored the flag.

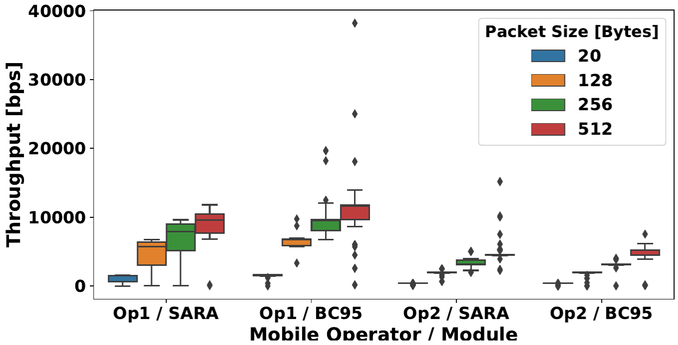


Figure 7.19.: Throughput per packet size in good locations.

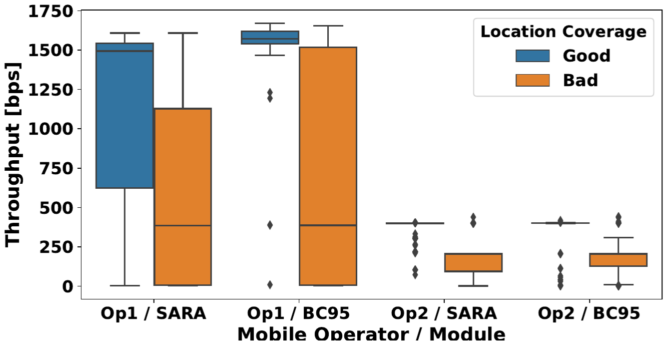


Figure 7.20.: Throughput per location coverage when sending a 20-byte packet. All packet sizes.

Both operators perform reasonably well under poor coverage. The difference in median energy consumption between well and poorly covered locations is attributed to ECL choice. Operators should carefully select their ECL thresholds to avoid wasting energy. Packet loss is very rare, even when operating in extremely challenging conditions, where normal LTE devices would be "out of signal". We could not find a correlation between packet loss and ECL level. Instead, the operator with the more aggressive ECL thresholds exhibits higher packet loss, while also suffering from increased RTT due to the high number of repetitions associated with transmissions with `ECL 2`. Payload size only matters if RAI is in use and the dominant factor dictating energy consumption is the module choice, especially under poor coverage.

Tab. 7.6 attempts to prioritize the impact of several factors affecting energy consumption. This table can be used as a starting point for building a power model estimating an NB-IoT device's lifetime. The choice of the module is a decisive factor in both Idle and Connected states. The operator choice has the least, but still measurable, importance, unless there are bugs present or the devices operate in locations with very poor signal, where fine tuning of ECL

Table 7.6.: Parameter importance hierarchy per state.

State	Parameter Importance
PSM	1) Module
eDRX - sleep	1) Module
eDRX - listening	1) Signal Quality 2) Module 3) Operator
Connected	1) RAI flag 2) Signal Quality 3) Module 4) Operator (assuming no bugs) 5) Packet Size (Only with RAI flag)

thresholds is important. Based on the above, applications' developers should carefully select their platform.

In brief, we conclude that NB-IoT deployments need careful parameter tuning. The use of Active Timer should be thought through. A slight misconfiguration can result in excessive energy consumption and finally, the measured NB-IoT deployments seem to fare well under poor coverage conditions except for extreme cases.

7.8 Discussion

7.8.1 Device lifetime

In order to have an estimation of the device lifetime for a given battery capacity, network configuration, and transmission frequency, we need to quantify the energy consumption for the three distinct states of an NB-IoT device lifecycle: 1) PSM, 2) eDRX and 3) Connected state. Thus, the expected lifetime $T_{lifetime}$ of the device, assuming no battery degradation and a fixed transmission interval T_{ti} is:

$$T_{lifetime} = \frac{E_{batteryCapacity}}{(E_{Con} + E_{eDRX} + E_{PSM})} * T_{ti}, \tag{7.2}$$

We sketch a toy example, to explore how different configurations and the choice of the UE impact device lifetime. In this example, a UE under good coverage sends a 20-byte UDP packet to an echo server, which responds to it. This activity is repeated in 3 different intervals, with each interval being representative of an NB-IoT use case. The intervals are: i) 1h (e.g., environment monitoring), ii) 4h (e.g., irrigation) and iii) 24h (e.g., vehicle automation) [192]. The UE spends the rest of the day in Idle state. We explore two different configurations: default timers (i.e., the RAI flag is not set) and RAI-400. We make two simplifications. First, we ignore the energy consumed

Table 7.7.: Expected battery lifetime, in years, grouped by transmission intervals, for good signal conditions. We assume a battery with the 3GPP target size of 5 Wh (18000 Joule) and no battery degradation over time. The transmission is a single 20-byte UDP packet, which is echoed back.

Module	Op.	Default timers			RAI-400		
		1h	4h	24h	1h	4h	24h
Quectel-BC95	Op 1	0.8	3.2	9.9	6.1	25.5	45.4
	Op 2	2.4	8.5	13.0	6.4	30.1	47.6
SARA-N211	Op 1	0.5	1.9	6.0	6.0	18.5	44.1
	Op 2	1.6	5.9	5.7	5.9	17.7	43.3

in the eDRX mode. This is a reasonable assumption for a big number of NB-IoT use cases, where a sensor reports data (Uplink), but is not needed to be contacted (Downlink), thus eDRX can be disabled. Second, we ignore the energy consumption associated with periodic TAU updates. Given the frequency of Uplink messages, TAU is not needed in this scenario. According to 3GPP's objectives for NB-IoT, devices should be able to achieve "up to ten years battery life with battery capacity of 5 Wh (Watt-hours), even in locations with adverse coverage conditions" [16]. Thus, we assume a 5 Wh (18000 Joule) battery, which does not suffer from degradation. To estimate E_{Con}, we use the median values from tables 7.1 and 7.3, which is a good approximation given the compactness of the respective distributions. The energy consumption during PSM is calculated by multiplying the median power consumption values from Sec. 7.6.1 with the duration spent in Idle state.

Tab. 7.7 shows the expected battery lifetime in years for different operator, module and configuration combinations. Misconfiguring energy saving procedures, for example the lack of cDRX in early measurements of Op 1, drastically reduces the expected lifetime. Using RAI leads to significant energy saving extending the battery lifetime by several years. Even in this favorable scenario (i.e., good signal, small packets), the use of RAI is necessary to achieve the 10-year lifetime goal of 3GPP. Also, the differences between modules translates to months of difference in battery lifetime, even in the 1h interval scenario. Note that most of the energy consumption takes place while the UE is in deep sleep, because it spends the bulk of its lifetime in that state. We have also evaluated other experiment conditions to gauge their impact on battery lifetime. Taking the best case in Tab. 7.7 above, that is Quectel-BC95 with Op 2, we increase the payload size to 512 bytes, which consumes 0.20 J per message for RAI-400. The expected lifetime per interval becomes: i) 8.6 ii) 23.3 and iii) 44.2 years. If we further assume bad signal conditions, with

RAI-400 and payload 512 bytes, the median consumption becomes 3.09 J, thus making the expected lifetime i) 0.7, ii) 2.5 and iii) 12.3 years.

The above logic may be applied to a plethora of other NB-IoT use cases. For example, a home alarm is expected to be in Idle state, until a very infrequent trigger (e.g., once every three months), that is safe to ignore in our calculations. In this case, we would have to take into account the periodicity of TAU updates, (T_{TAU}) which energy-wise can be assumed to be equal to a 20 byte packet transmission with RAI-200, as these are the only other events that provoke a transition to the Connected state. The expected lifetime would be given by Eq. (7.2) with $T_{ti} = T_{TAU}$. On the other hand, a door lock use case would be dominated by random triggers. We can get an estimate for the T_{ti} from the expected number of triggers per week. Eq. (7.2) is not affected by the distribution of T_{ti}, so using the average value is sufficient. In both use cases, we would also be interested in one-way delay and packet loss. As we have shown, RTT is rarely above 10 seconds, even in extreme cases, and we may ensure delivery through an application layer protocol. In the rare Downlink-heavy use cases, such as unlocking city bikes, the dominant factor in Eq. (7.2) would be E_{eDRX}, because continuous reachability is needed and thus, the heavy use of eDRX is a necessity. eDRX is given by Eq. (7.1) and T_{ti} could again be an average value based on the expected number of triggers per week.

Based on the above, the use of default timers should be carefully thought through, employing the RAI flag whenever possible. Any use case that does not involve multiple communication from the server side, following the initiation of an UL transmission, should do away with it. It is the default configuration, however, which means that most users might end up using it unknowingly. It is not reasonable to assume that application developers will be well versed in all aspects of energy saving in NB-IoT. However, they need to familiarize themselves with the terms in Eq. (7.2). Furthermore, the UE vendors need to publish power ratings for their devices when in deep sleep. Operators need to publish details on how they implement energy saving and to certify common UEs chipsets. The availability of such information will make it easier for use case owners to come up with reasonable battery lifetime estimates.

7.8.2 Feedback to operators

Our findings were reported back to both operators, which took appropriate actions to correct the deficiencies we observed. Op 2 had a bug with RAI-200, that was fixed after reporting it during our main measurement campaign, achieving 80% better energy efficiency (see Sec. 7.5.2). During the measurement period, Op 1 did not support some NB-IoT power saving mechanisms, resulting in higher energy consumption than Op 2. During the Inactivity timer period, instead of performing cDRX, the modules were constantly on a high-power paging state. We have informed Op 1 of this anomaly, they later informed us that it has now been fixed. We have then collected a complementary dataset in the first half of July 2019, where we observe clear improvements. The new measurements confirm that Op 1 now implements cDRX. In these experiments we use SARA-N2 to send 20 bytes, from a location with good coverage, using the default timers. The median energy consumption of the Connected state is now 0.912 Joule, having improved by 77%. Actually, the energy consumption has become lower than Op 2's, because the cDRX mode of Op 1 has fewer and more spaced out listening occasions. Op 2 supported these power saving features from the beginning of our measurements, thus we do not observe any differences at the newer dataset. The immediate impact of our study highlights the need for similar studies as NB-IoT is being rolled out and soon 5G will be.

7.8.3 Comparison with other IoT communication technologies

As we discussed in Sec. 7.5.1, NB-IoT's energy consumption is highly variable, especially under poor conditions. Other competing IoT communication technologies such as LoRaWAN may have a similar median consumption, but significantly smaller variance [183], making them more attractive to application developers, due to their more predictable battery life expectancy. The main advantage of NB-IoT over these technologies is the use of licensed spectrum. This will be increasingly important as the number of IoT devices increases, and thus the possibility of interference becomes higher. Further, the use of unlicensed spectrum is typically associated with higher delay and does not offer delivery guarantees. In contrast, the HARQ mechanism of NB-IoT offers increased robustness and the centralized scheduling at the base station minimizes delay and interference.

