Large Scale Analysis and Scalability Enhancements for Low Power Wide Area Networks

Joseph Finnegan

A dissertation submitted for the degree of

Doctor of Philosophy

Department of Computer Science
Faculty of Science and Engineering
National University of Ireland Maynooth

October 2020.

Head of Department: Prof. Joseph Timoney

Supervisor: Dr. Stephen Brown

Abstract

The developing Internet of Things is leading to a much broader range of connected devices across many different industry sectors. The new paradigm of Low Power Wide Area Networks (LPWAN) contributes to this by enabling the deployment of cheap low-power wireless devices, allowing for pervasive smart city applications. However, research into LPWANs is in the preliminary stages, with much analysis to be done to quantify the actual limits of proposed protocols. The energy efficiency and scalability of these protocols are key if these technologies are to be deployed effectively in densely populated cities. The goal of this work is to quantify and improve the LPWAN paradigm in these two dimensions.

The first contribution is an energy consumption analysis of the LoRaWAN protocol. A study of the comparative energy consumption rate of the primary LPWAN technologies is performed, based on the physical and MAC layer states of the protocols. Then, an analysis of the feasibility of the powering of LoRaWAN applications fully through the harvesting of ambient RF energy is presented.

The next contribution is a quantification of the energy efficiency and scalability of bi-directional LPWANs, using LoRaWAN as a case study. This is performed through the implementation of a LoRaWAN energy model, LoRaWAN Class B, and the Adaptive Data Rate (ADR) of LoRaWAN in simulation. This work for the first time fully implements previously unsimulated sections of the protocol, enabling the simulation of realistic large scale LoRaWAN networks. Simulations of LoRaWAN networks under a variety of conditions are performed, with large networks of devices running applications with realistic traffic requirements. This analysis can be considered state-of-the-art as it constitutes the first time the analysed LoRaWAN features have been implemented in the used simulator.

Through the previous analysis, bottlenecks to performance for LoRaWAN networks are identified. The next contribution consists of enhancements to the ADR of LoRaWAN, which increases the scalability of the network and fixes issues which currently can limit the network adaptability. The final contribution is a novel lightweight collision avoidance algorithm which mitigates the effects of collisions from predictive traffic, based on the properties of LPWAN traffic. The scheme increases the scalability of the network while maintaining the energy efficiency of the protocol.

The work presented in this thesis enhances the sustainable operation of LPWANs in denser environments. This work is validated through analytical models and simulation. The applicability of this work to other network wireless protocols is examined and potential future research directions are discussed. The contributions presented in this thesis advance the evolution of LPWANs by enabling the analysis and enhancing the scalability of large scale deployments, further accelerating the acceptability and pervasiveness of LPWAN networks.

Statement of Originality

I have read and understood the Departmental policy on plagiarism.

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education.

Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Signed:

_____

Joseph Finnegan

18th April 2021

Acknowledgements

This thesis marks the end of a journey, and not one that I would have been able to finish by myself. I would firstly like to express my sincerest gratitude to my supervisor, Dr. Stephen Brown, for his guidance and support throughout this work and over these past four years. Thank you for teaching me how to how to research, how to write, and how to learn.

I would like to thank Dr. M. Shahwaiz Afaqui for his advice and his help in all areas of this work, and for his friendship. This thesis would look significantly different, and much worse, without his input. I would also like to thank Dr. Afaqui and Dr. Xavier Vilajosana for the great opportunity to undertake a research stay at the Universitat Oberta de Catalunya, even if it was unfortunately cut short by COVID-19.

I'd like to thank Declan Byrne, Paval Afanasyev, Dr. Kyriaki Niotaki, and all of my other colleagues in Maynooth and in Barcelona, who have made this experience so worthwhile. I'd like to thank all of my friends in Ireland and abroad for keeping me sane during this PhD, including Tim Quinlan, Jamie Kirby, Dr. Matthew Gleeson, Marta Piętowska, David Riand, Kevin Farrell, John Herbage, and Dr. Trent Rogers.

Finally, I'd like to thank my family: my brother Michael, my sister Sinead, their husbands Xavier and Brendan, and especially my parents, for your love, support, encouragement, and free dinners over these long four years. *Go raibh míle maith agaibh.* Next meal is on me.

Contents

List of Figures

List of Tables

List of Code Extracts

List of Algorithms

List of Acronyms

3GPP 3rd Generation Partnership Project.

ADR Adaptive Data Rate.

C-IoT Cellular-IoT.

CDMA Code-Division Multiple Access.

CRC cyclic redundancy check.

CSS Chirp Spread Spectrum.

D7AP Dash 7 Alliance Protocol.

DBPSK Differential Binary Phase-Shift Keying.

DFT discrete Fourier transform.

DSSS Direct-Sequence Spread Spectrum.

EC-GSM-IoT Extended Coverage - GSM - Internet of Things.

eDRX extended DRX.

eGPRS Enhanced General Packet Radio Service.

EH energy harvesting.

ERP effective radiated power.

ETSI European Telecommunications Standards Institute.

FEC Forward Error Correction.

FSK Frequency-Shift Keying.

GFSK Gaussian Frequency Shift Keying.

GMSK Gaussian Minimum Shift Keying.

GPRS General Packet Radio Service.

GPS Global Positioning System.

GSM Global System for Mobile Communications.

HARQ Hybrid Automatic Repeat Request.

HTC Human-Type Communications.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

ISM Industrial, Scientific and Medical.

ITU International Telecommunication Union.

LBT Listen-Before-Talk.

LoRa Long Range.

LoRaWAN Long Range Wide Area Network.

LPWAN Low Power Wide Area Network.

LTE Long-Term Evolution.

LTE-M LTE-Machine Type Communication.

LTS Lightweight Timeslot Scheduling.

MAC Medium Access Control.

MCL maximum coupling loss.

mMTC Massive Machine Type Communications.

NB-IoT Narrowband Internet of Things.

ns-3 Network Simulator 3.

OFDM Orthogonal Frequency-Division Multiplexing.

OFDMA Orthogonal Frequency-Division Multiple Access.

OOK On-Off Keying.

PA Power Amplifier.

PDR Packet Delivery Ratio.

PHY Physical.

PLL phased-locked loop.

PMU power management unit.

QPSK Quadrature Phase Shift Keying.

RF Radio Frequency.

RFDMA Random Frequency-Division Multiple Access.

RPMA Random Phase Multiple Access.

RSSI Received Signal Strength Indicator.

SC-FDMA Single-Carrier Frequency-Division Multiple Access.

SF spreading factor.

SINR signal-to-interference-plus-noise ratio.

TDMA Time-Division Multiple Access.

UNB ultra-narrowband.

uRLLC Ultra-Reliable and Low-Latency Communications.

CHAPTER 1

Introduction

The emerging Internet of Things (IoT) enables advanced services by equipping physical objects with sensing, communication and processing capabilities [1]. Equipped objects enable ambient intelligence, and, beyond this, enable the development of combined intelligent systems formed of disparate context-aware applications. These systems are expected to lead to self-organising networks of devices, achieving optimised results in common goals [2]. Potential IoT applications are numerous, across many different sectors including smart industry, smart home and buildings, smart transport, smart health, smart city, and smart agriculture [3]. The developing paradigm of the IoT is leading to a much broader range of connected devices, including consumer electronics, wearables, connected cars, and industrial sensors and meters [4]. Substantial growth in IoT applications is widely predicted, with Gartner projecting that the number of IoT-connected devices will increase to 43 billion by 2023 [5], and Statista to 75 billion by 2025 [6]. McKinsey project an economic impact from IoT of $11 trillion per year by 2025 [7], as more widespread and advanced applications of IoT move beyond anomaly detection and control and towards optimisation and prediction.

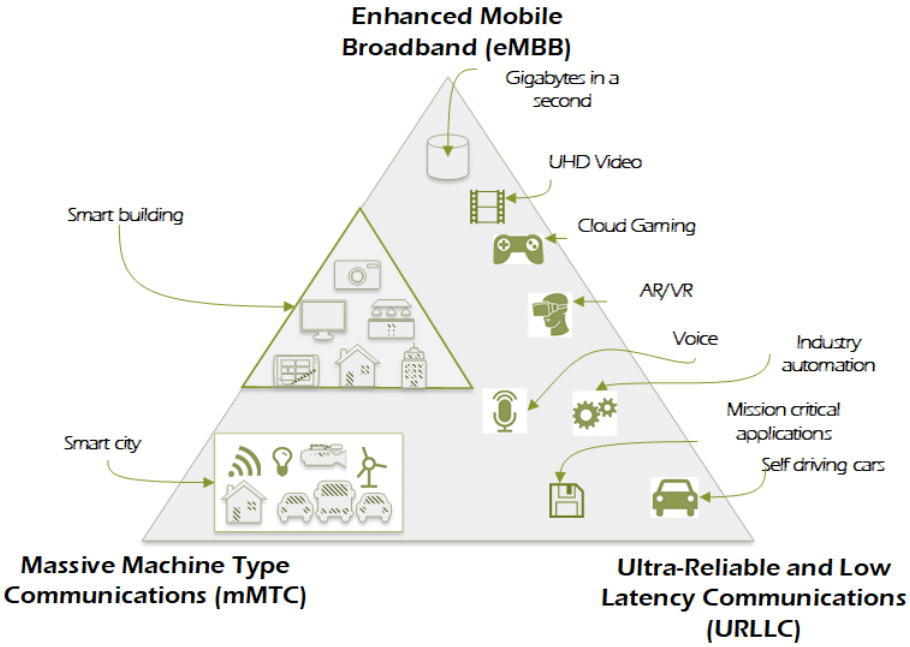


Figure 1.1: 5G Mobile Network Services, as Classified by the ITU

To meet the requirements of diverse IoT applications, new and evolving wireless technologies are expected to provide massive connectivity, increased security, global coverage, ultra-low latency, ultra-reliable communications, and higher throughput. These goals will be achieved through improvements in antenna design, use of higher frequencies, increased network heterogeneity and interoperability, and the development of new wireless standards [8]. The International Telecommunication Union (ITU) have classified next-generation 5G services into three categories, as shown in Figure 1.1 [9]. These categories are:

- Enhanced Mobile Broadband (eMBB) for multimedia and communications.
- Ultra-Reliable and Low-Latency Communications (uRLLC) for critical communications.
- Massive Machine Type Communications (mMTC) for infrequently transmitting devices without strict requirements on latency.

IoT applications are in general classified into the latter two categories. Many of these applications that fall into the category of mMTC are not well served by current common wireless technologies, because targeted performance metrics cannot be achieved in terms of energy efficiency (LTE, IEEE 802.11) or coverage (IEEE 802.15.4) [4]. While these applications could potentially be served using IEEE 802.15.4 using large scale mesh-based protocols, this can result in systems with multiple repeaters, significant control plane communication, and complex routing protocols [10]. This has led to the development of a new paradigm of networking protocols to handle this type of traffic. This paradigm is known as Low Power Wide Area Network (LPWAN). There are currently many different protocols all developing in parallel, competing to become the dominant option for this form of IoT traffic.

LPWAN technologies are characterised by a focus on low power operation, wide coverage, low cost, and scalability. This is enabled through the constraints of a low data rate and restricted limits on latency. In the mMTC space, it is expected that cellular 5G and selected LPWAN technologies will interwork with each other in the future, with 5G providing the backhaul required to enable LPWAN infrastructure [4].

Overall, there are four main identifiable attributes of state-of-the-art LPWAN technologies:

Long Range: LPWAN technologies typically target a range of 5-10 km in rural areas, and 1-2 km in urban areas. This long range allows the use of a lightweight Medium Access Control (MAC) layer and a star topology where even mobile devices can transmit to gateway nodes without the use of repeaters or multi-hop. This enables the coverage of large areas with reduced investment in infrastructure. This long range is enabled by transmission in sub-GHz Industrial, Scientific and Medical (ISM) spectrum, which has favourable characteristics for long range transmission in comparison to the use of the 2.4GHz band, and the use of modulation techniques that enable reception well below the noise floor.[1]

Energy Efficiency: A battery-powered device should be able to operate autonomously for up to 10 years without interruption, enabling a "deploy-and-forget" strategy for data collection [11]. This is enabled by a lightweight MAC, ALOHA-based access[2], star topology, and energy harvesting approaches, in order to reduce device complexity and active time, and extend battery life as long as possible.

Scalability: In order to allow the covering of environments such as cities without an excessive number of gateway nodes, an LPWAN gateway should be able to serve thousands of devices, including mobile devices. This is enabled by configurable data rates, and multiple access techniques such as orthogonal data rates or the use of a high number of selectable channels. Devices sending mMTC traffic generally have a typically low approximate amount of frames that are required to be sent per day, and so in the context of this type of traffic, there is little value in focusing on maximising the amount of frames that a single device can send. Rather, a more appropriate approach is to define realistic data throughput requirements for individual devices, and then focusing on maximising the number of devices in the network that can reliably deliver this traffic. Therefore, in the context of this thesis, scalability can be defined as the number of devices that can be connected to a single LoRaWAN gateway, sending frames following a defined traffic pattern, while maintaining an acceptably high level of Packet Delivery Ratio (PDR) across the network. The defined traffic pattern (including ratio of uplink to downlink traffic), and the acceptable level of PDR is dependent on the use case being modelled.

Low Cost: A typical LPWAN device should cost as little as possible and ensure deployments with minimal maintenance required, enabling the development of profitable IoT applications. This is enabled through the reduction of complexity on the device side and (for some LPWAN technologies)

[1] The newest generation of LoRa transceivers developed by Semtech enable transmission of LoRa frames in the 2.4GHz band. However, as this is a very recent development, and challenges of LoRa co-existance in this band differ significantly to LoRa at 868MHz, this mode is not a focus of this thesis.

[2] ALOHA-based transmission involves no scheduled transmission slots or listen-before-talk for devices, so devices only have to remain out of sleep mode when transmitting or receiving data. This minimises the energy consumption, but does not provide efficient use of a shared channel.

the use of license-free ISM bands.

These attributes are enabled through restrictions on the following factors [11]:

Data Rate: The low achievable data rate resulting from the modulation format, and (for some LPWAN technologies) the access restrictions on the sub-GHz ISM bands [12] result in only a low amount of mostly non-bursty traffic being transmittable by these devices.

Achievable latency: The use of a lightweight ALOHA-based MAC and low data rates mean that reception cannot be guaranteed within a strict low latency window. This reduces the scope of use cases that can be suitably covered by LPWAN technologies. Additionally, the regulations of the sub-GHz ISM bands impose a duty cycle limit on devices, further reducing the guarantee-able limit on latency that can be achieved.

Hardware: The low cost requirement limits typically available hardware for devices. In particular, it cannot be assumed that an LPWAN device will have access to a very accurate clock source or a sizeable amount of available memory.

1.1 Research Question

From the identified main attributes, we can derive the core research question of this thesis, which is: *How can the scalability of LPWAN networks be improved, without sacrificing the low cost and low power operation of devices?*

This is quite a broad question, and so this can be broken down in a set of objectives which have been tackled in this thesis:

- There are many different LPWAN technologies. *What are the primary differences and core similarities between LPWAN technologies? And how does the energy consumption rate of different LPWAN technologies differ, when serving the same application?*
- If there was more energy available and reduced energy consumption for a device, there would be more flexibility in the MAC layer design, as the impact of network overhead on the device lifetime would be reduced. *How can the battery lifetime of LoRaWAN devices be extended? And what forms of energy harvesting are applicable for LoRaWAN devices?*
- LPWAN networks are expected to scale, where potentially thousands of devices could be deployed per km^2 in urban areas. *What is an achievable number of devices that can be served by a single LoRaWAN gateway, and how is this affected by the adaptive features of the protocol?*
- Intelligent and reactive applications, which will go beyond data reporting, will have stronger requirements on downlink feedback. *How is the scalability of a LoRaWAN network affected when downlink feedback requirements for underlying applications are increased?*
- The expected format of traffic that is expected to flow through LoRaWAN networks is mMTC,

as identified by the ITU. *How do the characteristics of mMTC traffic affect the scalability of LoRaWAN networks?*

Technologies in the LPWAN paradigm are still in active development. While the intricacies of individual technologies are designed to combat the known issues of ALOHA-based access, the limits of these technologies in terms of energy efficiency and scalability is still unknown. The duty cycle regulations of the EU 868 MHz band are known to fundamentally limit the amount of downlink communication from network gateways, but how this affects the number of potential devices that can be served by an individual gateway has not been adequately explored.

1.1.1 A Focus on LoRaWAN

In the research presented in this thesis, the behaviour of LPWAN devices and large-scale LPWAN networks is explored to investigate the limits of performance. A particular focus is analysis of the LoRaWAN protocol. Findings are used to develop novel modifications to the protocol to enable greater overall performance of large-scale networks in terms of the key identified attributes, without impacting the low cost requirement. This research is conducted as part of CONNECT, an Ireland-wide Science Foundation Ireland-funded research collaboration for Future Networks and Communications. The LPWAN protocol LoRaWAN has been chosen as a platform for implementation and validation due to its open nature, and because of the availability of Pervasive Nation, a nation-wide LoRaWAN testbed developed and maintained by CONNECT researchers.

The research approach is the use of analysis and extended realistic simulation to understand behaviour and identify bottlenecks in the system. These results are used to develop novel enhancements to the system, which are implemented and evaluated through simulation. The use of simulation tools enables the evaluation of large scale systems of thousands of active nodes, far beyond what would be realistic to implement in hardware. Simulation also enables thorough and accurate collection of data relevant to the experiment. The simulator used in this research is Network Simulator 3 (ns-3), an open source discrete-event network simulator frequently used in research for the evaluation of wireless networks [13].