7.9 Related works

Experimental analysis close to our work are [183, 184, 185], which use the same devices but over different networks and in a smaller variety of scenarios. The authors of [183] performed measurements over a single commercial network in Spain with the same devices. Similarly to us, they observe that Quectel has better energy consumption than Ublox and that packet size does not affect energy consumption. In contrast, they report significant gains by using the RAI flag only for Ublox, and considerably less energy needed to listen for PO during eDRX for both modules. We expand on their work, by comparing the performance of two operators and attempt to identify the parameters mostly affecting lifetime. In [184], the authors use the same devices, but with older firmware that supported only release 13 features. The experiments reported are an integration study for the network of Telekom Malaysia, where they also study energy efficiency. They reach the same conclusion as us: in order to achieve the promised lifetime a careful set up of the NB-IoT device's firmware is necessary. In contrast, we perform our experiments over commercially available networks with the latest firmware of both devices that supports all the current-generation power saving features and under a variety of signal conditions. In [185], the authors perform a small scale experiment to measure the expected lifetime of an NB-IoT device, based on SARA-N2, in the context of aviation use cases. Their testbed connects over one private and two commercial networks and they discover that using PSM in Idle state has the highest impact on achievable battery lifetime. Compared to the three studies above, our experiments are more thorough and use the latest NB-IoT features commercially available. We further attempt to provide explanation of the artifacts we observe, identify key parameters for enhancing lifetime and cooperate with operators to improve their networks.

In the literature, various works have attempted to model NB-IoT power consumption and device lifetime. The authors of [193, 194] presented a Markov chain analysis of the average energy consumed to transfer one uplink report using the Control Plane procedure. In [195] an emulator has been used to create an empirical lifetime model, based on device configuration. The same testbed has been used in [196], where authors measured two early NB-IoT device prototypes and used the results to make lifetime estimation projections. An early simulation study of various IoT technologies' coverage, including NB-IoT, based on a Danish region's topology has been presented in [197]. In [198] Sultania et al. proposed an analytical model to estimate the average energy consumption of an NB-IoT device using the Release 14 power saving enhancements. The work in [199] presented an analytical model to explore the trade-offs between repetitions and the built-in MAC layer retransmission mechanism of LTE, concluding that fewer repetitions with more retransmissions achieves higher successful probability. Recently, [200] presented a theoretical mathematical model to predict performance and propose optimal network configuration in different scenarios.

El Soussi et al. [201] evaluated the overall performance of NB-IoT in the context of a smart city. They proposed a theoretical model for calculating the energy consumption and concluded that a lifetime of 8 years is possible, under poor coverage, while sending one message per day. Finally, the authors of [202] studied the relationship between signal strength and delay through a small number of experiments performed in a laboratory testbed measuring a device prototype based on the SARA-N2 module. These works rely either on emulating parts of the network or simulations. In contrast, we perform large scale experiments in the wild, using two different modules and two operators. Compared to [183, 184, 185], discussed in the previous section, our work is more thorough and uses the latest NB-IoT features available. Finally, to the best of our knowledge, there is no other empirical study on NB-IoT packet loss under real conditions.

7.10 Conclusions

We conducted a comprehensive measurement study of the energy consumption of two popular NB-IoT boards that connect to two commercial deployments in a European country. Our findings indicate that NB-IoT is far from being plug and play and requires careful setting for improved energy efficiency. Since we focus on configuration parameters and their impact on the energy consumption, our recommendations can be generalized to any NB-IoT deployment. We observe that the main factors determining energy consumption, and thus battery life, are: 1) module; 2) operator; 3) signal quality; 4) use of energy saving enhancements such as RAI and eDRX and 5) in a limited number of scenarios, packet size. Furthermore, our analysis has helped the measured networks identifying and fixing a couple anomalous configurations, and we could finally track the effectiveness of this adjustments. Finally, we have indicated strategies for improving energy efficiency, pointing out the elements that could bring to energy waste without improving the reliability, such as too aggressive ECL thresholds or not using the RAI flag. We also identified the key parameters needed for estimating the battery lifetime, and which of the metadata reported by the device are more meaningful. Possible future research directions include the energy impact of application layer protocols such as MQTT and CoAP, as well as recommendations for parameter tuning of these protocols.

Conclusions 8

IoT-enabled services and applications are becoming more common and widespread. The analysis of IoT networks performance offers useful insights that according to the scenario considered, can help in the choice of the communication technology and its best configuration, while respecting the requirements dictated by the specific use case.

In this thesis, we studied two LPWAN technologies, namely LoRaWAN and NB-IoT. For LoRaWAN we considered the network performance, whether the technology could fit also non-traditional applications, and whether it could be supported by battery-less devices. Additionally, we evaluated the energy consumption of NB-IoT devices.

In the first part of the thesis (Ch. 4), we tested the performance of LoRaWAN networks in traditional IoT use cases (i.e., monitoring applications in open air and urban scenarios). One of the main contribution has been the formulation of a mathematical model, which captures features from the protocol (unconfirmed/confirmed traffic, presence of re-transmissions), but also specificities of the chipsets (limited number of GW's demodulators) and local restrictions (DC regulations). Additionally, the validation of the model with the ns-3 simulator outcomes made it possible to identify the assumptions that impacted the most on the model performance. Finally, the model provides a computationally efficient tool to investigate the parameters space, also to optimize some combination of metrics of interest. The results obtained from the model, together with those provided by ns-3 simulations, showcased the impact of the communication settings on different metrics, from the packet delivery rate to the fairness, from the delay to the number of reception windows that are opened. In particular, we found that the two elements that played a major role in determining the system performance are the use of confirmed traffic and the SF used in RX2. Furthermore, with this analysis, we could link optimal network configuration to quantities such as node density and generated traffic, obtaining the range of performance that can be expected in a real network.

Further simulation studies, reported in Ch. 5, showed that, with a proper configuration of the aforementioned parameters and a denser network infrastructure, LoRaWAN can also support applications that are out of the classical use cases it was designed for. It should be noted that, however, LoRaWAN struggles to achieve the delay and reliability performance required by the IIoT use cases, and that when considering a drone patrolling application, acceptable performance of the drone tracking could only be achieved with a careful choice of the communication settings. Nevertheless, these applications can still take advantage of the long-range and low-power communication features of LoRaWAN.

In Ch. 6, we described the ns-3 implementation of a battery-less IoT node with harvesting capabilities, facing one of the hot topics in the current society: the exploitation of renewable energy sources for power supply. Therefore, in the perspective of enabling Green IoT solutions, the developed code provides a tool to evaluate whether such approaches are, or not, viable. Although the ns-3 code can be applied to different low power technologies, such as WiFi 802.11ah HaLow, in this thesis, we validated the implementation by considering LoRaWAN devices. The results presented in the chapter unveiled the potential of this approach, showing how the environment (i.e., harvesting capabilities) and the device hardware (i.e., capacitor size) affect the feasibility of such a solution. Furthermore, preliminary algorithms for smart energy-aware packet scheduling were proposed. The outcomes of preliminary experiments run in real testbeds have also been presented, prooving the feasibility of a battery-less solution for LoRaWAN devices. Finally, in Ch. 7 we considered energy consumption of NB-IoT devices, through extensive measurement campaigns. From this analysis, it was apparent that these devices can not be employed in a *plug-and-play* fashion, and many aspects, such as the position of the device with respect to the infrastructure, have a strong impact on the performance. Furthermore, it should be noted that, differently from LoRaWAN, NB-IoT and other cellular technologies can not be deployed in private networks, but only with the support of a telecommunication operator. Therefore, the control of the whole network and of the communication parameters (such as the timers enabling power saving features) do not depend on the user, who then may suffer from unwise network settings that could negatively affect the device's lifetime, with an impact on the maintenance costs (e.g., battery replacement).

From these considerations, it is clear that preliminary analyses are of uttermost importance to understand the functioning of a network and whether and how specific applications can be supported. Also, the adoption of Green IoT would make it possible to move towards more sustainable deployments, also limiting the impact on the environment.

8.1 Directions for future work

The analysis presented in this thesis aimed to include most of the tools that are commonly employed to evaluate communication technologies, i.e., mathematical formulation, simulative tools, and testbed experiments. Furthermore, several aspects have been analyzed, from traditional network metrics to performance in more specific scenarios, from Green IoT approaches to energy consumption evaluations. Although the presentation tried to cover all these aspects thoroughly, every single contribution can be further extended, providing more detailed insights.

First, the mathematical and simulative analysis presented in Ch. 4 can be enhanced to also consider multiple-GWs scenarios, and leveraged as tools to find optimal configurations. Secondly, the simulative framework employed throughout this thesis could be extended implementing also Class-B EDs, which could represent a good compromise in IIoT networks. Additionally, outdoor scenarios could be further investigated, both for industrial applications (e.g., an industrial harbor), or for tracking applications: in both these use cases, the communication technology would be further pushed to its limits, also applying features that are currently not part of the LoRaWAN standard to increase the communication reliability and minimizing the latency metrics. Finally, energy consumption turned out to be a critical element also in those devices conceived to minimize it, as LPWAN devices should be. Therefore, empirical studies that consider this aspect in real scenarios are always a useful method to evaluate energy performance previous actual installation and can help in identifying possible bugs in new network deployments. However, the increasing number of connected devices also motivates the migration towards Green IoT. However, this solution is still in its first phases and different directions should be explored. Firstly, the feasibility of this approach should be assessed, since the strict requirements imposed by the limited and inconstant energy may prevent the implementation of some types of services or the usage of some IoT technologies that have more costly operations in terms of energy. Secondly, the available technology parameters and communication protocol should be optimized to proficiently work in such conditions. Finally, we can observe that the usage of renewable sources is certainly an advantage for the environment and complies with a responsible vision of the long-term impact of the IoT technologies. However, it must be noted that there is no one-fits-all solution, and each deployment should be carefully configured also according to the different environmental conditions that characterize each specific scenario.

Appendix A

A

In this appendix, we prove the convergence of the model presented in 4.2, which aims at estimating the LoRaWAN network performance in a single-gateway scenario. In Sec. A.1, we provide an analytical proof of the existence of a fixed point solution for such a system. Then, in Sec. A.2, we report experimental results, showing that the system of the two inter-dependent equations provided by the model can be solved through fixed-point iterations, and that a limited number of iterations is enough to reach convergence.

A.1 Proof of existence of fixed point solution

For ease of writing, let $\mathbf{x} = [x_1, \ldots, x_n]$ be the vector of unknowns to be determined by solving the set of fixed point equations $f_i(\mathbf{x}) = x_i$. From the derivation of the model presented in the Sec. 4.2, it is apparent that the functions $f_i()$ result from the combination of a number of continuous (and differentiable) functions, and are themselves continuous (and differentiable) in all the unknowns x_i in the compact interval $(0, 1)$.

We can apply the following reasoning iteratively.

Let's start from $i = 1$. By fixing all the parameters other than x_1 we can define the functions $F_1(x_1; p_1) = f_1(x_1, \ldots, x_i, \ldots x_n)$ where $p_1 = [x_2, \ldots, x_n]$ is the vector of parameters $\mathbf{x}$ without the first term.

Now, analyzing the functions that yield to the expression of $F_1(x_1; p_1)$, we can see that $F_1(x_1; p_1)$ is greater than zero when x_1 tends to zero, and lower than one when x_1 tends to one. Therefore, there must exist a point $x_1^* \in (0, 1)$ such that $F_1(x_1^*; p_1) = x_1^*$. Clearly, this point in general depends on the parameter vector p_1. We hence denote by $F_1^*(p_1)$ the fixed point solution x_1^* of $F_1(x_1; p_1)$ for a certain p_1. We will later prove that this function is continuous in p_1.

We can now define the function

$$f_2^{\circ}(x_2; x_3, \ldots, x_n) = f_2(F_1^*(x_2, x_3, \ldots, x_n), x_2, x_3, \ldots, x_n).$$

Since f_2 is continuous in all the parameters, and F_1^* is continuous in $[x_2, x_3, \ldots, x_n]$, then f_2° is also continuous in $[x_2, x_3, \ldots, x_n]$.

Furthermore, the function turns out to be greater than zero when x_2 tends to zero and lower than one when x_2 tends to one. Therefore, we can repeat the

reasoning iteratively, until we reach the function $f_n^\circ(x_n)$ that hence admits a fixed point x_n^*.
Hence, we get that the fixed point solution of the original problem is given by the values $\{F_i^*(x_{i+1}^*, \dots, x_n^*)\}$, for $i = 1, 2, \dots, n$.

A.1.1 Proof of continuity of F_1^*

Theorem

The function $F_i^*(x_{i+1}, \dots, x_n)$ is continuous in p_i^*, where $p_i^* = (x_{i+1}, \dots, x_n)$.

Proof

We need to prove that $\forall p_i^*$, $\forall \epsilon > 0$, $\exists \delta > 0$, such that

$$|F_i^*(p_i^*) - F_i^*(\widetilde{p}_i)| < \epsilon, \forall \widetilde{p}_i : |p_i^* - \widetilde{p}_i| < \delta.$$

Recalling that $F_i^*(p_i^*) = x_i^*$ such that $f^\circ(x_i^*, p_i^*) = x_i^*$, then we can write

$$\Rightarrow |F_i^*(p_i^*) - F_i^*(\widetilde{p}_i)| = |x_i^* - \widetilde{x}_i|$$

where $\widetilde{x}_i$ is the fixed point of $f_i(x_i, \widetilde{p}_i)$, i.e., $f^\circ(\widetilde{x}_i, \widetilde{p}_i) = \widetilde{x}_i$.
We hence need to prove that $\widetilde{x}_i \in (x_i^* - \epsilon, x_i^* + \epsilon)$, i.e., that

$$g_{\widetilde{p}_i}(x_i) = f_i^\circ(x_i, \widetilde{p}_i) - x_i = 0$$

for some point $\widetilde{x}_i \in (x_i^* - \epsilon, x_i^* + \epsilon) \triangleq B(x_i^*, \epsilon)$.
Let's call $B(p_i^*, \delta)$ the ball of radius δ centered in p_i^*, i.e, $B(p_i^*, \delta) \triangleq (p_i^* - \delta, p_i^* + \delta)$.
Assume, by contradiction, that $\forall \delta > 0$, $\exists \hat{p}_i \in B(p_i^*, \delta)$ such that $g_{\widetilde{p}_i}(x_i) > 0$, $\forall x_i \in B(x_i^*, \epsilon)$. (The case $g_{\widetilde{p}_i}(x_i) < 0$ is similar).
Let

$$g_{min} = \min_{x_i \in B(x_i^*, \epsilon)} g_{\widetilde{p}_i}(x_i).$$

Therefore, $g_{\widetilde{p}_i}(x_i) \geq g_{min} \, \forall x_i \in B(x_i^*, \epsilon)$.
In particular,

$$g_{\widetilde{p}_i}(x_i^*) \geq g_{min} \qquad \Rightarrow \qquad g_{min} \leq f_i^\circ(x_i^*, \widetilde{p}_i) - x_i^* = f_i^\circ(x_i^*, \widetilde{p}_i) - f_i^\circ(x_i^*, p_i^*).$$

Recalling that $f^\circ(x_i, p_i)$ is continuous, then by taking $\hat{\epsilon} < g_{min}$, $\exists \hat{\delta}$ such that

$$|f^\circ(x_i^*, \widetilde{p}_i) - x_i^*| < \hat{\epsilon}, \forall \widetilde{p}_i \in B(p_i^*, \hat{\delta}).$$

$$\Rightarrow \exists\,\hat{\delta} > 0, \text{ such that } \forall p \in B(p_i^*, \hat{\delta}),\ g_{\widetilde{p_i}}(x_i^*) < g_{min}.$$

which contradicts the assumption that g_{min} does not admit zeros in the ball of radius ϵ around x_i^*. This concludes the proof.

Q.E.D.

A.2 Experimental results showing system's convergence

In order to provide some estimates on the convergence speed of the proposed model, we ran a series of experiments in which we solved the model through a fixed-point iteration procedure. First, we sampled the parameter space, varying each parameter (such as α, m, p, C, λ), thus obtaining a set of models to solve. Then, we solved each instance of the model starting from a randomly chosen point x in the solution space (i.e., the 12-dimensional space of S_i^{UL} and S_i^{DL}, with $i \in \mathcal{SF}$). We define

$$\mathbf{x}_i = \left[S_i^{UL}, S_i^{DL}\right] = \left[S_7^{UL}, \ldots, S_{12}^{UL}, S_7^{DL}, \ldots, S_{12}^{DL}\right].$$

We stop the fixed-point iteration when $||\mathbf{x}_i - \mathbf{x}_{i+1}||_2 < 10^{-3}$, where $\mathbf{x}_i$ is the solution found at the i-th step in the procedure, and $||\cdot||_2$ is the Euclidean norm.

Fig. A.1 contains the results of such analysis: the box plots show the distribution of the number of iterations necessary to reach convergence. The data is plotted here for various values of α, to highlight how the number of iterations might depend on the value of some parameters, while never exceeding 40 in the worst case, ensuring, thus, a quick convergence for all the explored combinations of the parameters and choice of the initialization point.

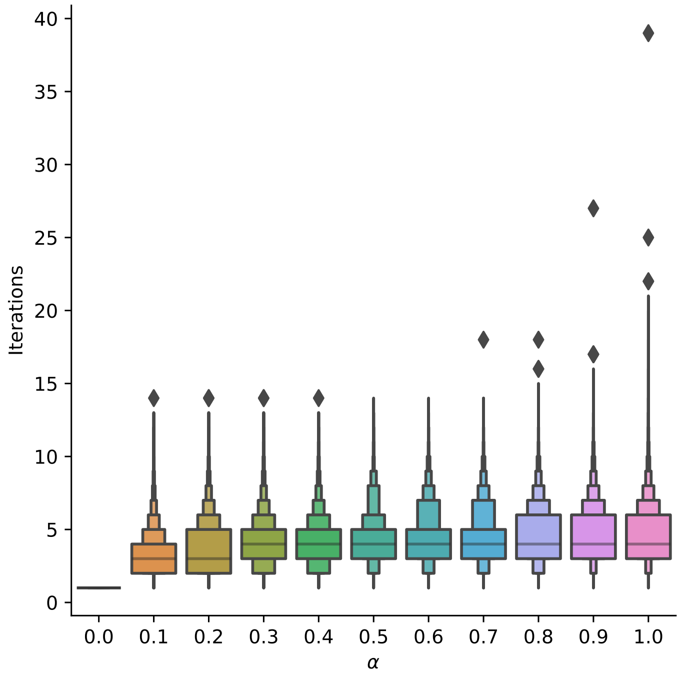


Figure A.1.: Distribution of the number of iterations necessary to reach convergence for a range of values of α.

Appendix B

B

B.1 Special conditions for 3D-CTRA

The 3D-CTRA model equations are well-defined with some conditions on the tilt ψ: it must be different from 0 and from the turn rate ω or its inverse. In those cases, the equations need to be derived separately in order to arrive at a valid mathematical result. First, when $\psi = 0$, the model is equivalent to CTRA+ and the value of $z(t)$ is given by (5.16). Then, when $\omega = \psi$, i.e., the rotations on the two axes have the same period, the values of $x(t)$ and $y(t)$ become:

$$x(t) = x(0) + \left[\frac{v(\tau)\sin(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)} + \frac{a\cos(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)^2} + \frac{2v(\tau)-a\tau}{4}\tau\cos(\theta(\tau)-\phi(\tau)) \right]_0^t ; \tag{B.1}$$

$$y(t) = y(0) + \left[\frac{a\sin(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)^2} - \frac{v(\tau)\cos(\theta(\tau)+\phi(\tau))}{2(\omega+\psi)} + \frac{2v(\tau)-a\tau}{4}\tau\sin(\theta(\tau)-\phi(\tau)) \right]_0^t . \tag{B.2}$$

The case in which $\omega = -\psi$ produces a similar result, with switched terms:

$$x(t) = x(0) + \left[\frac{v(\tau)\sin(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)} + \frac{a\cos(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)^2} + \frac{2v(\tau)-a\tau}{4}\tau\cos(\theta(\tau)+\phi(\tau)) \right]_0^t ; \tag{B.3}$$

$$y(t) = y(0) + \left[\frac{a\sin(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)^2} - \frac{v(\tau)\cos(\theta(\tau)-\phi(\tau))}{2(\omega-\psi)} + \frac{2v(\tau)-a\tau}{4}\tau\sin(\theta(\tau)+\phi(\tau)) \right]_0^t . \tag{B.4}$$

Along with the standard equations, these results complete the model, which is valid for all values of the relevant parameters.