1.2 Contributions and Publications

The aim of this research and objective of this thesis is to improve the performance and scalability of LPWAN networks. This is achieved through analysis of the behaviour of realistic LPWAN devices in terms of energy efficiency and reliability, as the network scales to a density to be expected of an urban environment. This is performed to demonstrate whether LPWAN protocols can achieve the necessary performance at scale to provide the required quality of service for IoT applications with mMTC traffic. The results are used to develop adaptions to LPWAN devices and networks designed to increase overall performance, whether it be alternative energy sources for long-term deployments, or modifications to the MAC layer that increase potential scalability while maintaining

the lightweight approach. A secondary goal is the development of tools and simulation software in order to facilitate further research into realistic LoRaWAN networks. All developed code has been released open-source on GitHub. As has been described, there are a number of different LPWAN protocols, with some unifying factors; the analysis and development are based on LoRaWAN as it the most open protocol in the space and the focus of the most attention in research, but the underlying concepts are generally applicable in any protocol serving mMTC traffic.

The majority of the work presented in this thesis has been published or submitted for publication.

The first publication focuses on a comparative analysis of LPWAN technologies. The remaining publications can be grouped into three main parts: the first focuses on an energy efficiency evaluation of LoRaWAN through analytical means and hardware analysis, and the second concentrates on large scale analysis of the LoRaWAN protocol in ns-3, made possible through new features implemented by the researcher. The third builds on the results of the second to propose and implement a lightweight timeslotting approach that is particularly suited for operation in the sub-GHz bands and serving of mMTC traffic.

1.2.1 The Emergence of the LPWAN Paradigm

To emphasise the similarities and highlight the key differences across LPWAN technologies, a description and comparison of the technologies which fit within the LPWAN paradigm will first be provided. Note that for LoRaWAN a relatively brief description is provided in Appendix A, with a more thorough examination forming Appendix B.

The material in Appendix A was published in the following paper:

- **Paper 1. Joseph Finnegan**, and Stephen Brown, "A Comparative Survey of LPWA Networking", presented at Zhejiang University International Doctoral Students Conference, May 2017. Submitted to arXiv as: "A Comparative Survey of LPWA Networking"

For this paper, the core of the work was undertaken by this researcher.

1.2.2 An Evaluation of LoRaWAN in Terms of Energy Efficiency

This section provides analysis of the LoRaWAN protocol in terms of energy efficiency, and shows the implications of the ALOHA-based MAC on the device lifetime and the potentially available energy harvesting approaches. Paper 2 develops analytical energy models of four LPWAN technologies (LoRaWAN, Sigfox, NB-IoT and EC-GSM-IoT), and compares the technologies in terms of energy consumption for an increasing daily throughput and set environments. The results show a clear distinction between the licensed and unlicensed options, demonstrating the feasibility of multiple coexisting LPWAN options in the future, with particular choice of technology chosen based on the application requirements. Paper 3 demonstrates the feasibility of Radio Frequency (RF) energy

harvesting to power LoRaWAN applications, through the combination of a LoRaWAN analytical model and a model of an RF energy harvester, and data from RF harvesting campaigns from around the world.

- **Paper 2. Joseph Finnegan**, and Stephen Brown, "An Analysis of the Energy Consumption of LPWA-based IoT Devices", published in proceedings of *IEEE International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1-6, Jun. 2018.
- **Paper 3. Joseph Finnegan**, Kyriaki Niotaki, and Stephen Brown, "Exploring the Boundaries of Ambient RF Energy Harvesting with LoRaWAN", published in *IEEE Internet of Things*, PP:1-8, *Impact factor: 9.936*, 2020.

This researcher was the first author on each of the listed publications. For the first paper, the core of the work was undertaken by this researcher. The second paper was undertaken in collaboration with a researcher in the field of RF energy harvesting, with the author of this thesis's contributions primarily focused on the LoRaWAN protocol modelling and the overall analysis of the combined models.

1.2.3 Large Scale Evaluation of LoRaWAN in ns-3

This section focuses on an evaluation of the LoRaWAN protocol in ns-3. Paper 4 utilises the LoRaWAN energy model from the previous research and hardware measurements to implement an energy module of the LoRa SX1272 transceiver in ns-3, enabling the evaluation of LoRaWAN in simulation. The accuracy of the module was demonstrated through comparison to the LoRaWAN analytical model. In Paper 5, the Class B of LoRaWAN was implemented in ns-3, enabling the analysis of this class of LoRaWAN in simulation. Results and analysis demonstrate the limits of the gateway in handling downlink traffic in scaled networks. In Paper 6, the Adaptive Data Rate (ADR) of LoRaWAN was implemented in ns-3, providing new insights into the behaviour of the algorithm in the allocation of data rates to individual devices. Based on the results, enhancements to the algorithm are proposed which reduce the convergence time of devices to reach the optimal state, and increase overall network PDR. Each of these works demonstrates the first time the feature was implemented in ns-3.

- **Paper 4. Joseph Finnegan**, Stephen Brown, and Ronan Farrell, "Modeling the Energy Consumption of LoRaWAN in ns-3 Based on Real World Measurements", published in proceedings of *Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1-4, Nov. 2018.
- **Paper 5. Joseph Finnegan**, Stephen Brown, and Ronan Farrell, "Evaluating the Scalability of LoRaWAN Gateways for Class B Communication in ns-3", published in proceedings of *IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1-6, Jun. 2018.

- **Paper 6. Joseph Finnegan**, Ronan Farrell, and Stephen Brown, "Analysis and Enhancement of the Adaptive Data Rate Scheme of LoRaWAN", published in *IEEE Internet of Things*, 7(8):7171-7180, *Impact factor: 9.936*, Aug. 2020.

For each of these papers, the core of the work was undertaken by this researcher.

1.2.4 Increasing the Scalability of LoRaWAN

ALOHA-based networks are known to be limited in terms of scalability because of the uncoordinated access of individual devices. As the number of devices scales, the probability of collisions greatly increases. However, the basis of LoRaWAN on ALOHA is key as it enables a low energy consumption rate for devices. Additionally, control plane traffic in LoRaWAN networks is limited because of the duty cycle regulations, which also apply to gateway nodes. In Paper 7, a new feature was proposed and implemented which uses features of mMTC and the LoRaWAN protocol to enable the lightweight timeslotting of periodic traffic. The feature was implemented in ns-3 and shows improvements in terms of PDR for large scale LoRaWAN networks.

- **Paper 7. Joseph Finnegan**, Ronan Farrell, and Stephen Brown, "Lightweight Timeslot Scheduling Through Periodicity Detection for Increased Scalability of LoRaWAN", published in proceedings of *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1-8, Aug. 2020.

In Paper 7, the core of the work was undertaken by this researcher.

1.2.5 Other Publications

The following work was also undertaken during the course of this PhD:

- **Paper 8.** Muhammad Shahwaiz Afaqui, **Joseph Finnegan**, Ronan Farrell, Stephen Brown, and Xavier Vilajosana, "Evaluation of the Potential of HARQ Schemes to Contribute to Reliability and Scalability of Wi-Fi", in preparation.
- **Paper 9. Joseph Finnegan**, Muhammad Shahwaiz Afaqui, Ronan Farrell, Stephen Brown, and Xavier Vilajosana, "Analysing the Effect of Diversity Combining in dense LoRaWAN Networks", in preparation.

These works were developed in parallel, as part of an analysis of the applicability of the same core concepts to both Wi-Fi and LoRaWAN. Hybrid Automatic Repeat Request (HARQ) schemes, which are to be included in IEEE 802.11be, increase the reliability of links through the saving and recombination of failed receptions with later retransmissions. In Paper 8, the impact of this proposed feature in next-generation Wi-Fi in dense environments is investigated. In Paper 9, this

same concept is also applied to LoRaWAN through diversity combining, where failed receptions from multiple gateways are directed to the LoRaWAN Network Server to be combined. This increases the reliability of reception in LoRaWAN networks with minimal changes to the LoRaWAN MAC or network structure.

The relevance of Paper 8 is limited to the overall theme of the thesis, so it has not been included. Paper 9 is more relevant, but is also not key to the core structure of the thesis, and as it is currently a work in progress it has not been included. In both of these works, the core of the work was undertaken in full collaboration with a researcher in the field of analysis and development of next-generation Wi-Fi, from the Universitat Oberta de Catalunya (UOC).

1.3 Thesis Outline

This thesis is structured as follows. The two appendices provide a background on LPWAN, LoRa and LoRaWAN, and as such are listed first here.

Appendix A: Background - The Emergence of the LPWAN Paradigm
This appendix fully defines LPWAN as a concept, and provides an overview and comparative analysis of LPWAN protocols.

Appendix B: A Review of LoRa & LoRaWAN
This appendix provides an explanation of LoRa and LoRaWAN. The modulation format of LoRa, and how LoRa achieves long range and orthogonal data rates are demonstrated, as well as the network structure, access method, and network configuration parameters defined by LoRaWAN. Finally, an overview of the development of tools for the simulation of LoRaWAN is provided.

Chapter 2: An Energy Efficiency Evaluation of LoRaWAN
This chapter describes analysis of the protocol in terms of energy efficiency. An understanding of the energy efficiency of the protocol is key in MAC-layer design in order to maintain the low power requirement. Firstly, analytical models of the energy consumption of LoRaWAN, Sigfox, NB-IoT and EC-GSM-IoT are developed and used to compare and contrast the prospective device lifetime for LPWAN devices running equivalent applications. Then, the LoRaWAN model is used in conjunction with a model of an RF energy harvester to evaluate the potential of powering LoRaWAN applications using ambient RF energy.

Chapter 3: Large Scale Evaluation of LoRaWAN in ns-3
This chapter illustrates the limitations of current LoRaWAN in terms of scalability through extensive simulations in ns-3. Firstly, a model of the energy consumption of a LoRa transceiver is implemented in ns-3 to enable the evaluation of large scale LoRaWAN networks in terms of energy efficiency. The transceiver model is verified through comparison to the analytical model. Then, implementations of Class B and the Adaptive Data Rate (ADR) of LoRaWAN are developed, and used to evaluate potential network performance as the network scales in terms of both uplink and downlink traffic.

Chapter 4: Enhancement of the LoRaWAN Adaptive Data Rate:

Inspired by the analysis presented above, this work presents the effect of proposed enhancements to the ADR of LoRaWAN, which increase the potential scalability of LoRaWAN networks. The existing algorithms are modified to increase the adaptability and overall performance of the system, and to remove the possibility of devices transitioning into edge cases which stop the algorithms from functioning as expected.

Chapter 5: Lightweight Timeslot Scheduling Through Periodicity Detection for Increased Scalability of LoRaWAN
As the final part of this work, this chapter presents an enhancement of the protocol that improves scalability. The enhancement relies on the inherent predictability of periodic mMTC traffic and the channel selection method of LoRaWAN to enable the forecast and subsequent delay of expected future traffic collisions. The proposed system is easily integrated into the existing LoRaWAN architecture, and can increase the scalability of LoRaWAN networks without increasing strain on downlink feedback, or decreasing the energy efficiency of devices.

Chapter 6: Conclusions and Future Work.
This chapter concludes the thesis by providing an overview of the contributions of the work, and describes potential future work.

CHAPTER 2

An Investigation into the Energy Efficiency of LoRaWAN

Energy efficiency is a key metric in the evaluation of LPWAN technologies. The primary constraint on LPWAN networks is the low energy requirement of the device – without this constraint, the PHY and MAC layer design of the protocols would completely change. For example, this constraint directly impacts the design choices for LPWANs related to time synchronisation, control plane traffic, data rate optimisation schemes, downlink frames, and more. Thus, an understanding of the energy consumption rate of LPWAN technologies is important in identifying use cases which are likely to be used with particular protocols, which is a considerable factor when predicting the likely characteristics of network traffic. Thus, it is key in developing realistic deployment strategies for particular protocols, and projecting achievable performance as the network scales. Energy consumption analysis is also vital in the consideration of modifications to the protocols – the trade-off between energy efficiency and scalability must be closely considered when increasing the amount of time device remain out of a deep-sleep mode.

This chapter focuses on the "Low Power" aspect of LPWAN, and evaluates LPWAN technologies, and in particular LoRaWAN, in terms of the potential for ultra low power operation. This chapter begins with a comparison of key LPWAN technologies in terms of energy efficiency, which is performed through the use of analytical models based on the MAC behaviour of the technologies. Current consumption values in individual states are derived from device datasheets and protocol specifications. An analytical model of the energy consumption of a LoRaWAN system is developed. Results from this model are compared and contrasted to equivalent output from models of Sigfox, NB-IoT, and EC-GSM-IoT, to show the expected energy discharge rate for applications requiring particular levels of daily throughput, and to show the suitability of different technologies for different applications. The results show a clear distinction between licensed and unlicensed LPWAN, where

non-licensed LPWAN can achieve a lower energy consumption rate, but is limited in throughput in comparison to licensed options.

The energy-constrained nature of the majority of LPWAN applications coupled with the challenge of battery in-the-field replacement for large scale deployments has led to recent interest in the usage of energy harvesting solutions for sustaining battery-less LPWAN operation. The ultra-low operation of LoRaWAN facilitates the usage of a greater variety of different energy harvesting approaches to enable sustainable long term deployments. In this chapter, the LoRaWAN analytical model is then used to evaluate the potential of powering LoRaWAN devices through harvested RF energy. This is achieved through the use of the model in tandem with a model of an RF energy harvester, and data from measurement campaigns from around the world. The feasibility of powering a device is dependent on the throughput requirements of the application, the available harvestable energy (which is higher in urban areas), and the efficiency of the harvester and storage. It is demonstrated that while frequently reporting long-distance LoRaWAN applications are not always feasible through RF energy harvesting, especially in rural areas, many urban locations can support daily, short-distance measurement reporting.

2.1 Research Analysing Energy Efficiency of LoRaWAN Class A

Since the release of the LoRaWAN protocol, there has been a significant amount of work in analysing the energy consumption of LoRaWAN devices, in particular Class A. In [14], a framework for the modelling of the energy consumption of LoRa-based IoT devices is presented. The system-level model accounts for each of the individual factors contributing to the energy expenditure on the node. LoRaWAN is used as the networking protocol for the device, but the primary contribution of the work is the proposed framework, rather than an exploration of the factors which influence energy consumption in the LoRaWAN protocol. This work is extended in [15] to account for the potential use of solar energy harvesting to supply energy to the node.

In [16], the energy consumption of LoRaWAN transmissions are characterised. The current consumption and times taken in each LoRaWAN state are derived from measurements of a LoRa microcontroller. These are used to develop a LoRaWAN analytical model, which is used to develop predictions of the expected battery lifetime of a typical LoRaWAN node sending realistic amounts of traffic, factoring in just the energy consumption due to the networking. In [17] and [18], an energy consumption model of a full LoRa-enabled sensor node is developed, including the factors of processing and sensor. Numerical results show the effect of LoRa parameters such as the spreading factor, bandwidth and coding rate on the energy efficiency. In [19], hardware experiments are performed to capture the energy consumption profile of a representative LoRaWAN device under a variety of conditions, and used to generate predictions of device lifetime for a set of LoRaWAN devices under different configurations. In [20], a model of LoRa energy consumption is developed and used to develop a device lifetime prediction model based on Markov chains. In [21], the power consumption of LoRaWAN Class A and Class C are compared, and the effect of the rate of the

transmission on the predicted device lifetime is demonstrated. In [22], the energy consumption of LoRa in an industrial environment is studied, both in the cases where energy harvesting approaches are available, and where they are not.

Comparative analysis of different LPWAN protocols is also available. In [23], the energy consumption of LoRa and Sigfox are compared. The results show that while Sigfox is suitable for seldomly transmitting devices with small payloads, the flexibility of LoRaWAN in terms of packet size and available data rates makes it an attractive option when targeting a wider variety of use cases. In [24], the energy consumption of LoRaWAN, DASH7, Sigfox, and NB-IoT are compared. It is shown that LoRaWAN and DASH7 are more energy efficient than Sigfox and NB-IoT.

Analysis of non-LoRaWAN LoRa-based communications has also been studied. In [25], the energy consumption of LoRa transmissions is studied. However, the MAC protocol assumed is not LoRaWAN, but instead a centralized-synchronous protocol designed for mission-critical IoT applications. [26] also analyses the energy efficiency of LoRa communications over a non-LoRaWAN MAC layer. In their work, a clustering MAC protocol suitable for a linear wireless sensor network (used in, for example, pipeline monitoring) is assumed. A LoRa-based system is compared against an equivalent Zigbee-based system and is shown to be more energy efficient in this form of application.

Use of energy consumption analysis in MAC layer behaviour has also been presented. In [27], a resource allocation solution for LoRa is proposed which optimises fairness of energy consumption. This is achieved by framing the problem as an optimisation problem and developing a greedy algorithm to achieve max-min fairness considering energy consumption and transmission reliability. Simulated results from ns-3 show that the algorithm can improve the energy fairness of the network in comparison to legacy LoRaWAN.

There has been some research into the feasibility of harvesting various ambient energy sources for powering LoRaWAN devices: thermoelectric was considered in [28], photovoltaic in [29], and electromagnetic vibration in [30]. Other works consider a combination of sources [31]. Wireless power transmission (WPT), where dedicated radio frequency (RF) power is transmitted, has also been considered as a potential solution in [32, 33]. In [34], the general operation of a batteryless, energy harvesting LoRaWAN device in the context of relatively high harvested power levels is investigated.

Overall, from analysis of the protocol it can be concluded that the energy consumption of a LoRaWAN module is dependant on six primary factors: the packet size, the transmission rate of uplink and downlink frames, the data rate used by the device, the transmission power of the transceiver, and the device class. Direct modelling of the energy consumption of a Class A LoRaWAN device has been performed, however, more detailed modelling of the energy consumption of other LoRaWAN features is not yet available. Additionally, research into the powering of LoRaWAN devices using energy harvesting is in the early stages. As the energy consumption of LoRaWAN devices is significantly lower than conventional Wi-Fi or LTE-connected devices, there is a greater availability of applicable energy harvesting approaches.

2.2 Comparison of LPWAN Technologies in Terms of Energy Efficiency

This section provides a comparison of LPWAN technologies in terms of energy efficiency. Quantitative analysis of network device efficiency vs. daily throughput enables identification of the changeover point between optimal solutions. The LoRaWAN model is initially studied alone, to demonstrate the different energy consumption rates of LoRaWAN data rates. Then, a set of deployment cases are used in the contrast of different LPWAN technologies, to provide a thorough comparison in terms of predicted device lifetime for a range of daily throughputs.

The approach is to calculate the effect changes in daily throughput has on energy consumption at defined data rates. The particular case considered is a static, wireless, battery-powered device that regularly reports data gathered from the environment to a gateway a number of kilometres away. This fits the requirements of many LPWAN use cases, such as smart water meters, agricultural data gathering, or environmental monitoring. In particular, for each technology the conditions set are that the device has minimal downlink traffic, adheres to European regulations, and that transmissions are periodic. Immediate transition between states is assumed, and a battery annual self-discharge rate of 1% is applied. For each technology, the device lifetime is calculated against increasing daily data throughputs for the *(a)* best case, *(b)* worst case, and *(c)* a defined comparative case. Using this approach, the effect changes in daily throughput requirements has on energy efficiency is shown for each LPWAN technology. As a precursor, a discussion of the relationships between the described units is provided in Appendix C.

The primary contributions are the integration of different energy models that have not been previously compared into a common framework, and the identification of the energy-efficiency crossover points between these models. As well as providing a quantitative differentiation of the different LPWAN technologies, the results also enable the selection of the most efficient wireless solution for specific Internet of Things applications, which is a key factor in optimising device lifetime. The findings indicate that there is a strong differentiation between the technologies that operate in the unlicensed bands and the Cellular-IoT (C-IoT) options.

2.2.1 LoRaWAN

As discussed in Appendix B, the energy consumption of a LoRaWAN module is dependant on six primary factors: the packet size, the transmission rate of uplink and downlink frames, the data rate used by the device, the transmission power of the transceiver, and the device class. LoRaWAN defines set data rates; use of a slower data rate results in a larger range but also a longer transmission time and less energy efficient transmission, and the ADR adjusts the data rate based on recent traffic. In the model it is assumed the device has settled to a particular data rate, and thus this is taken as an input parameter. Use of Class A is modelled as this class defines a MAC layer that is the most suited to applications with energy constraints and minimal downlink requirements. As a reminder, the format of a Class A transmission is shown in Figure 2.1.

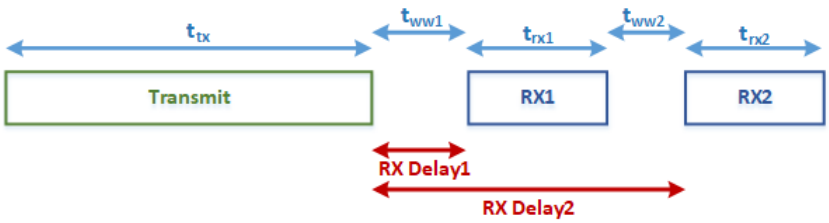


Figure 2.1: Format of a LoRaWAN Class A Transmission

The Network Server may decide to send a downlink frame for a number of reasons:

- The previous uplink frame requested the transmission success to be confirmed by the immediate transmission of a downlink frame.
- There is MAC layer data to be sent to the device.
- There is application layer data to be sent to the device.

It should be made clear that the device cannot reliably predict when downlink frames will be sent, in particular because of the duty cycle limits imposed on the gateway. As a result the opening of the RX1 and RX2 receive windows follow every single uplink frame. If a downlink frame is received in RX1, the device does not open RX2 and can go straight into sleep mode. Thus in modelling the energy consumption of a Class A frame there are three different situations which must be considered:

1. An unresponded frame
2. A frame responded to in RX1, using a data rate that is a function of the uplink data rate (usually the same as the uplink data rate)
3. A frame responded to in RX2, using a set data rate (usually DR0)

2.2.1.1 LoRaWAN Transmission Time

Before analysing the behaviour in each of the identified situations, the transmission time of LoRaWAN frames should be discussed. The transmission time of LoRaWAN frames is described in [35]. The Time-on-Air of a LoRaWAN packet can be calculated by summing the transmission times of the preamble and the payload [35]:

$$t_{packet} = t_{pre} + t_{payload} \quad [\text{s}] \quad (2.1)$$

LoRaWAN packets contain a preamble which by default includes 8 unshifted symbols (N_{pre}). The transmission time of the preamble is directly dependent on the bandwidth (BW) and spreading factor (SF):

$$t_{pre} = (N_{pre} + 4.25) * (2^{SF})/BW \quad \text{[s]} \quad (2.2)$$

As is the transmission time of the payload:

$$t_{payload} = p_{sym} * (2^{SF})/BW \quad \text{[s]} \quad (2.3)$$

The number of symbols is based on the length of the payload in bytes (PL), the spreading factor, and the code rate:

$$p_{sym} = 8 + max(ceil(\frac{8PL - 4SF + 44 - 20H}{4 * (SF - 2DE)}) * (CR + 4), 0) \quad (2.4)$$

where H indicates the existence of a physical header (0 when the header is enabled), and DE the use of low data rate optimisation, which is present only in transmissions using DR0 and DR1. Note that LoRaWAN mandates the use of an explicit header i.e. H must be 0. CR represents the code rate, and is 1 for 4/5, 2 for 4/6, etc. The LoRaWAN MAC header is 13 bytes long for uplink frames, and must be included in the payload in the calculations above (i.e. add 13 to PL).

Note that for short LoRaWAN packets the time taken to receive the preamble is not an insignificant amount of the total reception time. For example, in the reception of a packet containing 10 bytes of application layer data, the amount of time required just to receive the preamble is 20-27% of the total reception time. The length of time for the LoRaWAN preambles is shown in Table 2.1. The relative size of the preamble to the full transmission time does not follow a fully linear pattern; the faster data rates encode fewer bits per symbol and so there is a lower maximum amount of "wasted" bits in the final symbol of the transmission.

Table 2.1: Transmission Time for a LoRaWAN Preamble for each Data Rate, and the Relative Amount of Time used to Transmit the Preamble for a 10 Byte Payload

Data Rate	**Preamble Time**	**Total Transmission Time**	**% for Preamble**
DR0	0.401408s	1.482752s	27.07%
DR1	0.200704s	0.823296s	24.38%
DR2	0.100352s	0.370688s	27.07%
DR3	0.050176s	0.205824s	24.38%
DR4	0.025088s	0.113152s	22.17%
DR5	0.012544s	0.061696s	20.33%

In LoRa transceivers a Channel Activity Detection (CAD) process is available as an alternative to reading the medium for a full preamble. This process can be more energy efficient for frame detection, in particular for slower data rates because the length of time required for the detection process does not double with each slower data rate (as it does for the regular preamble read). However, as the use of CAD is not stipulated in the LoRaWAN specification, use of it is not modelled here. A receiver does not have to listen to the entire length of a preamble to detect the presence of a frame; in the

model it is assumed that the device listens for the length of time of the preamble that consists of the unmodulated upchirps.

2.2.1.2 Unresponded Frame

From Figure 2.1, it can be seen that the behaviour by state for an unresponded LoRaWAN Class A transmission consists of transmitting the uplink frame (*tx*), transferring to standby mode until the beginning of the first receive window (*ww1*), and then to receive mode to attempt to read a preamble (*rx1Pre*). Once no preamble is received, the device transfers back to standby mode until the beginning of the second receive window (*ww2*), where it once again wakes to receive a preamble (*rx2Pre*). In this case, again no preamble is received and the device returns to sleep mode. Since downlink traffic in LoRaWAN is limited, this is by far the most common behaviour for a Class A transmission. Thus, the total time for this form of transmission is:

$$t_{UnresA} = t_{tx} + t_{ww1} + t_{rx1Pre} + t_{ww2} + t_{rx2Pre} \quad [\text{s}] \quad (2.5)$$

and thus the total energy consumption is directly dependent on the current consumption and length of time in each state:

$$E_{UnresA} = (t_{tx} * I_{tx} + t_{ww1} * I_{std} + t_{rx1Pre} * I_{rx} + t_{ww2} * I_{std} + t_{rx2Pre} * I_{rx}) * Vcc \quad [\text{J}] \quad (2.6)$$

2.2.1.3 Responded Frame in RX1

When a transmission is responded to in RX1, the device transmits the frame, transfers to standby mode for the time in-between the end of the transmission and the first receive window, and then wakes to receive the incoming frame (*rx1Recv*). The total time is thus:

$$t_{ResA\_RX1} = t_{tx} + t_{ww1} + t_{rx1Recv} \quad [\text{s}] \quad (2.7)$$

and the total energy consumption is:

$$E_{ResA\_RX1} = (t_{tx} * I_{tx} + t_{ww1} * I_{std} + t_{rx1Recv} * I_{rx}) * Vcc \quad [\text{J}] \quad (2.8)$$

2.2.1.4 Responded Frame in RX2

When a transmission is responded to in RX2, the device transmits the frame, transfers to standby mode for the time in-between the end of the transmission and the first receive window, wakes long enough to receive a preamble (and receives none), then transfers back to standby mode until the start of the second receive window, and finally wakes back up to receive the incoming frame (*rx2Recv*). The total time is thus:

$$t_{ResA\_RX2} = t_{tx} + t_{ww1} + t_{rx1Pre} + t_{ww2} + t_{rx2Recv} \quad \text{[s]} \quad (2.9)$$

and the total energy consumption is:

$$E_{ResA\_RX2} = (t_{tx} * I_{tx} + t_{ww1} * I_{std} + t_{rx1Pre} * I_{rx} + t_{ww2} * I_{std} + t_{rx2Recv} * I_{rx}) * Vcc \quad \text{[J]} \quad (2.10)$$

Note that the value of t_{ww2} is dependent on the data rate used in the uplink transmission. For clarity, the sources of each of the t_x values above are presented in Table 2.2.

Table 2.2: Time-related Variables

Variable	**Dependent on**
t_{tx}	Uplink data rate and packet length
t_{ww1}	= $RXDelay1$ (default = 1 s)
t_{rx1Pre}	Uplink data rate and $RX1DROffset$
t_{ww2}	= $RXDelay2$ - $RXDelay1$ - t_{rx1Pre}
t_{rx2Pre}	a pre-set data rate (default = DR0)
$t_{rx1Recv}$	Uplink data rate, $RX1DROffset$ and downlink packet length
$t_{rx2Recv}$	a pre-set data rate (default = DR0) and downlink packet length

2.2.1.5 Proportional Usage of RX1 and RX2

In current Network Server implementations RX1 is used as a priority for downlink responses, with RX2 only being used when the gateway cannot transmit during RX1 because of duty cycle regulations. Thus, the comparative use of RX1 and RX2 is dependent on the load in the network, and for smaller networks with minimal downlink responses, RX1 will be predominantly used. In the model, an RX2 usage rate of 5% is applied, corresponding to a relatively small LoRaWAN network. This leads to an average energy consumption in a responded LoRaWAN frame of:

$$E_{ResA} = \frac{E_{ResA\_RX1} * 19 + E_{ResA\_RX2}}{20} \quad \text{[J]} \quad (2.11)$$

2.2.1.6 Expected Downlink Traffic

In the model, the sending of minimal downlink traffic is assumed i.e. downlink frames are sent as a result of the ADR scheme only. Use of the ADR requires the device to request the transmission of a downlink frame after the transmission of ADR_ACK_LIMIT uplink frames without a response. This frame may be transmitted in either of the receive windows of one of the next ADR_ACK_DELAY uplink frames from that device. Assuming the worst-case scenario in terms of energy efficiency, this results in a device receiving a downlink frame every ADR_ACK_LIMIT uplink frames. ADR_ACK_LIMIT and ADR_ACK_DELAY by default are both 32.

Thus, the average energy consumption of a LoRaWAN transmission for a particular device can be calculated as:

$$E_{packet} = \frac{(ADR\_ACK\_LIMIT - 1) * E_{UnresA} + E_{ResA}}{ADR\_ACK\_LIMIT} \quad [\text{J}] \quad (2.12)$$

and as LoRaWAN Class A is an ALOHA-based protocol, the power consumption for the networking side of the system can be defined purely based on the energy consumption for an individual packet and the transmission periodicity t:

$$P_{net}(t, DR) = \frac{E_{packet}(DR)}{t} \quad [\text{W}] \quad (2.13)$$

As the case modelled here is a simple reporting application without a sensor, the processing load is very low and the remaining energy consumption of the system is dependent on the remaining time per transmission period when not in a transmit-and-receive cycle, where the device is in the lowest power state:

$$E_{sleep}(t, DR) = I_{sleep} * (t - t_{packet}(DR)) * Vcc \quad [\text{J}] \quad (2.14)$$

And therefore, the total amount of energy consumed by the system in a time period t, in which the device performs one transmit-and-receive cycle and spends the remainder of the time in sleep mode, is:

$$E_{system}(t) = E_{packet}(DR) + E_{sleep}(t, DR) \quad [\text{J}] \quad (2.15)$$

Finally, the device reference parameters are outlined in Table 2.3. These parameters are derived from the datasheet of the BSFrance LoRa32u4II board (SX1276 LoRa transceiver, ATmega32U4 CPU).

Table 2.3: LoRaWAN Average Current Consumption and Operating Voltage

Mode	**Current**
Sleep (I_{sleep})	$15\mu A$
Transmission (I_{tx})	95.5 mA @ 17 dBm
Transmission (I_{tx})	35.5 mA @ 14 dBm
Transmission (I_{tx})	22.0 mA @ 0 dBm
Reception (I_{rx})	17 mA
Standby (I_{std})	6.6 mA
Operating Voltage (Vcc)	3.3 V

2.2.2 Sigfox

As described in Appendix A, the channel access method of Sigfox is RFDMA with no channel pre-transmission sensing; the device randomly transmits on three of 360 available 100Hz channels. The base station scans the spectrum listening at every channel and uses signal processing algorithms to retrieve the message. The amount of 12 byte-payload uplink frame that can be sent by a device is limited by the subscription tier; the maximum amount for the highest tier is 140 packets per day. The usage of a simple ALOHA-based access scheme in Sigfox enables the use of a similar, but simplified version of the LoRaWAN energy model for Sigfox:

$$E_{day} = E_{report} * Reports_{day} + E_{sleep} \quad [\text{J}] \quad (2.16)$$

Subsequent to the original publication of this work, new more detailed energy consumption models of the Sigfox protocol have been published [36]. Since the primary goal of this work is an analysis and comparison to the LoRaWAN protocol, the model in the original work has been replaced by an adapted version of the newer model introduced in [36].

In a regular unidirectional Sigfox transmission, the device first wakes, then begins a transmit-and-wait cycle of the same frame on three different random channels, before reverting to sleep mode. The Sigfox protocol flow is shown in Figure 2.2.

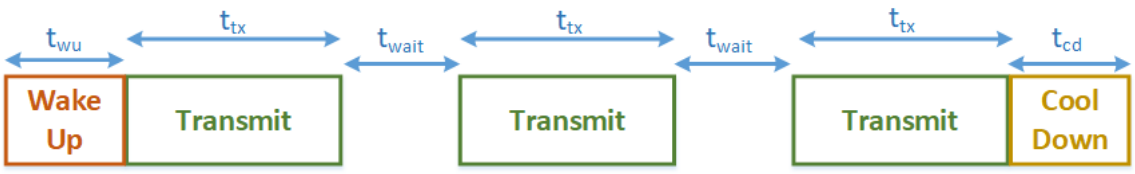


Figure 2.2: Sigfox Protocol Flow

In a unidirectional Sigfox transmission, there is no opportunity for downlink feedback from the network, and so there are no downlink receive windows. Therefore the energy consumption for a unidirectional Sigfox transmission can be defined as:

$$E_{report} = (t_{wu} * I_{wu} + 3 * t_{tx} * I_{tx} + 2 * t_{wait} * I_{wait} + t_{cd} * I_{cd}) * Vcc \quad (2.17)$$

with the device spending the remainder of the time in sleep mode. The current usage and time spent in each state and supply voltage are taken from the values presented in [36], which analysed a MKRFOX1200, and are detailed in Table 2.4.

Table 2.4: MKRFOX1200 Current Consumption and Times in each State

Mode	**Current**	**Time in State**
Sleep (I_{sleep})	16 μA	
Wait (I_{wait})	1.2 mA	486 ms
Cool-down (I_{cd})	1.2 mA	510 ms
Wake-up (I_{wu})	10.4 mA	287 ms
Transmission (I_{tx})	27.2 mA @ 14.5 dBm	2080 ms
Operating Voltage (Vcc)	3.0 V	

2.2.3 NB-IoT

The energy model for NB-IoT is based on the battery lifetime calculations defined in [37] and [38]. The cases defined in the documents were recalculated based on the protocol definition and arrived at battery lifetimes comparable to those found by the authors (within 5%). The calculations are then modified to model increasing daily throughputs for the worst, best, and a defined comparative case. Current consumption estimates for a typical device in different modes, also derived from NB-IoT, and are outlined in Table 2.5.