B.2 The Unscented Kalman Filter

Assuming that the process and measurement noises are purely additive[1], the UKF algorithm involves the following operations. Let us call $\hat{\mathbf{x}}(t)$ and $\mathbf{P}_x(t) = E[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T]$ the system state estimation and the related covariance matrix at time t, respectively. First, the state statistics are encoded into $2L+1$ sigma points $\chi_0(t), \chi_1(t), ..., \chi_{2L}(t)$, which are computed according to the so-called Unscented Transformation (UT):

$$\chi_0(t) = \hat{\mathbf{x}}(t); \tag{B.5}$$

$$\chi_i(t) = \hat{\mathbf{x}}(t) + \left(\sqrt{(L+\lambda)\mathbf{P}_x(t)}\right)_i, \qquad i = 1, ..., L; \tag{B.6}$$

$$\chi_i(t) = \hat{\mathbf{x}}(t) + \left(\sqrt{(L+\lambda)\mathbf{P}_x(t)}\right)_{i-L}, \qquad i = L+1, ..., 2L. \tag{B.7}$$

In the above equations, L is the dimension of $\mathbf{x}$, $\left(\sqrt{(L+\lambda)\mathbf{P}_x(t)}\right)_i$ indicates the i-th row of the square root of $(L+\lambda)\mathbf{P}_x(t)$, λ is a scaling parameter, and $\alpha \in \mathbb{R}$ represents the sigma points spread around $\chi_0(t)$. The value of the scaling parameter is computed as $\lambda = \alpha^2(L+k) - L$, where k is another tuneable parameter, which is customarily set to 0. Each of the sigma points is then associated to a set of scalar weights w_i^m and w_i^c, whose values are given by:

$$w_0^m = \lambda/(L+\lambda); \tag{B.8}$$

$$w_0^c = \lambda/(L+\lambda) + 1 - \alpha^2 + \beta; \tag{B.9}$$

$$w_i^m = w_i^c = \lambda/\left(2(L+\lambda)\right), \; i = 1, ..., 2L. \tag{B.10}$$

In (B.9), β is a constant value that has to be tuned according to the system state distribution; if $\mathbf{x}$ is Gaussian, the optimal value for β is 2.

When the UKF algorithm performs the *prediction* step, each sigma point is propagated through the (possibly non-linear) function $f(\cdot)$, given in the first equation of (5.24), thus generating a new set of sigma points: $f\left(\chi_i(t)\right), \; i =$

[1] The UKF formulation for the case in which $\zeta(t)$ and $\eta(t)$ are not additive is more complex and out of the scope of our research; for a deeper insight into the UKF algorithm, see [203].

$0, 1, ..., 2L$. The new sigma points are used to compute the *a priori* state estimation $\hat{\mathbf{x}}(t+T|t)$ and the related covariance matrix $\mathbf{P}_x(t+T|t)$:

$$\hat{\mathbf{x}}(t+T|t) = \sum_{i=0}^{2L} w_i^m f\left(\chi_i(t)\right); \tag{B.11}$$

$$\mathbf{P}_x(t+T|t) = R + \sum_{i=0}^{2L} w_i^c (f\left(\chi_i(t)\right) - \hat{\mathbf{x}}(t+T|t))(f\left(\chi_i(t)\right) - \hat{\mathbf{x}}(t+T|t))^T; \tag{B.12}$$

where R is the process noise covariance given in (5.24).
At the beginning of the *update* step, the original sigma points are propagated through the measurement function $h(\cdot)$, given in the second equation of (5.24). Hence, the new points $h\left(\chi_i(t)\right)$, $i = 0, 1, ..., 2L$, are used to predict the future observation $\hat{\mathbf{o}}(t+T|t)$ and the related covariance matrix $\mathbf{P}_o(t+T|t)$:

$$\hat{\mathbf{o}}(t+T|t) = \sum_{i=0}^{2L} w_i^m h\left(\chi_i(t)\right); \tag{B.13}$$

$$\mathbf{P}_o(t+T|t) = Q + \sum_{i=0}^{2L} w_i^c (h\left(\chi_i(t)\right) - \hat{\mathbf{o}}(t+T|t))(h\left(\chi_i(t)\right) - \hat{\mathbf{o}}(t+T|t))^T; \tag{B.14}$$

where Q is the measurement noise covariance. Hence, the optimal *Kalman gain* is computed as $K = \mathbf{P}_{xo}\left(\mathbf{P}_{xo}\right)^{-1}$, where $(\cdot)^{-1}$ indicates the inversion operation and $\mathbf{P}_{xo}$ is the residual covariance matrix:

$$\mathbf{P}_{xo} = \sum_{i=0}^{2L} w_i^c (\chi_i(t+T|t) - \hat{\mathbf{x}}(t+T|t))(\xi_i(t+T|t) - \hat{\mathbf{o}}(t+T|t))^T. \tag{B.15}$$

Finally, given the new observation $\mathbf{o}(t+T)$, the *a posteriori* state estimation $\mathbf{x}(t+T)$ and the related covariance matrix $\mathbf{P}_x(t+T)$ are obtained as

$$\mathbf{x}(t+T) = \hat{\mathbf{x}}(t+T|t) + K\left(\mathbf{o}(t+T) - \hat{\mathbf{o}}(t+T|t)\right); \tag{B.16}$$

$$\mathbf{P}_x(t+T) = \mathbf{P}_x(t+T|t) - K\mathbf{P}_o(t+T|t)K^T. \tag{B.17}$$

Appendix C

C

In this appendix, we detail how current consumption traces and log traces for a NB-IoT device have been syncronized to obtain an automatic matching of the current consumption values and phase reported by the log messages. Furthermore, we describe how current consumption traces have been automatically processed to extract the current consumption in each phase of the device. Finally, we discuss how to compute the mapping between SNR and RSRP.

C.1 Data pre-processing

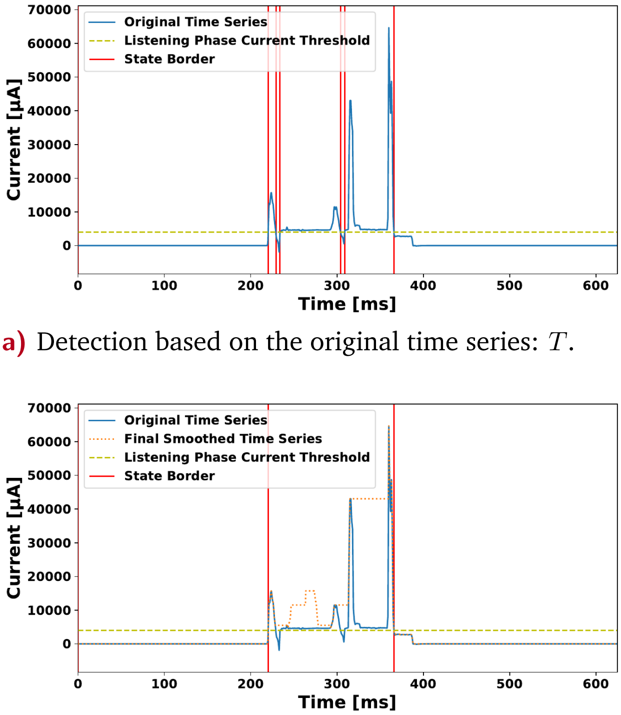
In this section we present how we synchronize the logs of the Otii Arc power measurement device with the logs of the UEs. The UEs report network metadata such as RRC connection state and DRX. These must be synchronized with the Otii power measurements to avoid misattributing energy consumption, connection state-wise. A listening phase in Idle state typically lasts less than 300 ms and a cDRX one less than 30 ms, thus the synchronization ideally should have an error of a few ms at most. Unfortunately, the UE and the power meter clocks could not be synchronized to the required accuracy. Instead, we resort to time series analysis to dissect the current consumption time series into phases. We leverage the fact that the power consumed in different phases is markedly different, as well as characterized by different patterns (see Sec. 2.2). For example, we are able to isolate the DRX listening phase or the synchronization procedure.

Fig. C.1 presents an example of how our phase detection algorithms operate when detecting an eDRX listening phase. Other events are detected in a similar fashion. Depending on which phase we are trying to detect, we can set a current threshold, above which we assume the device is within that phase. Thus, the edges of each phase are the points where the time series cross this threshold. The threshold is determined by the value of the 95th percentile of the current of a typical phase. As can be seen in Fig. C.1a, the original power monitor time series ($T = \{T_1, \ldots, T_n\}$, where n is the number of observations) is very volatile, crossing the threshold multiple times within a single phase, which makes phase detection hard. Based on T, we create two smoothed time series, each aimed to properly detect one edge of the target phase. The window size of the smoothing functions is determined empirically per measurement at a value that removes fluctuations, while avoiding overlap with neighboring listening phases. These are combined to create a `Final Smoothed Time Series`: $FSTS$, where both edges of every phase are well defined and we use this as a guide to time stamp the start and end of each phase. Then, we use the original time series T to get the energy consumption and the rest of the target metrics of the now well defined phases.

For example, to identify the eDRX listening phase, of Fig. C.1 we have to calculate: A) The moving max of all the values ahead of the current one in the window, aimed to detect the phase end. Each element of this time series: `Moving Max Forward`: MMF, is given by: $MMF_i = max\{T_i, \dots, T_{i+W}\}$, where W is the window size of the function. B) The moving max of all the values before the current one in the window `Moving Max Backward`: MMB, aimed to detect the phase start. The elements of MMB are given by: $MMB_i = max\{T_{i-W}, \dots, T_i\}$. C) Calculate the $FSTS$, by overlapping MMB_i and MMF_i. We do so by taking their minimum: $FSTS_i = min(MMB_i, MMF_i)$. $FSTS$ has the property of increasing as soon as the current increases, while not being sensitive to current fluctuations, thus creating a tight mask around the phase we want to detect. The resulting $FSTS$ and the phase borders it generates for the eDRX listening phase detection example are seen in Fig. C.1b. Finally, we apply some filtering on the detected events, to remove artifacts, such as current spikes when we poll the modules for metadata.

Detection of other events, with more distinct patterns, such as the transition between Connected and Idle state, can be simpler. For example, to identify Connected and Idle states, a single smoothed time series of a moving median around the central value of the window is enough to properly identify both the beginning and the ending of a state. This is possible due to the bigger difference in the power consumption between the two states and the bigger duration and periodicity of these events.

The smoothing functions used depend on the event. The parameters depend on the behaviour of the current time series, which is affected by experiment conditions and settings, thus might need adjustment per measurement. An added benefit of this method is that it is very computationally efficient, since it utilizes time series libraries instead of loops, providing fast results in processing the very big files provided by the power monitor tool.



(a) Detection based on the original time series: T.

(b) Detection based on the final smoothed time series: $FSTS$.

Figure C.1.: Results of the phase detection algorithms when trying to detect an eDRX listening phase. The red vertical lines signify the borders between different phases.