Table 2.5: NB-IoT Current Consumption in Different Modes

Mode	**Current Consumption**
Sleep	15 μA
Standby	1 mA
Transmit (+23 dBm)	
- Integrated PA	151.51 mA
- External PA	139.39 mA
Receive	
- Synchronization (PSCH)	24.24 mA
- Normal (PBCH, PDCCH, PDSCH)	21.21 mA

Detailed analytical modelling of NB-IoT is not a focus of this thesis; this section simply summaries the energy model presented in [37]. Since the original publication of this work, more detailed modelling of the energy consumption of an NB-IoT device has been performed, for example in [39, 40, 41, 42]. The structure of an example NB-IoT transmission is shown in Figure 2.3.

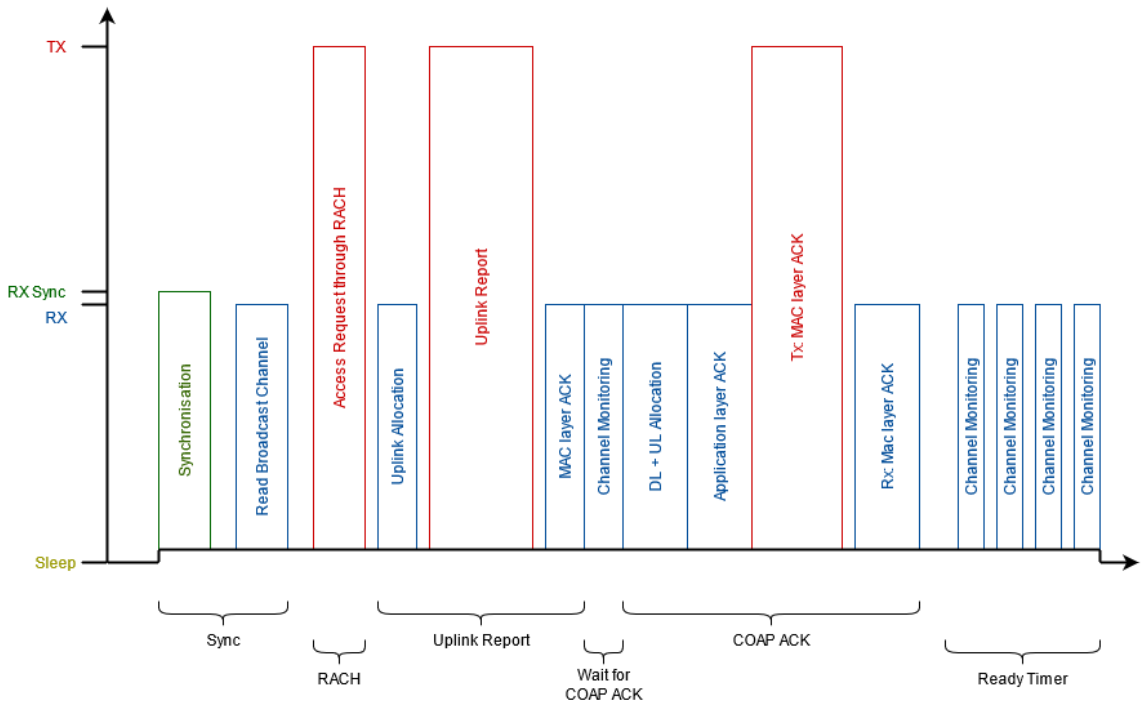


Figure 2.3: NB-IoT Protocol Flow

The format of an NB-IoT transaction is as follows:

- Firstly, the device reads the synchronisation channel, which allows the device to synchronize to the time slot boundaries of the cell.
- Then, reading the physical broadcast channel allows the device to receive system information about the cell.
- The device then requests access through a transmission in the Random Access Channel.
- When transmission is granted, the device transmits the uplink report, and receives an acknowledgement. The uplink report stage can be repeated if required using retransmissions.
- The application layer acknowledgement is then transmitted in a similar but reversed manner, with a downlink channel allocated, and the application-layer acknowledgement received.
- After a successful transaction, the device enters a period of discontinuous reception, and after a set amount of time reverts to deep sleep mode.

Therefore for a single NB-IoT transaction the total amount of energy consumed in TX mode can be calculated by summing the energy consumption for each of the transmitting states, which are the random access request, the uplink data transmission, and the acknowledgement for the application layer acknowledgement:

$$E_{tx} = (t_{access} + t_{uplink\_tx} * N + t_{CoAP\_ACK}) * I_{tx} * Vcc \quad [\text{J}] \quad (2.18)$$

where N in the number of retransmissions required to send the report. The energy consumption in receive mode can be calculated in a similar fashion:

$$E_{rx} = (t_{allocate} + t_{uplink\_ACK} * N + t_{CoAP\_rx}) * I_{rx} * Vcc \quad [\text{J}] \quad (2.19)$$

and the energy consumption for synchronisation, which includes an active receive period and a waiting period in standby mode, can be calculated as:

$$E_{sync} = (t_{rx\_sync} * I_{rx\_sync} + t_{std} * I_{std}) * Vcc \quad [\text{J}] \quad (2.20)$$

Note that synchronisation requires a higher power receive mode. Finally, the energy consumption for an average NB-IoT report can be calculated as:

$$E_{report} = E_{tx} + E_{rx} + E_{sync} \quad [\text{J}] \quad (2.21)$$

And the remainder of the time is spent in a sleep mode:

$$E_{sleep} = I_{sleep} * t_{sleep} * Vcc \quad [\text{J}] \quad (2.22)$$

The time spent in each state is dependent on the coverage class of the device, and these durations are outlined in Table 2.6. Retransmissions have been factored into the durations below to model a Block Error Ratio (BLER) of 10 %. In addition to the constraints outlined above, it is assumed that the NB-IoT network is deployed standalone.

Table 2.6: NB-IoT Timings for each State

Case	t_{tx} (ms)	t_{rx} (ms)	t_{sync} (ms)	t_{std} (ms)
Synchronisation				
144 dB	-	-	288	0
154 dB	-	-	301	0
164 dB	-	-	475	620
Random Access				
144 dB	40	-	-	-
154 dB	40	-	-	-
164 dB	320	-	-	-
Data Transmission				
144 dB	120	10	-	-
154 dB	960	60	-	-
164 dB	3840	440	-	-
CoAP ACK				
144 dB	10	60	-	-
154 dB	40	160	-	-
164 dB	320	1240	-	-

2.2.4 EC-GSM-IoT

The energy model used in these calculations for EC-GSM-IoT is based on the battery lifetime calculations defined in [37]. The cases defined in the document were recalculated based on the protocol definition and results comparable to the described battery lifetimes were achieved. The calculations were then modified to model increasing daily throughputs for the worst, best, and comparative cases. Detailed analytical modelling of EC-GSM-IoT is not a focus of this thesis; this section simply summaries the energy model presented in [37].

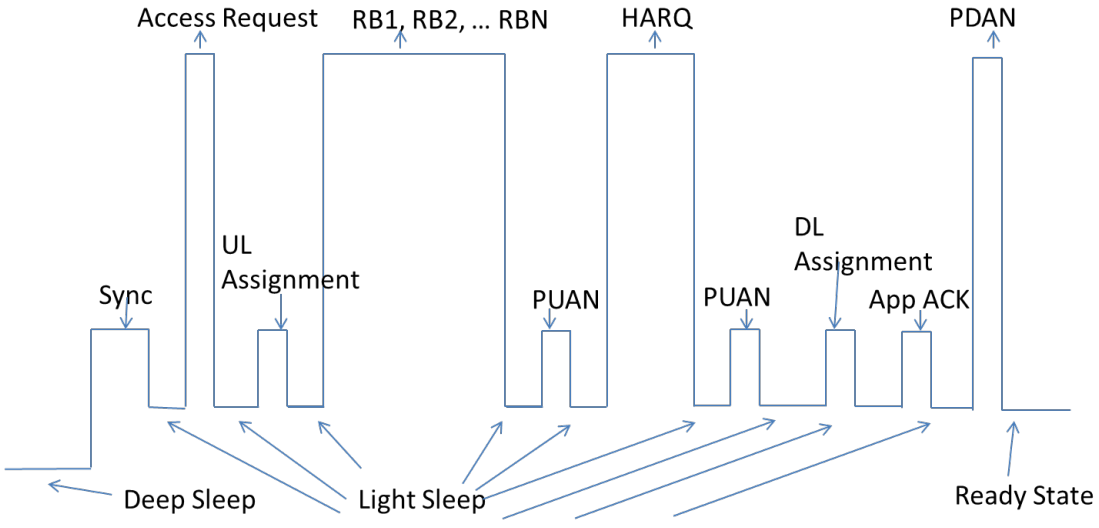


Figure 2.4: EC-GSM-IoT Protocol Flow

Figure 2.4 shows the protocol flow for an example EC-GSM-IoT transaction. The results presented model a BLER of 10%. The timings for each individual state, from [37] are provided in Table 2.7. The current consumption for each state is shown in Table 2.8 at the end of this section.

Table 2.7: EC-GSM-IoT Timings for each State

Case	t_{tx} (ms)	t_{rx} (ms)	t_{ls} (ms)	t_{pll} (ms)
Synchronisation				
GPRS	-	24.811	96.359	-
+10 dB	-	300	0	-
+20 dB	-	577	0	-
Random Access				
GPRS	0.577	-	4.039	0
+10 dB	2.308	-	16.156	12.117
+20 dB	18.464	-	290.808	121.17
Assignment				
GPRS	-	1.154	4.039	4.039
+10 dB	-	4.616	16.156	16.156
+20 dB	-	36.928	613.928	129.248
Data Transmission				
23dBm	68.663	13.848	0	82.511
23dBm +10 dB	667.589	34.043	0	701.632
33dBm +10 dB	69.24	29.427	0	98.667
33dBm +20 dB	716.057	315.619	0	1031.676
Light Sleep				
GPRS	-	580	1270	-
+10 dB	-	580	1270	-
+20 dB	-	340	980	-
Ready State			t_{ds} (ms)	
GPRS	-	2.308	19965.931	8.078
+10 dB	-	9.232	19634.156	56.546
+20 dB	-	73.856	18840.204	508.914

As can be seen in Figure 2.4, the first stage of the transaction is the synchronisation process. This process will take an amount of time that is dependent on the coverage class. The EC-GSM-IoT evaluation in the protocol specification calculates the average amount of time for synchronisation for the reference GPRS case, the +10 dB case, and the +20 dB case. For the reference case the device synchronises quickly and can move to light sleep mode; for the other cases the devices remain active for the entirety of the synchronisation process. Thus the energy consumption for synchronisation can be defined as:

$$E_{sync} = (I_{rx} * t_{rx} + I_{ls} * t_{ls}) * Vcc \quad \text{[J]} \quad (2.23)$$

Similarly, the random access will also require a larger amount of time for a higher coverage class, as a greater amount of access bursts are needed to be sent for each successive coverage class, as the number of blind retransmissions increases. In this state, the PLL remains running during non-random access time slots. Thus, the energy consumption in the random access phase can be calculated as:

$$E_{access} = (I_{tx} * t_{tx} + I_{ls} * t_{ls} + I_{pll} * t_{pll}) * Vcc \quad [\text{J}] \quad (2.24)$$

Next, the device is granted a control channel. Again, the average amount of time taken for this is dependent on the control channel, as different cases will need to listen to more bursts over the access grant control channel before receiving the grant message. The PLL must be kept running in between bursts in the same access grant control channel block; outside the block the device can transition to sleep mode. Overall the energy consumption for assignment can be calculated as:

$$E_{assignment} = (I_{rx} * t_{rx} + I_{pll} * t_{pll} + I_{ls} * t_{ls}) * Vcc \quad [\text{J}] \quad (2.25)$$

Following this, the device transmits the uplink data, which in the ideal case is responded to with an acknowledgement (the PUAN). For devices in higher coverage classes, HARQ retransmissions are used, where incorrectly received data is stored and combined with retransmissions to increase the probability of reception. For simplicity, aggregated payloads of 200 bytes are assumed. In order to transmit an application-layer acknowledgement (which has a payload size of 0 bytes), the device is assigned a downlink control channel, and transmits the acknowledgement and receives an acknowledgement for this over the downlink channel (PDAN). The PLL remains active for all timeslots within a block of blind repetitions that are to be HARQ combined. For simplicity, here it is assumed that the PLL remains active for all timeslots that are not used for transmission or reception. Overall, the energy consumption for this can be calculated as:

$$E_{data} = (I_{tx} * t_{tx} + I_{rx} * t_{rx} + I_{pll} * t_{pll} + I_{ls} * t_{ls}) * Vcc \quad [\text{J}] \quad (2.26)$$

As can be seen in Figure 2.4, between each expected reception and transmission opportunity the device reverts to a light sleep mode in order to save energy. In some of these states the device will infrequently have to monitor the medium. The time spent in receive mode and light sleep mode for an average transaction can be seen in Table 2.7, and the total energy consumption can be calculated as:

$$E_{light\_sleep} = (I_{rx} * t_{rx} + I_{ls} * t_{ls}) * Vcc \quad [\text{J}] \quad (2.27)$$

Finally, the device enters a ready state for the length of its ready timer. During this time the device searches for two DRX opportunities and otherwise spends the rest of the time in deep sleep mode. It is assumed that the duration of the ready state is 20 seconds. At the end of this time, the device enters deep sleep mode (and so will have to resynchronise for the next transaction). The current consumption for this state is thus:

$$E_{ready\_state} = (I_{rx} * t_{rx} + I_{pll} * t_{pll} + I_{ds} * t_{ds}) * Vcc \quad [\text{J}] \quad (2.28)$$

Therefore, the energy consumption for a single report can be calculated as:

$$E_{report} = E_{sync} + E_{access} + E_{assignment} + E_{data} + E_{light\_sleep} + E_{ready\_state} \quad [J] \quad (2.29)$$

The device, when not in any of the other states, remains in a deep sleep mode. Here, it is assumed that when the device wakes it will remain in the same cell as before and so will know what broadcast control channel to synchronise to. The energy consumption in deep sleep mode can simply be calculated as:

$$E_{deep\_sleep} = I_{deep\_sleep} * t_{deep\_sleep} * Vcc \quad [J] \quad (2.30)$$

Table 2.8: EC-GSM-IoT Average Current Consumption

Mode	**Current Consumption**
Deep Sleep (I_{ds})	$4.5\mu A$
Light Sleep (I_{ls})	1 mA
PLL (I_{pll})	30 mA
Transmit (I_{tx})	
- 33dBm	1.227431 A
- 23dBm	152.543 mA
Receive (I_{rx})	30 mA

2.2.5 Analysis of LoRaWAN

This section provides analysis of the tunable LoRaWAN parameters and the effect on the energy consumption.

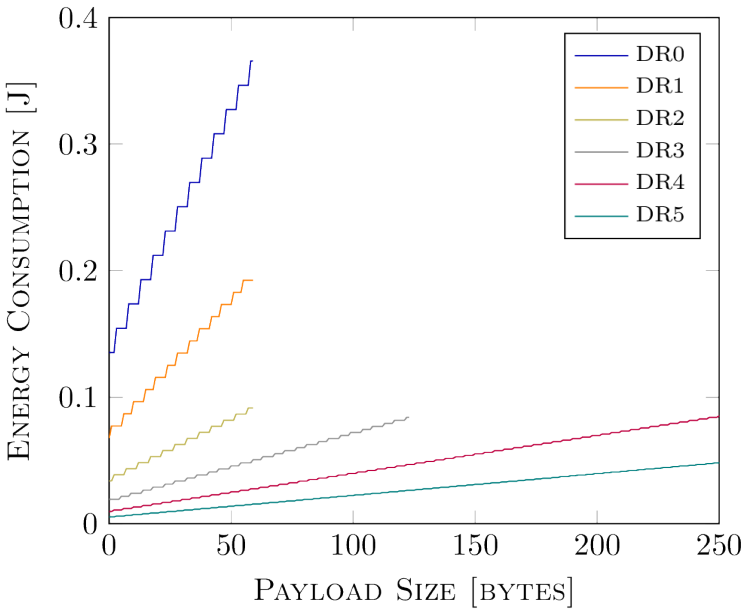


Figure 2.5: Payload Size vs Energy Consumption (Transmission Only)

Figure 2.5 shows the relationship between the payload size and energy consumption of a packet, for each data rate, up to the maximum payload size. Since the voltage and transmission power are constant, the energy consumption increases at a linear rate with time spent transmitting. The energy consumption increases in steps rather than in a purely linear fashion because each LoRaWAN symbol encodes SF bits of data. For example, in DR0, spreading factor 12 is used, and so the increase in energy consumption occurs every 12 bits, as a new symbol is required.

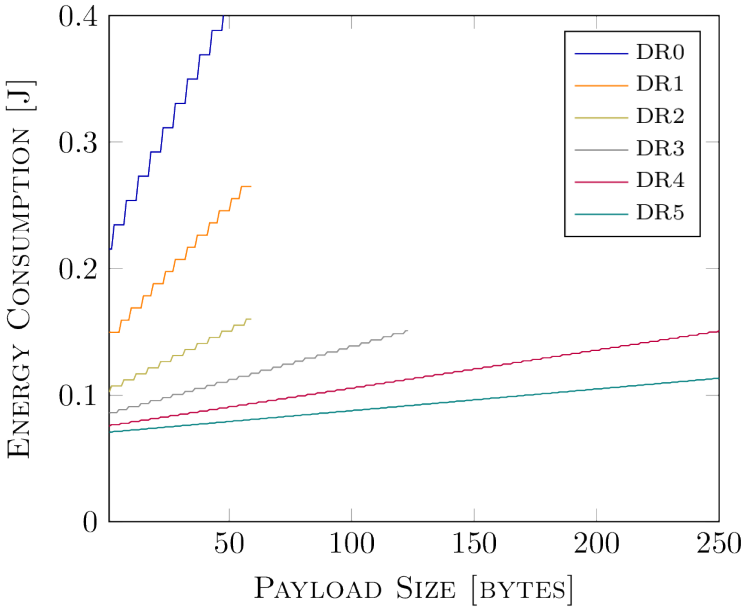


Figure 2.6: Payload Size vs Energy Consumption (full Class A transaction)

Figure 2.6 also shows the relationship between the payload size vs energy consumption, but includes an entire unacknowledged Class A transaction. No matter the payload size and data rate, there is a minimum energy consumption to a transmission because of the dual factors of the time spent in standby mode between receive windows, and the time spent to receive the preamble in RX2. Thus from an energy efficiency perspective, it is more efficient to send fewer, larger packets. In the general case this is true because unaggregated application layer payloads for mMTC traffic tend to be small in size and so the existence of the 13-byte header in LoRaWAN can result in a significant proportion of the energy consumption of the packet spent just on the transmission of the headers. However, in the use of faster LoRaWAN data rates this is particularly true because the energy consumption for the standby and receive states exceeds the energy consumption for the transmission itself.

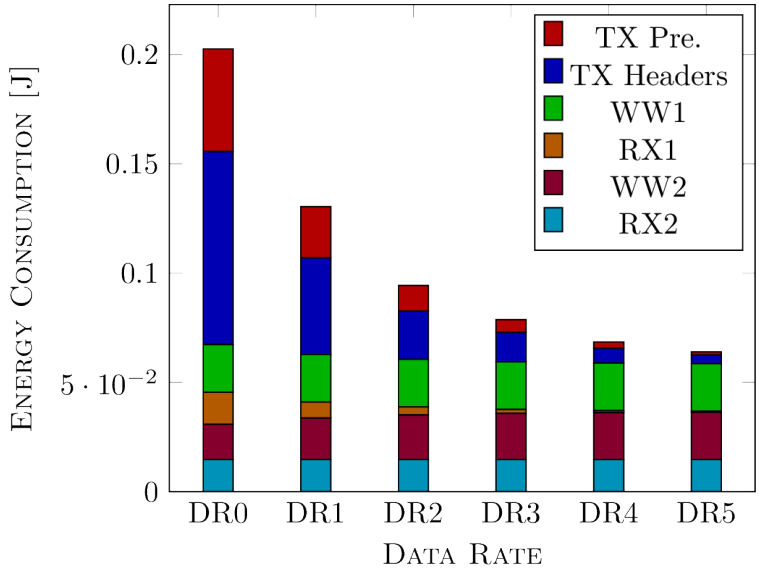


Figure 2.7: Overhead of a LoRaWAN Class A Transmission in Terms of Energy Efficiency

This point is further emphasised in Figure 2.7, which shows a breakdown of the full overhead of an unresponded Class A LoRaWAN packet in terms of energy consumption (assuming transmission at 14 dBm). This includes the impact of all states *excluding* the energy consumption for the transmission of the payload. For faster data rates the overhead from the LoRaWAN header has little impact in comparison to the remainder of the transaction, because of the long wait times between receive windows and the use of DR0 in the RX2 window. For the slower data rates, the impact of the preamble and header increases. The impact of RX2 and WW1 remain constant, as the length of time in WW1 is set by the standard, and the data rate of RX2 is by default DR0 no matter the data rate of the previous uplink transmission. The total time spent in RX1 and WW2 together is also a constant (1 s by default), but the energy consumption impact for the two states is higher for slower data rates as the current consumption in the receive state is higher than in standby.

Overall, even though a LoRaWAN packet sent using DR0 is transmitted at a data rate that is approximately 18 times slower than DR5, the impact of the overhead is only approximately 3 times higher. Of course, as shown in the previous figures, the impact of the payload itself must also be considered.

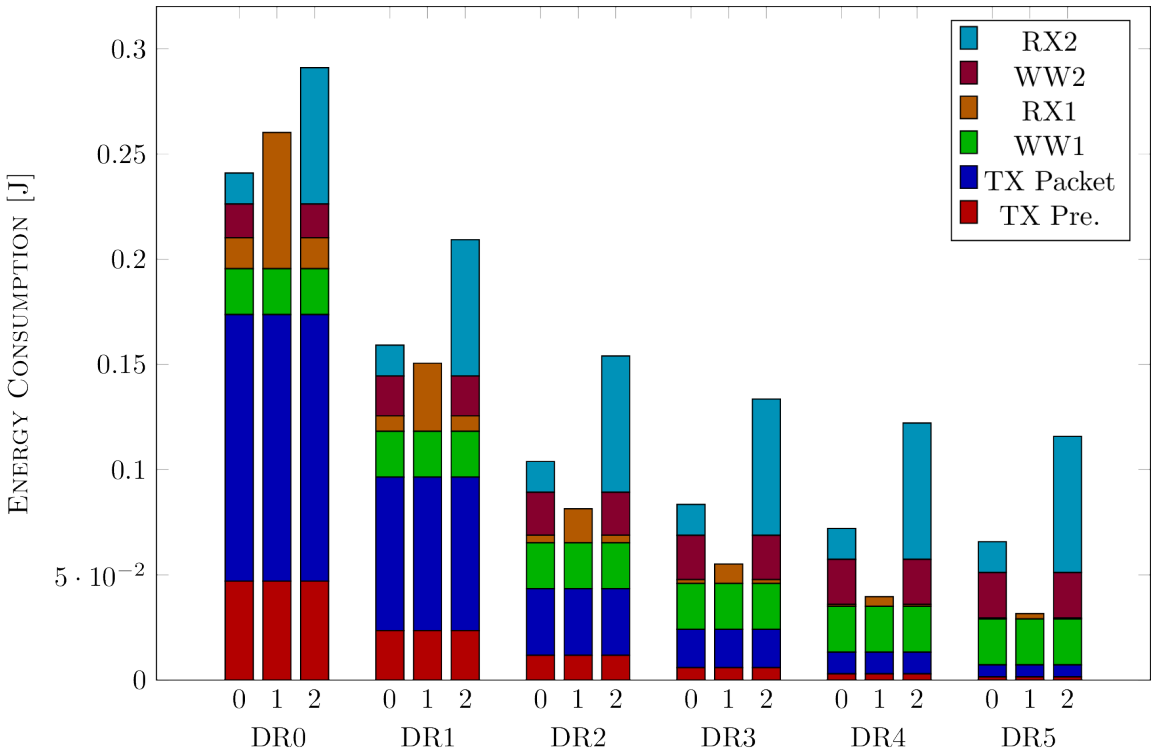


Figure 2.8: Energy Consumption of an Unacknowledged, RX1-acknowledged and RX2-acknowledged LoRaWAN Class A Transmission, for each Data Rate

Figure 2.8 shows the energy consumption of a LoRaWAN transmission for the three LoRaWAN cases: an unacknowledged frame (referred to as "0" in the figure, a frame acknowledged in RX1 ("1" in the figure), and a frame acknowledged in RX2 ("2" in the figure). These results model an example uplink payload sizes of 10 bytes and a minimal downlink packet size with 0 bytes of application layer data, resulting in a packet size of 11 bytes (as LoRaWAN downlink frames do not include a CRC and so the overhead is smaller). Note that the ordering of the bars has been flipped from the previous figure, to demonstrate the effect of the changing in time spent in the latter states for the three different cases. The figure demonstrates that for faster data rates especially, it is actually more energy efficient to receive a downlink frame in RX1 than to not receive any downlink, because of the length of the time in standby and in receive mode for the RX2 preamble.

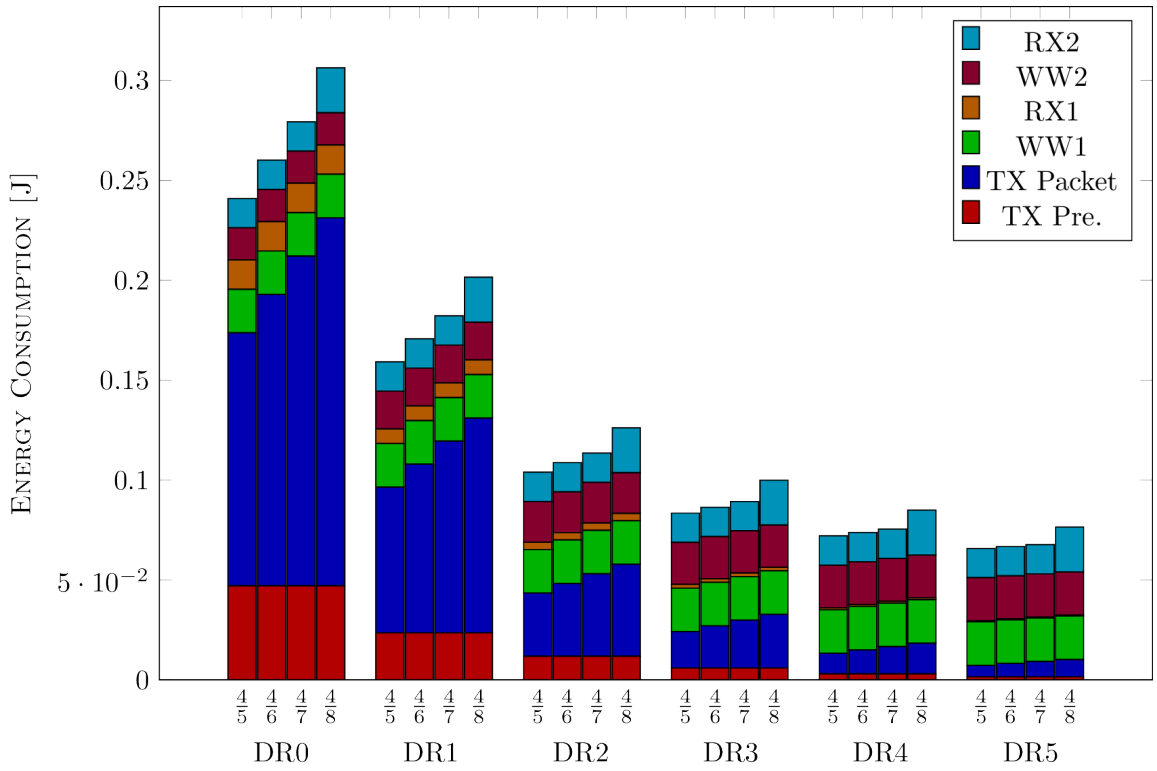


Figure 2.9: Energy Consumption of a LoRaWAN Class A transmission using each of the Configurable Coding Rates, for each Data Rate

Figure 2.9 shows the relationship between the energy consumption of an unacknowledged Class A packet and the different choices for the coding rate configurable parameter. These results model an example uplink payload sizes of 10 bytes. The length of time to send or receive a preamble is dependent on the spreading factor and bandwidth, but not the coding rate. Thus, for an unacknowledged frame, the only varying factor in a change in the coding rate is the length of time of transmission of the packet. Additionally, as the coding rate for the LoRa physical header is always sent at a coding rate of $\frac{4}{8}$, even the effect of this is mitigated. However, the choice of coding rate in LoRa does not have a significant effect on the reception sensitivity, in comparison to the choice of data rate [43].

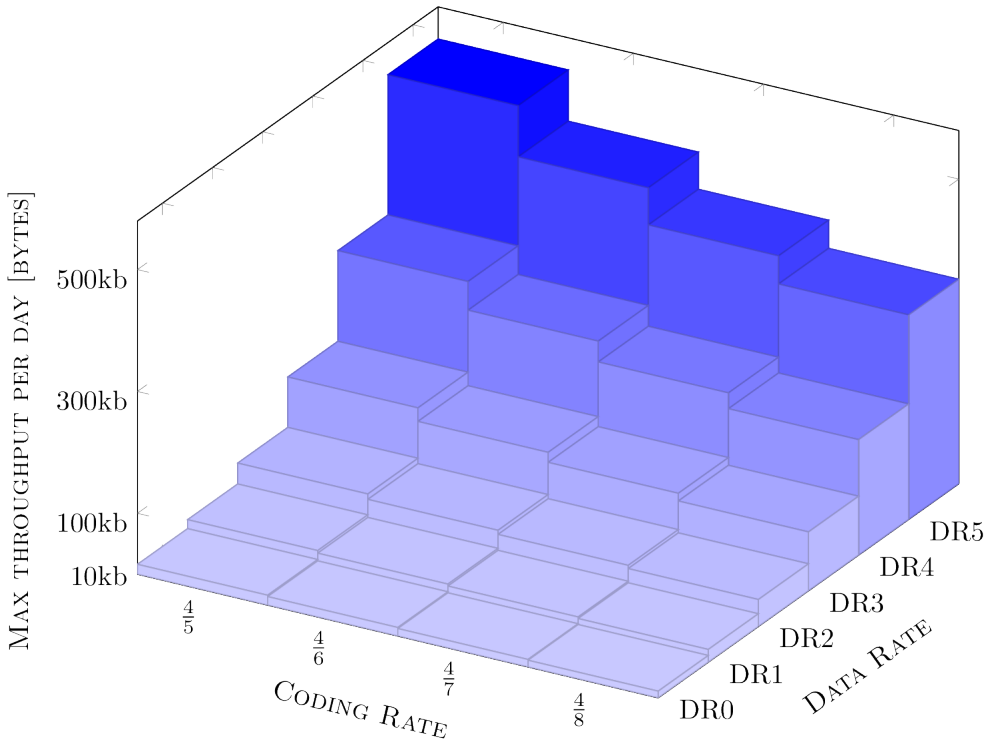


Figure 2.10: Coding and Data Rate vs Maximum Throughput per Day

Finally, Figure 2.10 shows the relationship between the coding and data rate combinations, and the maximum permissible regular throughput allowed per day for a LoRaWAN device while adhering to the access limits of the 868 MHz ISM band. The limits shown in the figure assume access to channels within a single subband with a 1% duty cycle access, which corresponds to a configuration simply using the mandatory LoRaWAN channels. For faster data rates, the amount of permissible traffic is large enough that, for mMTC applications, the duty cycle limits are not a limiting factor. Even for lower data rates, 10 kB per day is typically more than enough for mMTC applications [44, 45]. The challenges for LoRaWAN are typically not for a device to be able to access enough, but instead enabling operation at a low enough power consumption to allow an extended lifetime, and ensuring the gateway can handle the traffic of all nearby devices, as these have to adhere to the same access restrictions.

2.2.6 Comparative Analysis

This section now provides a comparative analysis of the energy efficiency of LoRaWAN, against the other modelled LPWAN technologies. For each technology, the device lifetime is calculated against increasing daily data throughputs for the *(a)* best case, *(b)* worst case, and *(c)* a defined comparative case. The best case is the configuration of each technology which results in the longest possible battery life while still delivering the necessary throughput. Similarly, the worst case is the configuration which results in the shortest possible battery lifetime. A battery capacity of $5Wh$ is modelled; a choice of different capacity will result in shortened/extended curves of the same shape

as those presented (depending on whether the capacity is reduced or increased). As a result of the fundamental differences in each of the technologies, the coverage achieved in the best and worst cases will differ for each technology; the estimated MCL [37] for each is shown in each technology's case table. The comparative case is defined as the necessary configuration of each technology to obtain the closest MCL to 154 dB.

The LoRaWAN defined cases are outlined in Table 2.9. The chosen best and worse cases for LoRaWAN are selected based on the fastest and slowest state which can be reached through use of the ADR algorithm; in the best case, the device transmits using the fastest data rate and with the lowest possible configurable transmission power. In the worse case, the maximum transmission power is used along with the slowest data rate. As there is no significant gain from an increase in the coding rate, and the LoRaWAN ADR does not modify this parameter for a device, the expected coding rate is set to the same for each case. The comparative case has been defined to achieve an MCL of 145 dB; note that the transmission power used in this configuration is only permissible in one of the subbands of the 868 MHz band and so this case is primarily for illustrative purposes.

Table 2.9: LoRaWAN Defined Cases

Case	**Data Rate**	**Code Rate**	**Transmission power**	**MCL**
Best	5	4/5	+0dBm	123 dB
Worst	0	4/5	+14dBm	151 dB
Comparative	0	4/5	+17dBm	154 dB

The Sigfox defined cases are outlined in Table 2.10. In comparison to LoRaWAN, there is a lack of configurability and adaptability in Sigfox transmissions. Sigfox transmissions are sent only at one set data rate. In addition, for the majority of development and evaluation Sigfox boards, the only available transmission power is at 14.5 dBm, which is the highest transmission power allowed for the subband [36]. Thus in Sigfox transmissions there are no configurable parameters, and each of the cases are the same.

Table 2.10: Sigfox Defined Cases

Case	**Power Level**	**MCL**
Best	+14.5 dBm	168 dB
Worst	+14.5 dBm	168 dB
Comparative	+14.5 dBm	168 dB

The NB-IoT defined cases are outlined in Table 2.11, and are based on the defined NB-IoT coverage classes, and whether it is assumed that an external or internal PA is used in the device.

Table 2.11: NB-IoT Defined Cases

Case	**Coverage Class (MCL)**	**PA**
Best	144dB	External
Worst	164dB	Integrated
Comparison	154dB	Integrated

Finally, the EC-GSM-IoT defines cases are outlined in Table 2.12, and are based on the defined EC-GSM-IoT coverage classes, and the transmission power. Note that the +0 dB coverage class is effectively use of legacy GSM.