C.2 SNR to RSRP mapping

NB-IoT devices calculate SINR over the whole 180 KHz channel: $SINR = \frac{12*RSRP}{I_{tot}+N_{tot}}$. In contrast, RSRP is calculated over a single Resource Element (RE), which has 15KHz bandwidth and is assumed to be free of noise and interference. Thus, to map SINR and RSRP we need to modify the above equation to take into account only 15KHz: $SINR_{15KHz} = \frac{RSRP}{I_{15KHz}+N_{15KHz}}$. In our experiments, due to the limited adoption of NB-IoT and the nature of GuardBand deployment, we can safely assume that interference is minimal, especially in the poor coverage scenarios, thus: $SNR_{15KHz} = \frac{RSRP}{N_{15KHz}}$ The N_{15KHz} depends on the thermal noise density and the receiver noise figure, which have typical values of $N_{thermal} = -1740cBm/Hz$ and $NF_{receiver} = 70cB$, respectively. Thus the thermal component of the noise is:

$$\begin{aligned} N_{thermal\_15KHz} &= -1740cBm/Hz + 100log(15000Hz) \\ &\approx -1322cBm. \end{aligned} \tag{C.1}$$

N_{15KHz} then becomes:

$$\begin{aligned} N_{15KHz} &= N_{thermal\_15KHz} + NF_{receiver} \\ &= -1322.39cBm + 70cB = -1252cBm. \end{aligned} \tag{C.2}$$

Finally the ideal mapping of SNR values to RSRP under our assumptions in logarithmic scale is:

$$\begin{aligned} SNR_{15KHz} &= \frac{RSRP}{N_{15KHz}} \Rightarrow \\ SNR_{cB} &= RSRP_{cBm} + 1252. \end{aligned} \tag{C.3}$$

List of Publications

Journals

[J1] G. Cisotto, M. Capuzzo, A. V. Guglielmi, and A. Zanella, “Feature Stability and Setup Minimization for EEG-EMG-enabled monitorning system”, submitted to EURASIP Journal on Advances in Signal Processing, 2021.

[J2] D. Magrin, M. Capuzzo, A. Zanella, and M. Zorzi, “A Configurable Mathematical Model for Single-Gateway LoRaWAN Performance Analysis,” submitted to IEEE Transactions on Wireless Communications, 2021.

[J3] F. Mason, M. Capuzzo, D. Magrin, F. Chiariotti, A. Zanella, and M. Zorzi, “Tracking of UAV swarms via 3D mobility models and LoRaWAN communications,” submitted to IEEE Transactions on Wireless Communications, 2021.

[J4] F. Michelinakis, A. S. Al-Selwi, M. Capuzzo, A. Zanella, K. Mahmood and A. Elmokashfi, “Dissecting Energy Consumption of NB-IoT Devices Empirically”, in IEEE Internet of Things Journal, vol. 8, no. 2, pp. 1224-1242, 15 Jan.15, 2021.

[J5] D. Magrin, M. Capuzzo, and A. Zanella, “A Thorough Study of LoRaWAN Performance Under Different Parameter Settings,” IEEE Internet of Things Journal, vol. 7, no. 1, pp. 116–127, Jan. 2020.

[J6] D. Magrin, M. Capuzzo, A. Zanella, L. Vangelista, and M. Zorzi, “Performance Analysis of LoRaWAN in Industrial Scenarios,” in IEEE Transactions on Industrial Informatics, 2020.

Conference proceedings

[C1] M. Capuzzo, C. Delgado, A. Kumar Sultania, J. Famaey, and A. Zanella, “Enabling Green IoT: Energy-Aware Communication Protocols for Battery-less LoRaWAN Devices”, in International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2021.

[C2] M. Capuzzo, “LoRaWAN networks evaluation through extensive ns-3 simulations”, in IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Ph.D. Forum, 2021.

[C3] M. Capuzzo, C. Delgado, J. Famaey, and A. Zanella, “An ns-3 implementation of a battery-less node for energy-harvesting Internet of Things”, in Workshop on ns-3 (WNS3), 2021.

[C4] G. Cisotto, M. Capuzzo, A. V. Guglielmi, and A. Zanella, “Feature selection for gesture recognition in Internet-of-Things for healthcare”, in IEEE International Conference on Communications (ICC), 2020.

[C5] F. Mason, F. Chiariotti, M. Capuzzo, D. Magrin, A. Zanella, and M. Zorzi, “Combining LoRaWAN and a New 3D Motion Model for Remote UAV Tracking,” in IEEE Workshop on Wireless Sensor, Robot and UAV Networks (INFOCOM WISARN), 2020.

[C6] M. Capuzzo, D. Magrin, and A. Zanella, “Mathematical Modeling of LoRaWAN Performance with Bi-Directional Traffic,” in IEEE Global Communications Conference (GLOBECOM), 2018.

[C7] M. Capuzzo, D. Magrin and A. Zanella, “Confirmed Traffic in LoRaWAN: Pitfalls and Countermeasures,” in Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2018.

Acknowledgements

This thesis represents the finish line of a three-years path, but also a meaningful occasion to acknowledge all the signs and preliminary directions that drove me towards that trail. Therefore, I take the opportunity to express my sincere gratitude to the following people and organizations, who supported me during the writing of the thesis and the research behind it.

My supervisor, Prof. Andrea Zanella, for his enthusiasm, trust, useful feedbacks and guidance, and for mentoring and supporting me throughout my doctoral program.

The whole SIGNET research group, students and professors for the time spent together, Giulia and Anna Valeria for the steadfast collaboration in developing the *FeSC* pipeline which, unfortunately, could not find place in this thesis.

The IDLab team of the University of Antwerp, especially Prof. Jeroen Famaey and Carmen Delgado, with whom I collaborated to explore Green IoT applied to LoRaWAN.

Also, a special thanks to every member of Office 6.03 (and extensions): although the office was not crowded, I really felt part of the group, and always found an hand when looking for some help.

My collegues at DEI, University of Padova, for mutually backing ourselves up thoroughout this adventure, and having been as a second family: the group of the "olders", injecting doses of craziness in the new drafts: Chiara, Silvia, Giulia, FedeC, Michele, Daniel, Gianluca, Britz, Elvina, Marco. The group of the "younger" with whom we could not rellay spread all this craziness because of the weird historical period we are facing: Matteo, Paolo, FedeM, Marco, Giovanni, Jacopo, Silvia, Anay, Salman, Adriano, Elena, Laura, Francesco & more, and, most of all, my "pairs": Seba, Umbi, Leo, Cheese, Mattia, Francesca, Alberto, Tommaso. Last but not the least, Davide, for having shared joys and pains of these last three (and a bit more) years.

You do not go far if you do not have solid roots. I thank mum Lucia and dad Paolo for being always next to me, my sister Giada, for providing me with hundreds of examples of how to turn normality into occasions of good humor, my grandmum Margherita and my whole family and relatives.

Bibliography

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014, ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2306328.

[2] F. Chiariotti, M. Condoluci, T. Mahmoodi, and A. Zanella, "Symbiocity: Smart cities for smarter networks," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 1, e3206, 2018.

[3] N. Dlodlo and J. Kalezhi, "The internet of things in agriculture for sustainable rural development," in *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, 2015, pp. 13–18. DOI: 10.1109/ETNCC.2015.7184801.

[4] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.

[5] Y. Yuehong, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: An overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, 2016.

[6] M. Polese, M. Centenaro, A. Zanella, and M. Zorzi, "M2M massive access in LTE: RACH performance evaluation in a Smart City scenario," in *IEEE International Conference on Communications (ICC)*, 2016.

[7] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: A survey," *arXiv preprint arXiv:1606.07360*, 2016, version 1.

[8] S. M. Darroudi and C. Gomez, "Bluetooth Low Energy Mesh Networks: A Survey," *Sensors*, vol. 17, no. 7, p. 1467, 2017.

[9] B. Mihajlov and M. Bogdanoski, "Overview and analysis of the performances of ZigBee-based wireless sensor networks," *International Journal of Computer Applications*, vol. 29, no. 12, pp. 28–35, 2011.

[10] R. Challoo, A Oladeinde, N. Yilmazer, S. Ozcelik, and L Challoo, “An overview and assessment of wireless technologies and co-existence of ZigBee, Bluetooth and Wi-Fi devices,” *Procedia Computer Science*, vol. 12, pp. 386–391, 2012.

[11] JFR, “Z-Wave Protocol Overview,” Zensys, Tech. Rep., Apr. 2006.

[12] D. Flore, “GPP Standards for the Internet-of-Things,” Feb. 2016.

[13] Ericsson, “Cellular Networks for Massive IoT,” Tech. Rep., Jan. 2016, [Online] Available: https://www.ericsson.com/assets/local/publications/white-papers/wp_iot.pdf.

[14] L. Vangelista, A. Zanella, and M. Zorzi, “Long-range IoT technologies: The dawn of LoRa™,” in *Future access enablers of ubiquitous and intelligent infrastructures*, Springer, 2015, pp. 51–58.

[15] C. Goursaud and J. M. Gorce, “Dedicated Networks for IoT: PHY / MAC State of the Art and Challenges,” *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, Oct. 2015.

[16] 3GPP, “Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT),” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 45.820, Nov. 2015, Version 13.1.0.

[17] O. Liberg, M. Sundberg, E. Wang, J. Bergman, J. Sachs, and G. Wikström, *Cellular Internet of Things: From Massive Deployments to Critical 5G Applications*. Academic Press, 2019, ISBN: 978-0-08-102902-2.

[18] Rohde and Schwarz, “Narrowband Internet of Things Measurements - Application Note 1MA296_2e,” Tech. Rep., 2019.

[19] R. S. Sinha, Y. Wei, and S.-H. Hwang, “A survey on LPWA technology: LoRa and NB-IoT,” *ICT Express*, vol. 3, no. 1, pp. 14 –21, 2017, ISSN: 2405-9595.

[20] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment,” *ICT express*, vol. 5, pp. 1–7, Mar. 2019. DOI: 10.1016/j.icte.2017.12.005.

[21] 3GPP, “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 136.304, Jul. 2017, Version 14.3.0.

[22] K. Kousias, G. Caso, Ö. Alay, A. Brunstrom, L. De Nardis, M.-G. Di Benedetto, and M. Neri, "Coverage and Deployment Analysis of Narrowband Internet of Things in the Wild," *arXiv preprint arXiv:2005.02341*, 2020.

[23] u-blox, *SARA-N2 modules. AT commands manual*, 2018. [Online]. Available: `https://www.u-blox.com/site{s/default/files/SARA-N2_ATCommands_\%28UBX-16014887\%29.pdf}`.

[24] Semtech, "Application note: Lora modulation crystal oscillator guidance, an1200.14 rev. 2," Semtech, Tech. Rep., 2017.

[25] Semtech Corporation, *Sx1301 datasheet*, SX1301, version 2.01, 2014.