Table 2.12: EC-GSM-IoT Defined Cases

Case	**TX Power**	**Coverage Class**	**MCL**
Best	23dBm	+0dB	144dB
Worst	33dBm	+20dB	164dB
Comparison	33dBm	+10dB	154dB

These cases and the previously defined models (Equation 2.15 for LoRaWAN, Equation 2.16 for Sigfox, Equations 2.21 and 2.22 for NB-IoT, and Equations 2.29 and 2.30 for EC-GSM-IoT) were then used to generate the following results.

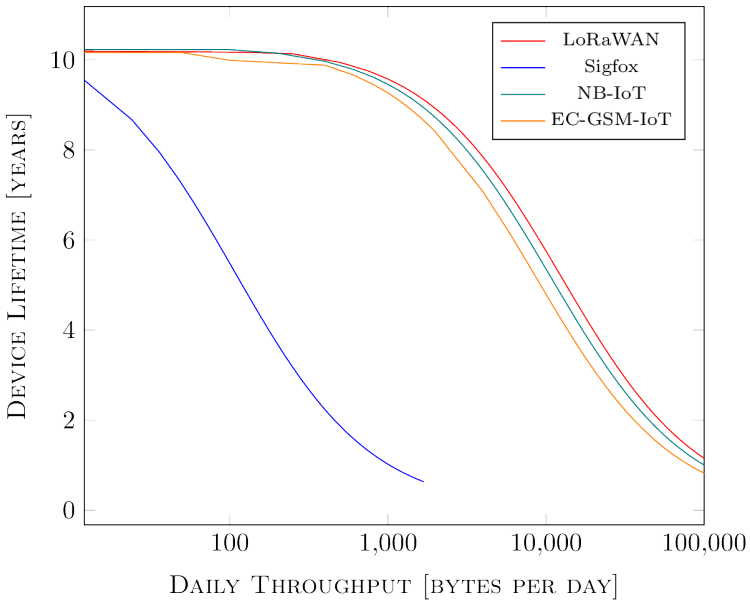


Figure 2.11: Device Lifetime for the *(a)* Best Case

Figure 2.11 shows the device lifetime results for the *(a)* best case. In the best case, LoRaWAN outperforms NB-IoT and EC-GSM-IoT in terms of energy efficiency. On average, the predicted device lifetime for the NB-IoT device is 93% of the predicted lifetime for the LoRaWAN device. For EC-GSM-IoT, the predicted lifetime is 85% relative to LoRaWAN. For very low transmission rates, the energy consumption becomes dominated by the sleep current and battery self-discharge; this is true for every analysed case. Use of the fastest data rate enables over 100,000 bytes to be sent over LoRaWAN without breaking the duty cycle regulations. For Sigfox, the average device lifetime is 50% relative to LoRaWAN, though since the energy discharge curve is much steeper for Sigfox than for LoRaWAN the predicted lifetime relative to LoRaWAN is heavily dependent on the actual daily throughput required.

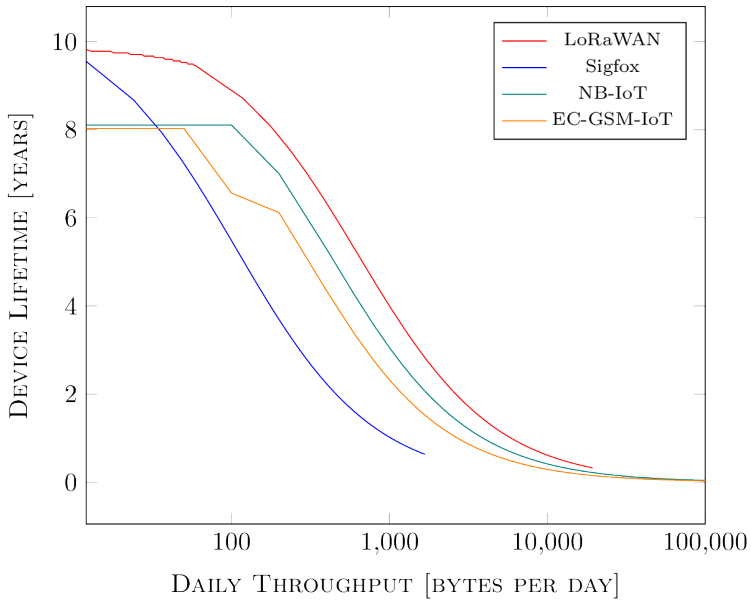


Figure 2.12: Device Lifetime for the *(b)* Worst Case

Figure 2.12 shows the *(b)* worst case. LoRaWAN outperforms every other technology in the worst case. NB-IoT device lifetime is on average 73% relative to LoRaWAN, with EC-GSM-IoT and Sigfox on average 54% and 56% relative to LoRaWAN respectively. However, in this case transmissions become limited for LoRaWAN by the duty cycle regulations. Additionally, it should be noted that in this worst case, LoRaWAN transmissions are being sent at a lower MCL than for the other options.

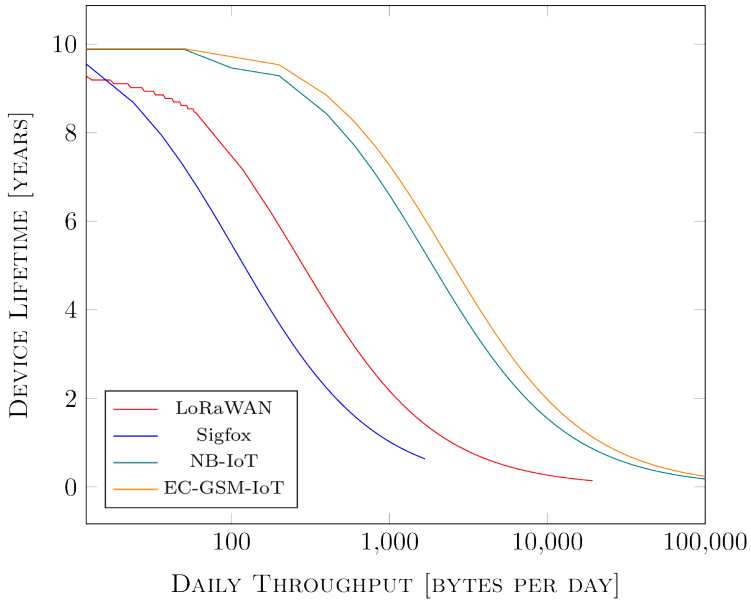


Figure 2.13: Device Lifetime for the *(c)* Comparative Case

Figure 2.13 shows the *(c)* comparative case (equivalent MCL). Interestingly, the C-IoT options

outperform the unlicensed options in terms of energy efficiency for the comparative case. NB-IoT device lifetime is on average 449% relative to LoRaWAN, with EC-GSM-IoT and Sigfox 552% and 70% relative to LoRaWAN, respectively. For the comparative case, the parameters for each wireless option have been configured to achieve the same MCL, as an illustrative example to demonstrate the energy consumption rate of each technology when achieving the same coverage. However, for LoRaWAN this does require the use of a transmission power that is higher than permitted in the majority of subbands, and so this can not be considered a realistic deployment option for a LoRaWAN device.

Sigfox is limited by system regulations, which prevent the device from transmitting more than 1680 bytes per day. The Sigfox device will only last just over one year sending the maximum of 140 packets. For an extended lifetime, the device should only transmit at most 100 bytes per day. For LoRaWAN in every case throughput eventually becomes restricted by ETSI regulations, though this becomes a factor at such a high daily throughput that in reality it is not a significant limiting factor. NB-IoT and EC-GSM-IoT are not limited by duty cycle regulations, but the maximum throughput is still limited by the energy efficiency requirements. There is extra reliability provided by the C-IoT options; Sigfox sends redundant packets, LoRaWAN uses infrequent downlink frames to help ensure a healthy link, but only the C-IoT options can provide a downlink acknowledgement for every frame.

If reliability is a factor, the C-IoT options are the suitable choice, but these come with a typically higher deployment and access cost. The configurability of LoRaWAN makes it an attractive choice for ultra lower power mMTC applications, especially if use of faster data rates can be guaranteed in static deployments. The Sigfox energy consumption is typically higher and the potential throughput is limited, but low access cost to an extensive network of potential gateways make it a suitable choice for mobile applications. For a daily throughput over 10 kB use of LPWAN technologies may be feasible but a larger battery would be required, or the consideration of energy harvesting approaches.

2.2.7 Findings

The unique combination of range and low energy that LPWAN technologies provide give them the potential to be used in a variety of IoT applications. In this section, a number of LPWAN technologies were directly compared in terms of energy efficiency and impact on battery life. The results identify the complexity in selecting the optimum solution for maximum device lifetime. The contribution is a methodology, which enables the quantification of the changeover points between optimal solutions. Three different cases are provided, but it should be noted that devices will often exist in intermediate cases, and external factors may cause a device to essentially move between these cases. From the results provided it can be concluded that there is no overall best solution – the most suitable technology depends on availability, range, noise, and required throughput and reliability. There is a clear distinction between licensed and unlicensed approaches. The results provided enable the identification of the optimal technology for any particular use case. NB-IoT and EC-GSM-IoT can realistically send between 1000 and 10,000 bytes per day while maintaining an extended device lifetime. LoRaWAN and Sigfox are more suitable when the amount of data that must be sent daily

is less than 1000 bytes, or when strict requirements on mobility or energy efficiency are required for the device. However, it should be noted that, especially at very low reporting intervals, the power consumption for the sensing task should also be carefully considered. In addition, LPWAN technologies are not suitable when the required daily throughput exceeds 10,000 bytes. The overall energy consumption for LPWAN devices is very low, and so these devices are particular suited to be powered by energy harvesting approaches, which could enable a continuous device lifetime over the long term.

2.3 Exploring the Boundaries of Ambient RF Energy Harvesting with LoRaWAN

The energy consumption of LoRaWAN applications has been shown to be a major factor in the design and performance of the protocol. Typical LoRaWAN applications deployed outdoors consist of energy-constrained devices powered by batteries. However, battery usage as the main power source for LoRaWAN applications is challenging as battery power requires in-the-field replacement, and the chemicals involved are environmentally harmful, so harvested energy is a useful alternative. To overcome the battery limitations associated with LoRaWAN long scale deployment, there has been a recent interest in using energy harvesting (EH) solutions for sustaining battery-less LoRaWAN operation. As highlighted at the beginning of this chapter, previous research has demonstrated the feasibility of harvesting various ambient energy sources for powering LoRaWAN sensors: thermoelectric, photovoltaic, electromagnetic vibration [28, 29, 30] and a combination of them [46, 31]. Wireless power transmission (WPT), where dedicated RF power is transmitted, has also been considered as a potential solution in [32, 33]. All of these approaches have considered relatively high levels of energy, and address the issue primarily by identifying a single solution that works.

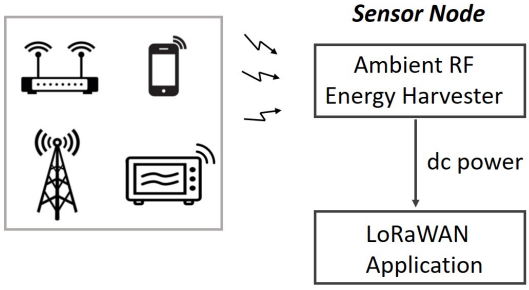


Figure 2.14: Real-life Ambient RF Energy Harvesting Scenario Where Many Existing RF Sources are Present

In contrast to the previous work, the research presented in this section investigates the significantly more difficult issue of exploiting low-level, *ambient* RF power to sustain autonomous operation of LoRaWAN devices. The available harvestable power from RF energy harvesting is typically very low, but, as the previous section has shown, the energy consumption of LoRaWAN devices is also low so this solution is still potentially suitable. The approach taken is to identify the boundaries of

feasibility by exploring the impact of various device and environmental parameters on the feasibility of operation. Ambient RF energy harvesting, as opposed to WPT, refers to the usage of available RF energy from existing sources such as TV broadcasts, radio base stations and cellular phones, as shown in Figure 2.14. The ambient RF energy is collected from the energy harvester which converts it to useful dc energy for LoRaWAN applications. The main advantage of RF EH over WPT is that the power does not have to be purposely provided by the operator of the harvester and thus constitutes a "free" energy source. However, RF EH is considered one of the most challenging energy sources as the available harvestable power varies over frequency, time and location, with the distance from the energy source, and as a result of various environmental conditions. Additionally, the power density available through RF EH is low compared with other renewable energy sources [47]. As highlighted by the recent survey [48], there is currently no other published work which analyses the potential of powering LoRaWAN applications through ambient RF EH. In this work, an analytical model of the energy consumption of a representative device is used, along with analysis of a typical ambient RF EH environment and RF energy data from around the world, to demonstrate the feasibility of using ambient RF energy harvesting to sustain the operation of LoRaWAN devices. A number of limitations related to design and environmental factors are identified.

2.3.1 Extension of the LoRaWAN Analytical Model

The core of the analysis of the energy consumption of LoRaWAN is based on the LoRaWAN analytical model introduced in the previous section. However, in this section the model has been extended to further feature the energy consumption of a representative sensor. According to the system-level approach to modelling the power consumption of an IoT device outlined in [14], the power consumed by a device can be separated into four main parts: power for sensing (P_{sens}), networking (P_{net}), data processing (P_{prc}), and other system tasks (P_{sys}), where the total power is the simply the sum of these constituent parts:

$$P_{system} = P_{sens} + P_{net} + P_{prc} + P_{sys} \quad \text{[W]} \quad (2.31)$$

Considering that the ALOHA-based Class A of LoRaWAN is modelled, the power consumption of the sensor reading can simply be added to the results of the previously defined model. The energy consumption required for sensing depends on the sensors used; different sensors will have different sampling rates, sampling duration, and current consumption for the particular sensor module. In addition, sampling can occur as a result of some external stimulus (resulting in event-based traffic) or can take place after a regular interval (resulting in periodic traffic). The use case considered requires just a single reading immediately prior to the transmission of an uplink frame, leading to the simple expression:

$$P_{sens} = \frac{E_{sample}}{t} \quad \text{[W]} \quad (2.32)$$

where t is the periodicity of uplink transmissions and is taken as an input to the analytical model,

and E_{sample} is the energy consumption to take one reading from a sensor, and is dependent on the sensors used.

The representative sensor is an SHT3x-DIS temperature and humidity sensor [49] using a field-effect transistor (FET) isolation switch, to minimise the sleep current. An empirical characterisation of the system was performed by measuring the current consumption and time taken in each state reached in retrieving a sample from the sensor, when operating at 3.3 V. These are used to calculate E_{sample}, and are provided in Table 2.13.

Table 2.13: Measurements of SHT3x-DIS

Measurement	**Value**
I_{sense}	13.8 mA
t_{sense}	13.2 ms
E_{sample}	0.601128 mJ

Similarly to the LoRa use case described in [14], as a simple reporting application is modelled there is no required processing of the data on the device, and so this factor can be omitted. The remaining energy consumption is thus once again dependent on the remaining time per transmission period when the device is not active, where the device is in the lowest power state. However, differently from in the original model, this time period also factors in the time required for sensing:

$$E_{sleep}(t, DR) = I_{sleep} * (t - t_{packet}(DR) - t_{sense}) * Vcc \quad \text{[J]} \quad (2.33)$$

Finally, the preceding formulae can be recombined to redefine Equation 2.31:

$$\begin{aligned} P_{system}(t) &= P_{sens}(t) + P_{net}(t, DR) + P_{prc}(t) + P_{sys}(t) \\ &= \frac{E_{sample} + E_{packet}(DR) + E_{sleep}(t, DR)}{t} \end{aligned} \quad \text{[W]} \quad (2.34)$$

which defines the average amount of energy consumed by the entire system in a time period t, where there is only one transmission in the period. Additionally, the total amount of energy consumed by the system in that time period is:

$$E_{system}(t) = E_{sample} + E_{packet}(DR) + E_{sleep}(t, DR) \quad \text{[J]} \quad (2.35)$$

2.3.2 Radio-Frequency Energy Harvesting

Next, the general structure of a typical ambient RF EH environment is analysed. The approach taken is to firstly quantify the available dc power based on RF measurement campaigns worldwide. Then, an analytical model of a real-life EH environment where an RF energy harvester scavenges

power over time from various existing RF sources is analysed. The impact of each EH component (rectenna, power management unit and storage device) on the system performance is discussed. Finally, an aggregated energy model is presented, taking into account the total amount of energy consumed by the system, as discussed in the previous section.

2.3.2.1 Ambient RF Power Level

The term RF energy harvesting refers to the usage of available RF energy from existing RF sources (as shown in Figure 2.14). The available RF energy is collected from an energy harvester that converts it to dc energy to supply the LoRaWAN application. Figure 2.15 shows the structure of a typical RF energy harvesting LoRaWAN node. Each of these components are discussed later in this section.