[26] LoRa Alliance, *LoRaWAN™ 1.1 Specification*, Oct. 2017.

[27] K. Mikhaylov, J. Petajajarvi, and J. Janhunen, "On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-Network Interference," *arXiv preprint arXiv:1704.04257*, 2017.

[28] Semtech Corporation, *Sx1272 datasheet*, SX1272, version 2.01, 2015.

[29] ——, *Sx1276 datasheet*, SX1276, version 7, 2020.

[30] 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 Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[31] L. Vangelista, "Frequency shift chirp modulation: The lora modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, 2017.

[32] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, 2018, ISSN: 1089-7798.

[33] M. Bor and U. Roedig, "LoRa Transmission Parameter Selection," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Jun. 2017.

[34] M. Knight and B. Seeber, "Decoding LoRa: Realizing a Modern LPWAN with SDR," in *Proceedings of the GNU Radio Conference*, vol. 1, 2016.

[35] LoRa Alliance, "LoRaWAN™ 1.0.2 Specification," Jul. 2016.

[36] *The Things Network*, Available: https://www.thethingsnetwork.org.

[37] ——, “LoRaWAN™ 1.1 Regional Parameters,” 2017.

[38] CEPT, “ERC Recommendation 70-03 - Relating to the use of Short Range Devices (SRD),” CEPT ECC, Tech. Rep., 2019.

[39] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of lora networks in a smart city scenario,” in *IEEE International Conference on communications (ICC)*, IEEE, 2017, pp. 1–7.

[40] M. Capuzzo, D. Magrin, and A. Zanella, “Confirmed traffic in LoRaWAN: Pitfalls and countermeasures,” in *Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun. 2018. DOI: 10.23919/MedHocNet.2018.8407095.

[41] ns 3 App Store, *Lorawan*, https://apps.nsnam.org/app/lorawan/, 2019.

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

[43] *lorawan module*, Available: https://github.com/signetlabdei/lorawan.

[44] Marais, Jaco M. and Abu-Mahfouz, Adnan M. and Hancke, Gerhard P., “A Review of LoRaWAN Simulators: Design Requirements and Limitations,” in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019, pp. 1–6. DOI: 10.1109/IMITEC45504.2019.9015882.

[45] A. I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, “Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?” In *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2017. DOI: 10.1109/GLOCOM.2017.8254509.

[46] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.

[47] B. Reynders, Q. Wang, and S. Pollin, “A LoRaWAN module for ns-3: implementation and evaluation,” in *Proceedings of the 10th Workshop on ns-3*, Jun. 2018.

[48] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology,” in *International Conference on ITS Telecommunications (ITST)*, Dec. 2015.

[49] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks," in *IEEE SENSORS*, Oct. 2016.

[50] F. Bonafini, D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, M. Pasetti, S. Rinaldi, and E. Sisinni, "Evaluating indoor and outdoor localization services for lorawan in smart city applications," in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*, IEEE, 2019, pp. 300–305.

[51] P. J. Basford, S. J. Johnston, M. Apetroaie-Cristea, F. M. J. Bulot, and S. J. Cox, "LoRaWAN for city scale IoT deployments," in *2019 Global IoT Summit (GIoTS)*, 2019, pp. 1–6. DOI: 10.1109/GIOTS.2019.8766359.

[52] J. Haxhibeqiri, A. Karaagac, F. V. den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa indoor coverage and performance in an industrial environment: Case study," in *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017. DOI: 10.1109/etfa.2017.8247601. [Online]. Available: https://doi.org/10.1109/etfa.2017.8247601.

[53] A. V. T. Bardram, M. Delbo Larsen, K. M. Malarski, M. N. Petersen, and S. Ruepp, "LoRaWan capacity simulation and field test in a harbour environment," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 2018, pp. 193–198. DOI: 10.1109/FMEC.2018.8364064.

[54] S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez, " LoRaWAN Network for Fire Monitoring in Rural Environments," *Electronics*, vol. 9, no. 3, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9030531. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/531.

[55] D. Carrillo and J. Seki, "Rural area deployment of internet of things connectivity: LTE and LoRaWAN case study," in *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2017, pp. 1–4. DOI: 10.1109/INTERCON.2017.8079711.

[56] R. El Chall, S. Lahoud, and M. El Helou, " LoRaWAN Network: Radio Propagation Models and Performance Evaluation in Various Environments in Lebanon," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2366–2378, 2019. DOI: 10.1109/JIOT.2019.2906838.

[57] K. Kousias, G. Caso, Ö. Alay, and F. Lemic, “Empirical analysis of LoRaWAN adaptive data rate for mobile internet of things applications,” in *Proceedings of the Workshop on Wireless of the Students, by the Students, and for the Students*, 2019, pp. 9–11.

[58] O. Georgiou and U. Raza, “Low Power Wide Area Network Analysis: Can LoRa Scale?” *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.

[59] Z. Li, S. Zozor, J.-M. Drossier, N. Varsier, and Q. Lampin, “2D Time-frequency interference modelling using stochastic geometry for performance evaluation in Low-Power Wide-Area Networks,” in *IEEE International Conference on Communications (ICC)*, May 2017.

[60] J. Toussaint, N. E. Rachkidy, and A. Guitton, “Performance analysis of the on-the-air activation in LoRaWAN,” in *IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct. 2016.

[61] G. Ferré, “Collision and packet loss analysis in a LoRaWAN network,” in *European Signal Processing Conference (EUSIPCO)*, Aug. 2017.

[62] R. B. Sorensen, D. M. Kim, J. J. Nielsen, and P. Popovski, “Analysis of Latency and MAC-Layer Performance for Class a LoRaWAN,” *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, 2017.

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

[64] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, “LoRa Technology Demystified: From Link Behavior To Cell-Level Performance,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 822–834, 2020. DOI: `10.1109/twc.2019.2948872`. [Online]. Available: `https://doi.org/10.1109/twc.2019.2948872`.

[65] M. Heusse, T. Attia, C. Caillouet, F. Rousseau, and A. Duda, “Capacity of a LoRaWAN Cell,” in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2020, pp. 131–140.

[66] D. Bankov, E. Khorov, and A. Lyakhov, “Mathematical model of LoRaWAN channel access with capture effect,” in *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017.

[67] ——, “LoRaWAN Modeling and MCS Allocation to Satisfy Heterogeneous QoS Requirements,” *Sensors*, vol. 19, no. 19, p. 4204, 2019.

[68] M. Capuzzo, D. Magrin, and A. Zanella, “Mathematical Modeling of LoRa WAN Performance with Bi-directional Traffic,” in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018.

[69] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, “Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN,” *Sensors*, vol. 18, no. 7, p. 2104, 2018.

[70] F. H. Khan, R. Jurdak, and M. Portmann, “A Model for Reliable Uplink Transmissions in LoRaWAN,” in *15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 147–156.

[71] N. Kouvelas, V. Rao, and R. V. Prasad, “Employing p-CSMA on a LoRa Network Simulator,” *CoRR*, vol. abs/1805.12263, 2018. arXiv: 1805.12263 [cs.NI]. [Online]. Available: http://arxiv.org/abs/1805.12263.

[72] D. Zucchetto and A. Zanella, “Uncoordinated Access Schemes for the IoT: Approaches, Regulations, and Performance,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 48–54, 2017, ISSN: 0163-6804.

[73] N. Varsier and J. Schwoerer, “Capacity limits of LoRaWAN technology for smart metering applications,” in *IEEE International Conference on Communications (ICC)*, May 2017.

[74] 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 *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2017, pp. 1–8. DOI: 10.1109/WiMOB.2017.8115779.

[75] F. Cuomo, J. C. C. Gámez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “Towards traffic-oriented spreading factor allocations in lorawan systems,” in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2018, pp. 1–8. DOI: 10.23919/MedHocNet.2018.8407091.

[76] V. Hauser and T. Hégr, “Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure,” in *International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug. 2017.

[77] M. Slabicki, G. Premsankar, and M. Di Francesco, “Adaptive Configuration of LoRa Networks for Dense IoT Deployments,” in *16th IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*, Apr. 2018.

[78] B. Reynders, W. Meert, and S. Pollin, “Power and Spreading Factor Control in Low Power Wide Area Networks,” in *IEEE International Conference on Communications (ICC)*, May 2017.

[79] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, “Enhanced ADR for LoRaWAN networks with mobility,” in *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, IEEE, 2019, pp. 1–6.

[80] J. Park, K. Park, H. Bae, and C.-K. Kim, “EARN: Enhanced ADR with coding rate adaptation in LoRaWAN,” *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 873–11 883, 2020.

[81] *Simpy. Event Discrete Simulation for Python. [Online]*, 2016.

[82] E. Lima, J. Moraes, H. Oliveira, E. Cerqueira, S. Zeadally, and D. Rosário, “Adaptive priority-aware lorawan resource allocation for internet of things applications,” *Ad Hoc Networks*, p. 102 598, 2021.

[83] *lorawan-model*, Available: https://signetlabdei.github.io/lorawan-model/.

[84] D. Magrin, M. Capuzzo, and A. Zanella, “A Thorough Study of LoRaWan Performance Under Different Parameter Settings,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, Jan. 2020. DOI: 10.1109/jiot.2019.2946487.

[85] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[86] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme, “The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2740–2751, 2008.

[87] R. Sanchez-Iborra and M.-D. Cano, “State of the art in LP-WAN solutions for industrial IoT services,” *Sensors*, vol. 16, no. 5, 2016.

[88] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless networks*, vol. 23, no. 1, pp. 23–41, 2017.

[89] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[90] F. Bonavolontà, A. Tedesco, R. Schiano Lo Moriello, and A. Tufano, "Enabling wireless technologies for industry 4.0: State of the art," in *IEEE International Workshop on Measurement and Networking (M&N)*, IEEE, 2017, pp. 1–5.

[91] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *International Conference on Contemporary Computing and Informatics (IC3I)*, IEEE, 2014, pp. 634–638.

[92] J. P. Queralta, T. Gia, Z Zou, H Tenhunen, and T Westerlund, "Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: beyond LoRa and LoRaWAN," *Procedia Computer Science*, vol. 155, pp. 343–350, 2019.

[93] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

[94] A. Hoeller, R. D. Souza, O. L. A. López, H. Alves, M. de Noronha Neto, and G. Brante, "Analysis and performance optimization of LoRa networks with time and antenna diversity," *IEEE Access*, vol. 6, pp. 32 820–32 829, 2018.

[95] L. Tessaro, C. Raffaldi, M. Rossi, and D. Brunelli, "Lora performance in short range industrial applications," in *2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, IEEE, 2018, pp. 1089–1094.

[96] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, "Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 298–310, 2019.