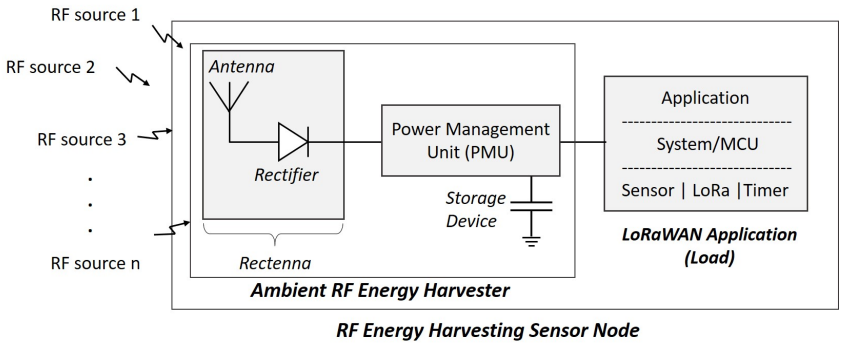


Figure 2.15: General Structure of a Typical Ambient RF Energy Harvesting LoRaWAN System

To quantify the power level availability, a series of measurement campaigns have been conducted. The results are either expressed in terms of ambient RF channel power density levels (usually measured in dBm/cm^2) or as the input power level at the antenna (measured in dBm or W), as shown in Figure 2.15. The measured power level at the antenna of the energy harvester device is used in this analysis. The ambient RF power level in China was measured within the frequency range of 0.7 GHz - 3 GHz outside a shopping mall and in residential areas in [50]. [51] evaluated the RF power density at GSM bands in Singapore with a peak power level of -31 dBm. The authors in [52] measured a peak power level of -32 dBm and -22 dBm half a meter from a mobile phone and a microwave oven, respectively. A set of indoor and outdoor measurements at Universidad Politecnica de Madrid (UPM) have also been carried out recently in Madrid [53]. Figure 2.16 summarises the peak measured indoor and outdoor power level at various frequencies in different locations around the world. For example, a reading of -31 dBm was recorded at 1848 MHz in Singapore, which corresponds to 0.31 μW. Closely spaced data points for the same location represent slightly different frequencies.

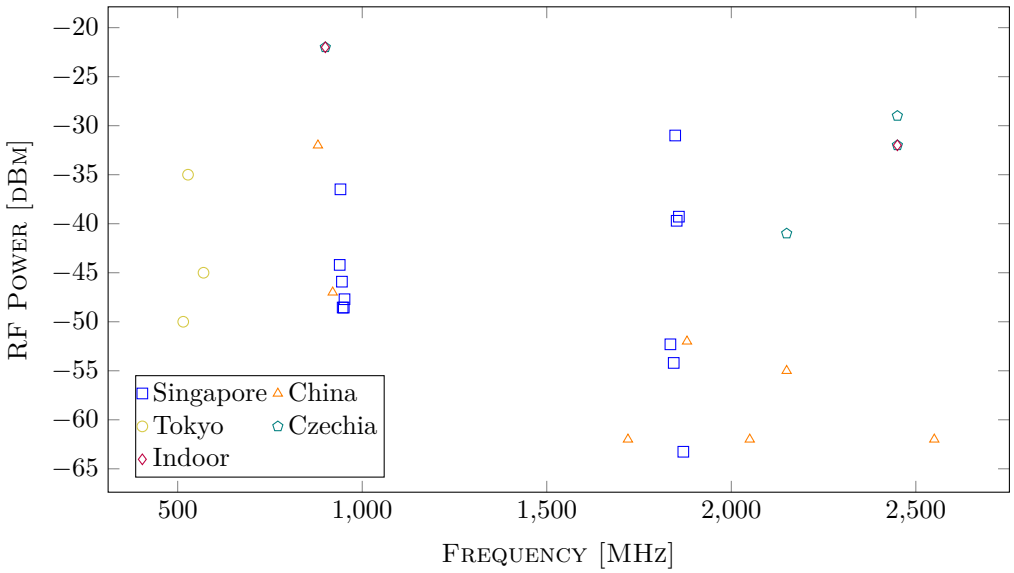


Figure 2.16: Measured Indoor and Outdoor Environmental Power Level at Single Frequencies Around the World [50], [51], [52], [54], [55]

Although the measurements shown in Figure 2.16 give an indication of the power level across a frequency band, to calculate the total instantaneous power across a wireless channel (from f_l to f_h), the following formula can be used:

$$P_{RFchan} = \int_{f_l}^{f_h} P_{RF(f)} df \quad \text{[W]} \quad (2.36)$$

where $P_{RF(f)}$ is the spectral power from f_l to f_h. Indeed, wider signals have the potential to accumulate more power if compared with narrower signals [51]. As an example, in [55], [56] the authors carried out measurement campaigns in downtown Japan and USA respectively from 50 MHz to 900 MHz and measured a peak carrier power level of -35 dBm at 6.3 km away from a TV broadcast source. However, across the band, the total measured power level was -8.99 dBm. As the available ambient RF power comes from various RF sources, the total instantaneous power from the multiple sources is an aggregate of the harvested power per each frequency channel:

$$P_{RFtot} = \sum_{i=c_1}^{c_n} P_{RFchan(i)} \quad \text{[W]} \quad (2.37)$$

where $P_{RFchan(i)}$ is the harvested power from each frequency channel $(c_1, ..., c_n)$. Due to its nature, ambient power can fluctuate over time. According to some recent measurements in Houston, the ambient RF power is relatively stable in the time-domain, however the available RF power is less stable during the daytime due to the activities of people [57, 58]. In each case, the available RF energy level in a time interval (from t_1 to t_2) can be calculated as [59]:

$$E_{RFtot}(t) = \int_{t_1}^{t_2} P_{RFtot(t)} dt \quad \text{[J]} \quad (2.38)$$

where $P_{RFtot(t)}$ is the RF power distribution versus time and indicates the energy harvester input power level. Table 2.14 shows the total measured RF power P_{RFtot} in various locations. For an easy illustration and comparison, from now on, all power levels will be expressed in W rather than dBm.

Table 2.14: Total Ambient RF Power and Harvested DC Power

Measurement Scenario	**Available RF Power**	**Harvested dc Power**
6.3km from TV Tower in Tokyo (512-566 MHz) [55]	126 μW	16 μW
Base stations & Wi-Fi Hotspots (London underground) [60, 61]	-	1-50 μW
Inside UPM (0.1-6 GHz) [53]	480 μW	-
Outside UPM (0.1-6 GHz) [53]	1535 μW	-
Inside UPM (770-820 & 920-970 MHz) [53]	380 μW	-
Outside UPM (770-820 & 920-970 MHz) [53]	1300 μW	>80 μW *

* Exact dc power not shown in this paper.

2.3.2.2 Rectenna

To scavenge RF power, a rectifying antenna (or rectenna) is needed to collect the RF power and convert it to useful dc power [62]. Rectennas usually consist of an antenna and a rectifier at its simplest form (as in Figure 2.15). The harvested dc power depends on the efficiency of the rectenna. The RF-to-dc conversion efficiency of the rectenna depends on its operating bandwidth; narrower rectennas tend to be more efficient [63]. Also, the RF-to-dc conversion efficiency depends on the output dc voltage and the RF input power level; for example, it can be in the order of 10% and 40% for 10 μW and 1 mW, respectively [64] [65]. To minimize the sensitivity of the rectifier on these parameters, novel circuits have been proposed in the literature [65]. Table 2.14 shows the aggregated dc power as measured in various locations worldwide. Note that the dc power is a few orders less than the RF power due to the low RF-to-dc efficiency of the rectenna. Based on these measurements, this research uses a range of 10 μW to 80 μW to represent a reasonable range of achievable dc power levels. This harvested power is used as an input parameter in the energy model. In this work, as a simplification, and due to the lack of high-granularity data on the energy variation reported in the measurement campaigns, an average figure is used. This is a reasonable approximation as the device is in the sleep state for long periods of time, and the integral of the energy harvested over this long period will be the same whether the actual energy level varies or not during this time. The energy harvested during a time interval (δt) can be calculated using Equation 2.39.

$$E_{harv}(t) = P_{harv} * \delta t \quad \text{[J]} \quad (2.39)$$

2.3.3 Power Management Unit

The microwatts (μW) of harvested dc power can be managed from power management unit (PMU) in conjunction with an energy storage element. The PMU circuitry can include boost-up/boost-down converters and/or regulators and can be either an off-the-shelf solution or custom made board

[66]. In [32], the authors use the off-the-shelf PMU BQ25504 for LoRaWAN, while in [67] the PMU LTC3108 is used in an EH system collecting energy from Wi-Fi networks. A PMU can increase the effective use of harvested power by using maximum power point tracking (MPPT) techniques [68], and by regulating the output voltage (which also decreases the current consumed by the load e.g. the CPU and transceiver). However, a PMU also consumes a small current to operate - this can become a significant factor when the load is in sleep mode. Also, any boost-up/boost-down and regulation operation are subject to losses, typically characterised by an efficiency figure for the PMU. Equation 2.40 is used in this work to study the impact of a PMU on the harvested power requirement. It reflects the relative energy used when compared to a system with no PMU.

$$E_{pmu} = E_{nopmu} * (1.0 + f_{pmu}) \quad \text{[J]} \quad (2.40)$$

where:

$$E_{nopmu} = E_{system} + E_{leak} \quad \text{[J]} \quad (2.41)$$

The factor f_{pmu} reflects the relative net loss or gain provided by the PMU operation. For no PMU, f_{pmu} has the value 0, for a net loss it is positive, and for a net gain is negative.

2.3.3.1 Storage Device

Capacitors, supercapacitors, batteries or a combination of the three can be used as the storage mechanism. Each storage option comes with advantages and disadvantages [69]. In this work, a single supercapacitor is adopted as the storage device due to its long life-time and in accordance with the required capacitance level. The maximum energy storage capability (E) depends on the capacitance C, as shown in Equation 2.42:

$$E = \frac{C}{2} * (V_{rated})^2 \quad \text{[J]} \quad (2.42)$$

where V_{rated} is the rated voltage of the capacitor, as per the manufacturer datasheet. However, in reality, a capacitor will operate across a range of useful voltages (from V_{min} to V_{max}) and the stored energy level will be:

$$E = \frac{C}{2} * (V_{max}^2 - V_{min}^2) \quad \text{[J]} \quad (2.43)$$

While large capacitors can store a large amount of energy, a long time is taken to charge/discharge and part of the stored energy is lost due to the leakage current $I_{leak}(t)$. The supercapacitor leakage power depends on the operating voltage and temperature, however in this work, for simplicity, the supercapacitor leakage power $E_{leak}(t)$ is considered to be constant (as per the datasheet), as also shown in [70], [71].

2.3.3.2 Combined Energy Model

The energy level without a PMU in the storage device at a specific time $t + \delta t$ can be modelled as shown in Equation 2.44.

$$E(t + \delta t) = E(t) + E_{harv}(\delta t) - E_{system}(\delta t) - E_{leak}(\delta t) \quad \text{[J]} \quad (2.44)$$

where $E_{harv}(\delta t)$ is the harvested dc energy and comes from Table 2.14, $E_{system}(\delta t)$ is the energy consumed by the load (LoRa, MCU (microcontroller unit), sensor and timer) and defined in Equation 2.34, and $E_{leak}(\delta t)$ is the energy loss due to the supercapacitor leakage current $I_{leak}(t)$ [72]. The voltage on the storage device (without a PMU) can be calculated at time $t + \delta t$ using the standard equation shown in Equation 2.45.

$$V(t + \delta t) = \sqrt{2 * E(t + \delta t)/C} \quad \text{[V]} \quad (2.45)$$

Equation 2.46 shows the stored energy level at a specific time $t + \delta t$ when a PMU is included.

$$E(t + \delta t) = E(t) + E_{harv}(\delta t) - E_{pmu}(\delta t) \quad \text{[J]} \quad (2.46)$$

The output voltage of the PMU is fixed in this case.

Table 2.15: Reference Parameters

Name	**Value**	**Comment**
Min operating voltage	2.7 V	SX1276 lower limit
Max operating voltage	3.7 V	ATmega32U4 upper limit
Base sleep current	510 nA	ATmega32U4, SX1276, PCF2123 Specifications
Storage capacitance	320 mF	GW109F
Base leakage current	1 μA	GW109F
Transmitting	35.5 mA	SX1276 at 14 dBm
Receiving	17 mA	SX1276
Standby	6.6 mA	SX1276
CPU only (active at 4 MHz)	5 mA	ATmega32U4
Sensor Sample	13.8 mA	SHT3x-DIS
Sensor Sample Time	0.0132 s	SHT3x-DIS

2.3.4 Results

The results shown in this section are calculated using the energy model presented in the previous section. The energy is calculated over a sleep-sense-and-transmit cycle of length $t = t_{periodicity}$. This assumes a single sense operation, a single packet (5-byte payload) sent, and a recharging/sleep time per cycle (send-and-transmit cycle). The minimum and maximum voltages are calculated using

Equation 2.45.

A base-level/reference device is used to evaluate the feasibility of a LoRaWAN node powered entirely from harvested RF energy. This is loosely based on the BSFrance LoRa32u4II board (SX1276 LoRa transceiver, ATmega32U4 CPU) with an external PCF2123 wakeup timer, GW109F supercapacitor, and SHT3x-DIS temperature and humidity sensor [49]. The use of an external wake-up timer enables a super low power sleep mode for the device when not in a transmission cycle. The values used for the baseline calculations are shown in Table 2.15. Note that the leakage and sleep current are varied in subsequent results to show the feasibility of such a node for variations in these values. The sensor was isolated from the voltage supply using a FET isolation switch to minimise the sleep current. An empirical characterisation of the sensor was performed to calculate the sensor sample current and time. Baseline calculations are presented with and without a PMU.

2.3.4.1 Base-level Results

Figure 2.17 shows the stored energy vs time for each of the data rates over one sense-and-transmit cycle using the baseline parameters. The energy level starts at a value that matches the lowest operating voltage (2.7 V), and gradually increases until enough energy is stored to perform a single sense and transmit operation. The energy level required is greatest for DR0, the slowest data rate, and smallest for DR5, the fastest data rate. The energy then drops back to the initial energy value over the short period of time required for the sense-and-transmit operation. Note that in practice, a node would monitor the stored energy level, and only start a sense-and-transmit operation once it was sufficient for the current data rate.

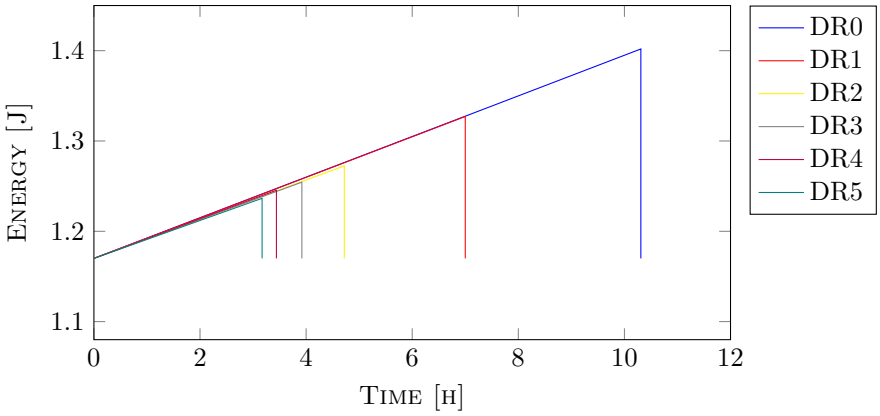


Figure 2.17: Stored Energy vs Time Over 1 Cycle for each Data Rate

Figure 2.18 shows the base-level sense-and-transmit cycle period for the different data rates, using a value for the harvested energy of 10 μW, which is at the low end of what the reviewed papers have measured. This result shows that a sensor in such an environment can operate using harvested wireless power with a period between 4 hours and 10 hours. Note that the shortest and longest cycles different by a factor of 2.5 – the data rate has a significant impact on the cycle time.

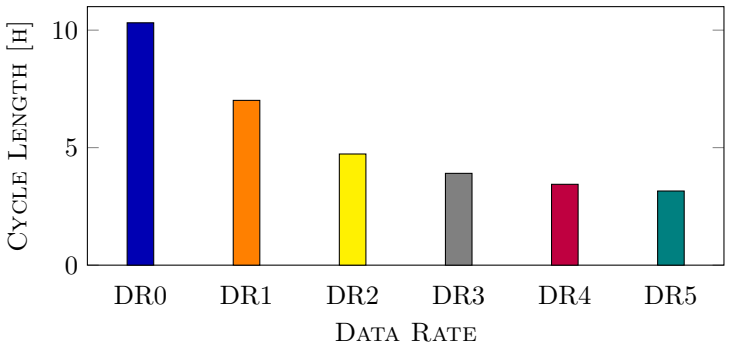


Figure 2.18: Cycle Lengths (Sense-and-Transmit) at 10μW Harvested Power

2.3.4.2 Sensitivity Results

A baseline is shown in Figure 2.19, which shows the required level of power that must be harvested in order to achieve one sense-and-transmit cycle per day at the different LoRaWAN data rates. Note that the increased losses during the sleep phase of the cycle make the power levels less sensitive to the data rate. The highest data rate requires just under 6 μW, and the lowest data rate requires just under 8 μW.

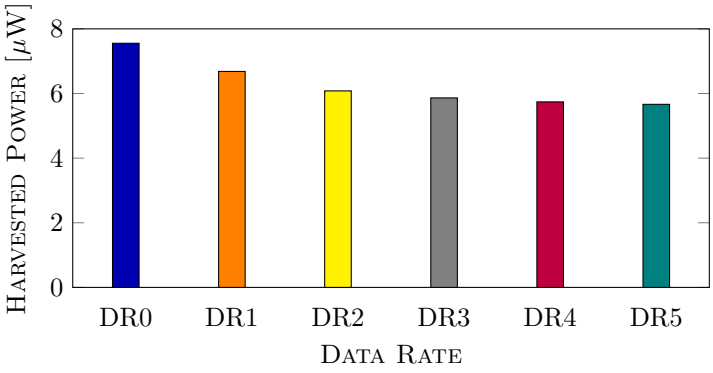


Figure 2.19: Required Harvested Power for 1 Sense-and-Transmit Cycle per Day

The capacitor leakage current is a significant limiting factor. The sensitivity of the results to this leakage current is shown in Figure 2.20. The dotted lines highlight the 10 μW and 80 μW power levels: from the RF energy harvesting results in the literature, results below 10 μW are regarded as having a high degree of feasibility, and results above 80 μW as having a low degree. This figure shows two results. By comparing the required energy level across the data rates for each leakage current, it can be seen that the transmit power has relatively little impact for leakage currents over 1 μA, and more impact for low leakage currents. However, at high leakage currents the leakage current dominates the power budget, and based on the reported figures for harvested wireless power, it is unlikely to be feasible to operate from harvested energy alone with a leakage current significantly over 10 μA.