[97] P. Neumann, J. Montavont, and T. Noel, “Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study,” in *IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2016, nil. DOI: 10.1109/wimob.2016.7763213. [Online]. Available: https://doi.org/10.1109/wimob.2016.7763213.

[98] H. H. R. Sherazi, L. A. Grieco, M. A. Imran, and G. Boggia, “Energy-efficient LoRaWAN for Industry 4.0 Applications,” *IEEE Transactions on Industrial Informatics*, 2020, ISSN: 1941-0050. DOI: 10.1109/TII.2020.2984549.

[99] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund, “Using LoRa for industrial wireless networks,” in *IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, 2017, pp. 1–4. DOI: 10.1109/WFCS.2017.7991972.

[100] L. Leonardi, F. Battaglia, G. Patti, and L. L. Bello, “Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018, nil. DOI: 10.1109/iecon.2018.8591568. [Online]. Available: https://doi.org/10.1109/iecon.2018.8591568.

[101] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, “Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things,” *IEEE Communications Magazine*, vol. 56, no. 2, pp. 60–67, 2018. DOI: 10.1109/mcom.2018.1700625. [Online]. Available: https://doi.org/10.1109/mcom.2018.1700625.

[102] M. Cheffena, “Propagation channel characteristics of industrial wireless sensor networks [wireless corner],” *IEEE Antennas and Propagation Magazine*, vol. 58, no. 1, pp. 66–73, 2016.

[103] Y. Ai, M. Cheffena, and Q. Li, “Radio frequency measurements and capacity analysis for industrial indoor environments,” in *9th European Conference on Antennas and Propagation (EuCAP)*, IEEE, 2015, pp. 1–5.

[104] M. Nakagami, “The m-distribution—a general formula of intensity distribution of rapid fading,” in *Statistical methods in radio wave propagation*, Elsevier, 1960, pp. 3–36.

[105] J. Ferrer-Coll, P. Ängskog, C. Elofsson, J. Chilo, and P. Stenumgaard, "Antenna Cross Correlation and Ricean K-Factor Measurements in Indoor Industrial Environments at 433 and 868 MHz," *Wireless personal communications*, vol. 73, no. 3, pp. 587–593, 2013.

[106] K. Blackard, T. Rappaport, and C. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 7, pp. 991–1001, 1993. DOI: 10.1109/49.233212. [Online]. Available: https://doi.org/10.1109/49.233212.

[107] M. G. Sanchez, I. Cuinas, and A. V. Alejos, "Interference and impairments in radio communication systems due to industrial shot noise," in *IEEE International Symposium on Industrial Electronics*, IEEE, 2007, pp. 1849–1854.

[108] "TR 103 526 v1.1.1," ETSI, Tech. Rep., 2018.

[109] ETSI, "300 220-1 (v2. 4.1)," *Electromagnetic compatibility and Radio spectrum Matters (ERM)*, 2012.

[110] D. Zucchetto and A. Zanella, "Uncoordinated access schemes for the IoT: approaches, regulations, and performance," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 48–54, 2017.

[111] J. Ortin, M. Cesana, and A. Redondi, "Augmenting LoRaWAN Performance With Listen Before Talk," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3113–3128, 2019. DOI: 10.1109/twc.2019.2910512. [Online]. Available: https://doi.org/10.1109/twc.2019.2910512.

[112] L. Leonardi, L. Lo Bello, F. Battaglia, and G. Patti, "Comparative Assessment of the LoRaWAN Medium Access Control Protocols for IoT: Does Listen before Talk Perform Better than ALOHA?" *Electronics*, vol. 9, no. 4, p. 553, 2020.

[113] "RF Performance Tests of LoRa Gateways without an External Network Server - Application Note RWC5020A and PC Application Software, RAN1940001-rev2," RedwoodComm, Tech. Rep.

[114] 3GPP, "Technical Specification Group Services and System Aspects; Service requirements for cyber-physical control applications in vertical domains; Stage 1," Tech. Rep. 22.104 V.17.4.0, 2020.

[115] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, “Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges,” *IEEE Communications Surveys & Tutorials*, pp. 3417–3442, Mar. 2019.

[116] S. Sekander, H. Tabassum, and E. Hossain, “Multi-tier drone architecture for 5G/B5G cellular networks: Challenges, trends, and prospects,” *IEEE Communications Magazine*, vol. 56, no. 3, pp. 96–103, Mar. 2018.

[117] T. Long, M. Ozger, O. Cetinkaya, and O. B. Akan, “Energy neutral Internet of Drones,” *IEEE Communications Magazine*, vol. 56, no. 1, pp. 22–28, Jan. 2018.

[118] B. Hament and P. Oh, “Unmanned aerial and ground vehicle (UAV-UGV) system prototype for civil infrastructure missions,” in *IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Jan. 2018, pp. 1–4.

[119] S. Zakaria, M. R. Mahadi, A. F. Abdullah, and K. Abdan, “Aerial platform reliability for flood monitoring under various weather conditions: A review,” in *GeoInformation for Disaster Management Conference*, Springer, Mar. 2018, pp. 295–314.

[120] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, “An autonomous multi-UAV system for search and rescue,” in *Proceedings of the 1st Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ACM, May 2015, pp. 33–38.

[121] H. Huang and A. V. Savkin, “Towards the Internet of Flying Robots: A survey,” *Sensors*, vol. 18, no. 11, p. 4038, Nov. 2018.

[122] V. Sharma, I. You, G. Pau, M. Collotta, J. Lim, and J. Kim, “LoRaWAN-based energy-efficient surveillance by drones for intelligent transportation systems,” *Energies*, vol. 11, no. 3, p. 573, Mar. 2018.

[123] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, “Farmbeats: An IoT platform for data-driven agriculture,” in *14th Symposium on Networked Systems Design and Implementation (NSDI)*, USENIX, Sep. 2017, pp. 515–529.

[124] A. Ismail, B. Bagula, and E. Tuyishimire, “Internet-of-things in motion: A UAV coalition model for remote sensing in smart cities,” *Sensors*, vol. 18, no. 7, p. 2184, Jul. 2018.

[125] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, “Decentralized planning and control for UAV–UGV cooperative teams,” *Autonomous Robots*, vol. 42, no. 8, pp. 1601–1618, Dec. 2018.

[126] J. Chen, T. Ma, and C. Su, “The study of UAV intelligent support mode based on battlefield networks,” in *2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, IEEE, Mar. 2017, pp. 1912–1915.

[127] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos, “A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors,” in *IEEE Conference on Control Applications (CCA)*, IEEE, Sep. 2016, pp. 519–525.

[128] C. V. Angelino, V. R. Baraniello, and L. Cicala, “UAV position and attitude estimation using IMU, GNSS and camera,” in *15th International Conference on Information Fusion*, IEEE, Jul. 2012, pp. 735–742.

[129] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, “Precision wildlife monitoring using unmanned aerial vehicles,” *Scientific reports*, vol. 6, no. 1, pp. 1–7, Mar. 2016.

[130] O. Alvear, C. T. Calafate, N. R. Zema, E. Natalizio, E. Hernández-Orallo, J.-C. Cano, and P. Manzoni, “A discretized approach to air pollution monitoring using UAV-based sensing,” *Mobile Networks and Applications*, vol. 23, no. 6, pp. 1693–1702, Dec. 2018.

[131] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, “A review on UAV-based applications for precision agriculture,” *Information*, vol. 10, no. 11, p. 349, Nov. 2019.

[132] J. Langhammer, J. Bernsteinová, and J. Miřijovský, “Building a high-precision 2D hydrodynamic flood model using UAV photogrammetry and sensor network monitoring,” *Water*, vol. 9, no. 11, p. 861, Nov. 2017.

[133] B. D. Anderson, B. Fidan, C. Yu, and D. Walle, “UAV formation control: Theory and application,” in *Recent advances in learning and control*, Springer, Jan. 2008, pp. 15–33.

[134] F. Mason, F. Chiariotti, M. Capuzzo, D. Magrin, A. Zanella, and M. Zorzi, “Combining LoRaWAN and a new 3D motion model for remote UAV tracking,” in *INFOCOM Workshop on Wireless Sensor, Robot and UAV Networks (INFOCOM WiSARN)*, IEEE, Jul. 2020.

[135] R. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME - Journal of basic Engineering*, vol. 82, pp. 35–45, 1960.

[136] P. Del Moral, “Non-linear filtering: Interacting particle resolution,” *Markov processes and related fields*, vol. 2, no. 4, pp. 555–581, 1996.

[137] F. Mason, M. Giordani, F. Chiariotti, A. Zanella, and M. Zorzi, “An adaptive broadcasting strategy for efficient dynamic mapping in vehicular networks,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5605–5620, Aug. 2020.

[138] M. Tsogas, A. Polychronopoulos, and A. Amditis, “Unscented Kalman filter design for curvilinear motion models suitable for automotive safety applications,” in *7th International Conference on Information Fusion*, 2005. DOI: 10.1109/ICIF.2005.1592006.

[139] J.-D. M. M. Biomo, T. Kunz, and M. St-Hilaire, “An enhanced Gauss-Markov mobility model for simulations of unmanned aerial ad hoc networks,” in *7th IFIP Wireless and Mobile Networking Conference (WMNC)*, IEEE, May 2014, pp. 1–8.

[140] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, “A mobility model for UAV ad hoc network,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2014, pp. 383–388.

[141] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, “A smooth-turn mobility model for airborne networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3359–3370, Mar. 2013.

[142] J. Tiemann, F. Schweikowski, and C. Wietfeld, “Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments,” in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Oct. 2015, pp. 1–7.

[143] R. Opromolla, G. Inchingolo, and G. Fasano, “Airborne visual detection and tracking of cooperative UAVs exploiting deep learning,” *Sensors*, vol. 19, no. 19, p. 4332, Jan. 2019.

[144] D. Fan, F. Gao, B. Ai, G. Wang, Z. Zhong, Y. Deng, and A. Nallanathan, “Channel estimation and self-positioning for UAV swarm,” *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7994–8007, Aug. 2019.

[145] R. Mendrzik, D. Cabric, and G. Bauch, “Error bounds for Terahertz MIMO positioning of swarm UAVs for distributed sensing,” in *International Conference on Communications Workshops (ICC Workshops)*, IEEE, May 2018.

[146] A. Farhad, D.-H. Kim, S. Subedi, and J.-Y. Pyun, “Enhanced LoRaWAN adaptive data rate for mobile internet of things devices,” *Sensors*, vol. 20, no. 22, p. 6466, 2020.

[147] Archimedes of Syracuse, *On Spirals*, $\sim$225BC.

[148] R. Van Der Merwe *et al.*, “Sigma-point Kalman filters for probabilistic inference in dynamic state-space models,” PhD thesis, OGI School of Science & Engineering at OHSU, 2004.

[149] T. Saadawi and A. Ephremides, “Analysis, stability, and optimization of slotted aloha with a finite number of buffered users,” *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 680–689, Jun. 1981.

[150] V. Anantharam, “The stability region of the finite-user slotted aloha protocol,” *IEEE Transactions on information Theory*, vol. 37, no. 3, pp. 535–540, May 1991.

[151] D. Sant, “Throughput of unslotted aloha channels with arbitray packet interarrival time distributions,” *IEEE Transactions on Communications*, vol. 28, no. 8, pp. 1422–1425, Aug. 1980.

[152] R. D. Yates and S. K. Kaul, “Age of information in uncoordinated unslotted updating,” in *International Symposium on Information Theory (ISIT)*, IEEE, Jun. 2020, pp. 1759–1764.

[153] M. Fonder and M. V. Droogenbroeck, “Mid-air: A multi-modal dataset for extremely low altitude drone flights,” in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019.

[154] “Global Positioning System (GPS) Standard Positioning Service (SPS) performance analysis report,” William J. Hughes Technical Center, Navigation Systems Verification and Monitoring Branch, Tech. Rep. 86, Jul. 2014.

[155] B. Kim and K. Yi, “Probabilistic and holistic prediction of vehicle states using sensor fusion for application to integrated vehicle safety systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2178–2190, May 2014.

[156] G. Falco, M. Pini, and G. Marucco, "Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios," *Sensors*, vol. 17, no. 2, p. 255, Feb. 2017.

[157] V. Tahiliani and M. Dizalwar, "Green iot systems: An energy efficient perspective," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, IEEE, 2018, pp. 1–6.

[158] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green iot: An investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15 667–15 681, 2017.

[159] F. Fraternali, B. Balaji, Y. Agarwal, L. Benini, and R. Gupta, "Pible: Battery-free Mote for Perpetual Indoor BLE Applications," in *Proceedings of the 5th Conference on Systems for Built Environments*, Shenzhen, China, 2018, pp. 168–171.

[160] C. Delgado, J. M. Sanz, C. Blondia, and J. Famaey, "Batteryless lorawan communications using energy harvesting: Modeling and characterization," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2694–2711, 2021. DOI: 10.1109/JIOT.2020.3019140.

[161] A. Sabovic, C. Delgado, D. Subotic, B. Jooris, E. D. Poorter, and J. Famaey, "Energy-Aware Sensing on Battery-Less LoRaWAN Devices with Energy Harvesting," *Electronics*, vol. 9, no. 6, p. 904, 2020.

[162] E. Gindullina, L. Badia, and X. Vilajosana, "Energy Modeling and Adaptive Sampling Algorithms for Energy-harvesting Powered Nodes with Sampling Rate Limitations," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 3, e3754, 2020.

[163] G. Peruzzi and A. Pozzebon, "A review of energy harvesting techniques for low power wide area networks (lpwans)," *Energies*, vol. 13, no. 13, p. 3433, 2020.

[164] M. Mabon, M. Gautier, B. Vrigneau, M. Le Gentil, and O. Berder, "The smaller the better: Designing solar energy harvesting sensor nodes for long-range monitoring," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[165] T. Polonelli, D. Brunelli, M. Guermandi, and L. Benini, "An accurate low-cost Crackmeter with LoRaWAN communication and energy harvesting capability," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, vol. 1, 2018, pp. 671–676.

[166] P. Boccadoro, B. Montaruli, and L. A. Grieco, "Quakesense, a LoRa-compliant earthquake monitoring open system," in *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, IEEE, 2019, pp. 1–8.

[167] M. M. Sandhu, K. Geissdoerfer, S. Khalifa, R. Jurdak, M. Portmann, and B. Kusy, "Towards Energy Positive Sensing using Kinetic Energy Harvesters," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Austin, Texas, USA, 2020, pp. 1–10. DOI: 10.1109/PerCom45495.2020.9127356.

[168] J. Finnegan, K. Niotaki, and S. Brown, "Exploring the boundaries of ambient rf energy harvesting with lorawan," *IEEE Internet of Things Journal*, 2020.

[169] G. Loubet, A. Takacs, E. Gardner, A. De Luca, F. Udrea, and D. Dragomirescu, "Lorawan battery-free wireless sensors network designed for structural health monitoring in the construction domain," *Sensors*, vol. 19, no. 7, p. 1510, 2019.

[170] F. Orfei, C. B. Mezzetti, and F. Cottone, "Vibrations powered lora sensor: An electromechanical energy harvester working on a real bridge," in *2016 IEEE SENSORS*, 2016, pp. 1–3.

[171] M. Capuzzo, *Capacitor-ns3*, https://github.com/signetlabdei/capacitor-ns3, 2021.

[172] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking Low-Power Energy Harvesting Devices: Measurements and Algorithms," in *Proc. IEEE INFOCOM'11*, 2011.

[173] "UM2115 User manual - Discovery kit for LoRaWAN, Sigfox, and LPWAN protocols with STM32L0," ST semiconductors, Tech. Rep., version Rev 5, Jun. 2018.

[174] GSMA, "Mobile IoT in the 5G future - NB-IoT and LTE-M in the context of 5G," Tech. Rep., 2018.

[175] Ericsson, "Ericsson Mobility Report - June 2020," Tech. Rep.

[176] .

[177] GSMA, "NB-IoT Deployment Guide – Release 3," Tech. Rep., 2019.

[178] *China unicom pilots nb-iot for smart parking, water metering and smart lighting*. [Online]. Available: https://www.gsma.com/iot/mobile-iot-pilots-operator/china-unicom-2/.

[179] R. Cockle, *Nb-iot now exceeds 100 million connections, and will prove an essential tool in adapting to covid-19*, 2020. [Online]. Available: `https://www.gsma.com/iot/news/nb-iot-now-exceeds-100-million-connections-and-will-prove-an-essential-tool-in-adapting-to-covid-19/`.

[180] J. Lee and J. Lee, "Prediction-based energy saving mechanism in 3GPP NB-IoT networks," *Sensors*, vol. 17, no. 9, p. 2008, 2017.

[181] P.-Y. Liu, K.-R. Wu, J.-M. Liang, J.-J. Chen, and Y.-C. Tseng, "Energy-Efficient Uplink Scheduling for Ultra-Reliable Communications in NB-IoT Networks," in *Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, 2018, pp. 1–5.

[182] S.-M. Oh and J. Shin, "An efficient small data transmission scheme in the 3GPP NB-IoT system," *IEEE Communications Letters*, vol. 21, no. 3, pp. 660–663, 2016.

[183] B. Martinez, F. Adelantado, A. Bartoli, and X. Vilajosana, "Exploring the Performance Boundaries of NB-IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5702–5712, 2019.

[184] C. Y. Yeoh, A. bin Man, Q. M. Ashraf, and A. K. Samingan, "Experimental assessment of battery lifetime for commercial off-the-shelf NB-IoT module," in *International Conference on Advanced Communication Technology (ICACT)*, IEEE, 2018, pp. 223–228.

[185] S. Duhovnikov, A. Baltaci, D. Gera, and D. A. Schupke, "Power Consumption Analysis of NB-IoT Technology for Low-Power Aircraft Applications," in *5th World Forum on Internet of Things (WF-IoT)*, IEEE, 2019, pp. 719–723.

[186] *Sara-n2 series power-optimized nb-iot (lte cat nb1) modules data sheet*, UBX-15025564, Rev. R18, u-blox, Nov. 2019.

[187] *Quectel bc95-g multi-band nb-iot module with ultra-low power consumption*, Rev. V1.7, Quectel, 2019.

[188] J.-B. Landre, Z. El Rawas, and R. Visoz, "LTE performance assessment prediction versus field measurements," in *24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2013, pp. 2866–2870.

[189] "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.214, Jul. 2012, Version 13.4.0.

[190] P. Joshi, D. Colombi, B. Thors, L.-E. Larsson, and C. Törnevik, "Output power levels of 4G user equipment and implications on realistic RF EMF exposure assessments," *IEEE Access*, vol. 5, pp. 4545–4550, 2017.

[191] A. Elmokashfi, A. Kvalbein, T. Dreibholz, and C. Jarvis, "Norske mobilnett i 2018," Centre for Resilient Networks and Applications, Tech. Rep., 2019.

[192] J. Mocnej, A. Pekar, W. K. Seah, and I. Zolotova, *Network traffic characteristics of the IoT application use cases*. School of Engineering and Computer Science, Victoria University of Wellington, 2018.

[193] P. Andres-Maldonado, M. Lauridsen, P. Ameigeiras, and J. M. Lopez-Soler, "Analytical Modeling and Experimental Validation of NB-IoT Device Energy Consumption," *IEEE Internet of Things Journal*, 2019.

[194] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Optimized LTE data transmission procedures for IoT: Device side energy consumption analysis," in *International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2017, pp. 540–545.

[195] A. Sørensen, M. Jérôme Remy, and N. Kjettrup, "An Empirical Estimation of the Battery Lifetime for LTE-M and NB-IoT Devices," Master's thesis, Aalborg University, 2018.

[196] M. Lauridsen, R. Krigslund, M. Rohr, and G. Madueno, "An Empirical NB-IoT Power Consumption Model for Battery Lifetime Estimation," in *Vehicular Technology Conference*, IEEE, 2018, pp. 1–5.

[197] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovács, P. Mogensen, and M. Sorensen, "Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT," in *Vehicular Technology Conference*, IEEE, 2017, pp. 1–5.

[198] A. K. Sultania, P. Zand, C. Blondia, and J. Famaey, "Energy Modeling and Evaluation of NB-IoT with PSM and eDRX," in *Globecom Workshops (GC Wkshps)*, IEEE, 2018, pp. 1–7.

[199] R. Harwahyu, R.-G. Cheng, W.-J. Tsai, J.-K. Hwang, and G. Bianchi, “Repetitions Versus Retransmissions: Tradeoff in Configuring NB-IoT Random Access Channels,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3796–3805, 2019.

[200] L. Feltrin, M. Condoluci, T. Mahmoodi, M. Dohler, and R. Verdone, “NB-IoT: performance estimation and optimal configuration,” in *European Wireless 2018*, VDE, 2018, pp. 1–6.

[201] M. El Soussi, P. Zand, F. Pasveer, and G. Dolmans, “Evaluating the performance of eMTC and NB-IoT for smart city applications,” in *International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–7.

[202] R. Mozny, P. Masek, M. Stusek, K. Zeman, A. Ometov, and J. Hosek, “On the Performance of Narrow-band Internet of Things (NB-IoT) for Delay-tolerant Services,” in *International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2019, pp. 637–642.

[203] E. A. Wan and R. V. D. Merwe, “The unscented Kalman filter for nonlinear estimation,” in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.