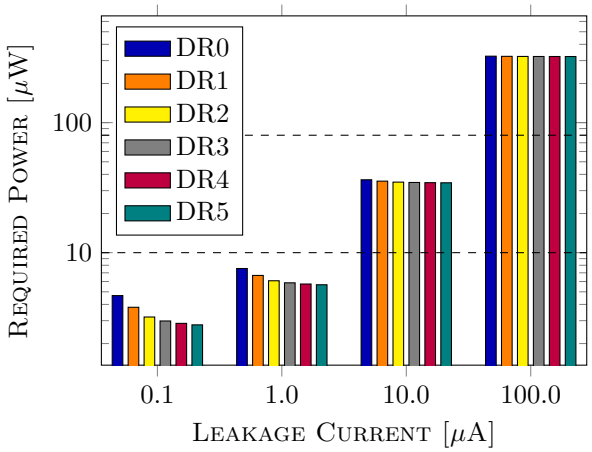


Figure 2.20: Log Plot Showing the Effect of Leakage Current on the Required Power

The sleep current (for the CPU and LoRa transmitter) is another significant limiting factor. The baseline results use a relatively low current, based on placing the CPU into its deepest sleep mode with a programmable external clock as described. The sensitivity of the results to the sleep current is shown in Figure 2.21. The dotted lines again highlight the 10 μW and 80 μW power levels. These results show that with the measured levels of harvested power (from 10 μW to 80 μW), a sensor node can support sleep mode currents slightly greater than 5 μA as long as the leakage current is low.

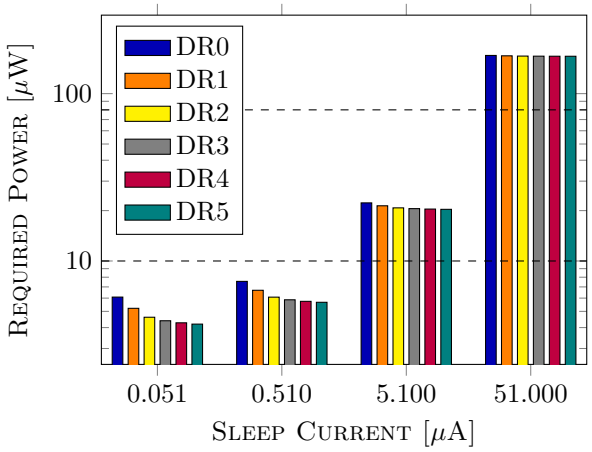


Figure 2.21: Log Plot Showing the Effect of Sleep Current on the Required Power

2.3.4.3 Sensitivity Results (with PMU)

The results shown previously do not include a PMU. Energy losses and inefficiencies in the PMU may be expected to reduce the effectiveness of these devices at the very low power levels typical of wireless harvesting. Figure 2.22 is calculated using Equation 2.46 to account for the impact of the PMU, which may produce an overall reduction (Net Gain) or increase (Net Loss) in the amount of power (in μW) required to be harvested by the device to operate at one cycle per day, compared to

a device with no PMU, as discussed in Section 2.3.3. The calculations for a net 50% gain show that a device can operate at 1 cycle per day down to 5 μW of harvested power. The calculations for a net 50% loss show that the required power to be harvested increases up to 15 μW.

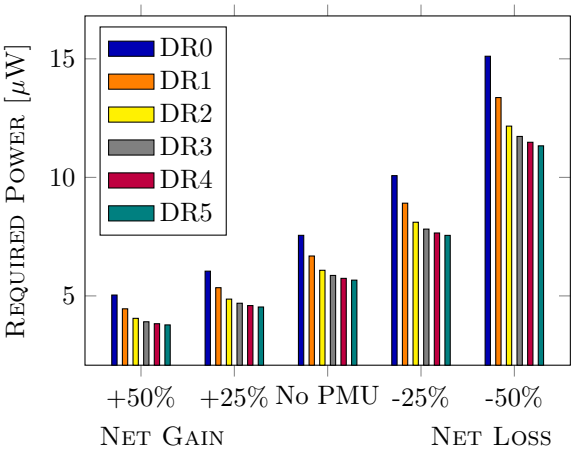


Figure 2.22: Effect of PMU Efficiency on the Required Power

2.3.5 Findings

The contribution of this section is to identify the boundaries of feasibility for using ambient RF energy harvesting for LoRaWAN sensor nodes. Four key design factors are identified: the required sense-and-transmit cycle length for the application, the leakage current, the sleep current, and the effect of a PMU. Two key environmental factors are also identified: the level of available ambient wireless energy in the waveband used by the energy harvester, and the distance from the base station (and background noise level), which determines the data rate (DR0-DR5) used, and thus the energy required to transmit a frame.

Based on published, real-world measurements of levels of harvestable energy at various locations worldwide, the results show that many urban locations have the potential to support daily, short-distance measurement reporting. Longer-distance communications, or more frequent reporting intervals, are shown to not always be sustainable through ambient RF energy harvesting only – i.e. where they exceed the boundaries of feasibility. Recently published results have shown that LoRaWAN sensor nodes are feasible using the higher levels of harvestable energy associated with solar and thermal energy for example, but this work identifies the conditions under which they can be also feasible at the lower energy levels associated with ambient RF harvesting. As hardware becomes more efficient, the design boundaries will change, making such scenarios ever more feasible – for example using the newer generation of LoRa transceivers (SX126* class). The importance of conducting site environmental surveys in order to determine whether the harvestable RF energy levels are sufficient for successful deployment is also emphasised. The main contribution of this work is to provide a systematic basis for the development of LoRaWAN-based sensor node platforms for environmental monitoring-style applications using ambient RF energy harvesting.

2.4 Conclusions

Energy efficiency is a key metric in the evaluation of LPWAN technologies. The work in this chapter provides findings on the energy consumption profile of LoRaWAN. Initially, a comparison of key LPWAN technologies in terms of energy efficiency has been performed, through the use of analytical models based on the MAC behaviour of the technologies. Results show a clear distinction between licensed and unlicensed LPWAN, where non-licensed LPWAN can achieve a lower energy consumption rate, but is limited in throughput in comparison to licensed options. Then, the LoRaWAN analytical model is used to evaluate the potential of powering LoRaWAN devices through RF energy harvesting. It is shown that while frequently reporting long-distance LoRaWAN applications are not always feasible through RF energy harvesting, especially in rural areas, many urban locations can support daily, short-distance measurement reporting.

In this chapter, the energy consumption profile of a LoRaWAN device has been considered standalone, with the data rate selected as a configurable parameter. However, in reality the nodes in the same vicinity of a device affect the energy efficiency of the device, as there will be an impact on the reliability of uplink and downlink communications. This is primarily as a result of on-air collisions on the uplink, and overload at the gateway on the downlink. This will change the used data rate, and so the energy consumption rate, for a device. The analytical work in this chapter provides a benchmark for verification of the energy consumption of representative device configurations in simulation. In the next chapter, extensive simulation of LoRaWAN is performed in ns-3 to show how devices behave in realistic large scale deployments.

CHAPTER 3

Evaluation of Large Scale LoRaWAN Networks in ns-3

LoRaWAN networks are expected to operate on a large scale, with individual gateways serving thousands of infrequently-reporting devices. This chapter focuses on large scale simulation of LoRaWAN networks in ns-3. Previous research simulating the LoRaWAN protocol, which only included the modelling of the core features of LoRaWAN Class A, is extended in this chapter to enable more realistic simulation of the full protocol.

In particular, this chapter focuses on the simulation of three main aspects of the LoRaWAN protocol in ns-3: the energy efficiency of the protocol, Class B mode, and the Adaptive Data Rate algorithm. Initially, a model of the energy consumption of a LoRa transceiver is implemented in ns-3 to enable the evaluation through simulation of LoRaWAN devices in terms of energy efficiency. Results from simulations are compared to the LoRaWAN analytical model and show that 1) the simulator correctly implements the LoRaWAN protocol and 2) the energy model is integrated successfully into the ns-3 module. This contribution permits the prediction of energy consumption of LoRaWAN devices in large scale simulations, and enables the analysis and evaluation of LoRaWAN MAC layer features in terms of energy efficiency.

The next area of focus is LoRaWAN Class B, which extends the functionality of Class A to enable Network Server-initiated downlink communication, at the expense of an increase in the power consumption. This feature enables downlink pings to devices and allows the possibility of multicast downlink frames, providing the means to develop evolving and reacting LoRaWAN-based applications. The final feature focused on in this chapter is the ADR algorithm of LoRaWAN, which modifies the data rate parameter of a device based on the current wireless conditions. An analysis of the original algorithm behaviour provides findings on the latency and convergence properties of the system, which is heavily dependent on the starting configuration of devices. For both Class B and

ADR, the research described in this chapter constitutes the first time either feature has been simulated and analysed in ns-3. This research enables the analysis and evaluation of realistic LoRaWAN deployments in large scale simulations.

3.1 Enabling Analysis of the Energy Efficiency of LoRaWAN in ns-3

This chapter begins with the development of a LoRa energy consumption research tool, enabling the analysis of large-scale LoRaWAN networks in terms of energy efficiency through simulation. As the expected size of LPWAN networks are with gateways expected to serve thousands of devices, realistically it is desirable to be able to perform this analysis through simulation. Real-world measurements of a standard LoRa chip were performed to develop a device profile, and the results were used to develop an energy consumption module in ns-3 to enable the evaluation of LoRa networks in terms of energy efficiency. The modular code has been adapted to function as part of three of the LoRaWAN ns-3 modules that have been thus far described in research [73, 74, 75]. The module has been evaluated through comparison to the theoretical limits computed through the analytical model of the LoRaWAN protocol introduced in the previous sections of this chapter. The results are comparable to those from the analytical model, showing both that the module has been correctly integrated into the externally-developed LoRaWAN module ([75]), and that [75] successfully simulates the LoRaWAN protocol. The contributions are an analysis of the energy consumption of different states in a LoRa transmission by the SX1272, a common LoRa transceiver, beyond what is provided in the datasheet, and an energy consumption module for use in three of the LoRaWAN ns-3 modules described in research, enabling more accurate energy consumption analysis of LoRa-based systems.

3.1.1 Real-World Measurements

Firstly, an analysis of the energy consumption of the LoRa SX1272 transceiver in each possible state is performed to derive a consumption profile beyond the subset of figures provided in the datasheet [76]. The setup is an RF-LORA-868-SO from RF Solutions with an integrated SX1272, mounted onto an atmega128RFA1 microcontroller. The current consumption was measured across the voltage and ground of the RF board (i.e. just the transceiver) using a Keithley 2000 multimeter, with the power of each transmission measured using an HP 8595E Spectrum Analyzer connected to the antenna pin of the RF board, terminating each transmission in a matched 50 Ω impedance. The software on the microcontroller was a modified version of LMIC (LoRa-MAC-in-C), a software implementation of LoRaWAN, integrated into Contiki OS. The transmission settings are those outlined in Table 3.1. The SX1272 also supports other non-LoRa modulation techniques (FSK and On-Off Keying (OOK)), but these are not modelled. The program on the microcontroller transmitted a Class A LoRaWAN frame using DR0 every 30 seconds. The values presented are based off of a minimum of twenty measurements for each reading, and the mean, min and max are provided for each. All

of the values presented in this section are derived from measurements of the transceiver; even the values for which there was a typical current consumption provided in the datasheet were repeated for verification.

Table 3.1: Default Values for Measurements

Setting	**Value**
Supply Voltage	3.3 V
Frequency	channel hopping in 868 MHz band
LoRaWAN Data Rate	DR0
Forward Error Correction	4/5
CRC	Enabled
Payload length	51 bytes
Preamble Length	8 symbols
Temperature	22.0°C

The SX1272 defines a set of operating modes which can be selected via SPI (Serial Peripheral Interface) to control the device's behaviour [76]. These modes are:

1. Sleep – Access to SPI available, configuration registers can be written and read, ability to switch between FSK/OOK and LoRa mode available. The LoRa FIFO (first in, first out) buffer cannot be read.
2. Standby – LoRa baseband blocks and crystal oscillator are on, phased-locked loops (PLLs) and RF front-end are off. Registers on the devices can be modified. The LoRa FIFO buffer can be read. The device is in this state after a transmission, while waiting for the receive windows to begin.
3. FSTX – Frequency synthesis mode for transmission. Selected PLL is active, RF front-end is off.
4. FSRX – Frequency synthesis mode for reception. Selected PLL is active, RF front-end is off.
5. TX – Transmit mode. All blocks required for transmit are on, Power Amplifier (PA) is ramped. Once the packet is transmitted the device returns to Standby mode.
6. CAD – Channel Activity Detection mode. Device checks a channel for a LoRa preamble.
7. RXSINGLE – Receive mode. All blocks required for reception are on, the device remains in receive mode until a packet is received, then returns to Standby mode.
8. RXCONTINUOUS – Receive mode. All blocks required for reception are on, the device remains in continuous receive mode.

The current consumption observed in each of these modes corresponds to the set of states listed in Table 3.2. An analysis of these states is thus performed to build a device energy consumption profile.

Table 3.2: SX1272 Modes and States

Device Mode	**Consumption State**
Sleep	Sleep
Standby	Standby
Freq. Synthesis TX	Synthesizer (TX)
Freq. Synthesis RX	Synthesizer (RX)
TX	TX
Channel Activity Detection	RX
RXCONTINUOUS	RX
RXSINGLE	RX

3.1.1.1 LoRa States Without Configuration Parameters

The current consumption for the three initial states outlined in Table 3.2, which have no further configuration parameters, are outlined in Table 3.3.

Table 3.3: SX1272 Other States Current Measurements

State	**Typ (mA)**
Sleep	10^{-3}
Standby	1.6 ± 0.042
Synthesizer (TX)	5.7 ± 0.154
Synthesizer (RX)	5.0 ± 0.145

3.1.1.2 RX Current Consumption

The SX1272 provides a Low Noise Amplifier (LNA) boost option, which improves performance in RX mode while increasing current consumption. Values for both configurations are given in Table 3.4.

Table 3.4: SX1272 RX Current Measurements

State	**Bandwidth (kHz)**	**Typ (mA)**
With LNA Boost	125	11 ± 0.014
With LNA Boost	250	12 ± 0.017
Without LNA Boost	125	10 ± 0.015
Without LNA Boost	250	11 ± 0.023

3.1.1.3 TX Current Consumption

There are two configurations for TX power selection on the SX1272 connected to two different output pins, and as such the choice of block must be made both in hardware and software, as shown in Figure 3.1. These are:

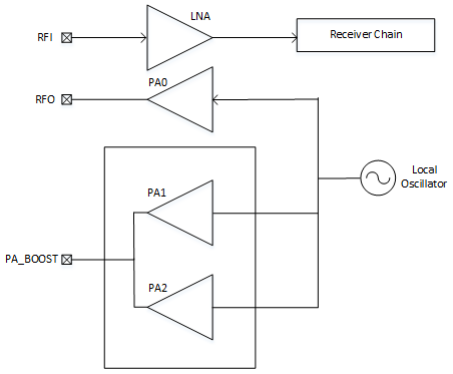


Figure 3.1: SX1272 PA Configuration

1. RFO pin - connected to the RFO pin is the unregulated single low power amplifier PA0, which can deliver an output power of between -1 and +14dBm, in 1dB steps.
2. PA_BOOST pin - connected to the PA_BOOST pin are the internally regulated PA1 and PA2 power amplifiers, which enable an output RF power of between +2 and +17dBm, in 1dB steps, as well as a High Power Operation (HPO) mode of +20dBm.

The particular LoRa module used, the RF-LORA-868-SO, only provides an output to the PA_BOOST pin. A set of values for that configuration are given in Table 3.5.

Table 3.5: SX1272 TX Current Measurements

State	**Typ (mA)**	**State**	**Typ (mA)**
PA_Boost + HPO		PA_Boost	
+20dBm	105 ± 0.812	+10dBm	42 ± 0.171
PA_Boost		+9dBm	40 ± 0.131
+17dBm	77 ± 0.475	+8dBm	39 ± 0.100
+16dBm	69 ± 0.526	+7dBm	37 ± 0.074
+15dBm	62 ± 0.809	+6dBm	36 ± 0.060
+14dBm	54 ± 0.408	+5dBm	35 ± 0.093
+13dBm	50 ± 0.462	+4dBm	34 ± 0.037
+12dBm	47 ± 0.284	+3dBm	33 ± 0.039
+11dBm	44 ± 0.219	+2dBm	32 ± 0.016

3.1.2 Device Energy Model Class in ns-3

A framework providing a general method of implementing energy models in ns-3 was introduced in [77]. The framework separates the modelling of energy consumption into components for expenditure and sources, with standard interfaces then enabling interaction. An entirely modular energy consumption framework is an inherently abstracted approach to profiling the energy consumption of a real device [78], but nevertheless provides an accurate depiction of the energy consumption of individual components [77], and is also the sole methodology for energy model integration that has been accepted into the official ns-3 repositories. This framework was initially used to create energy

models for Wi-Fi, and has since been applied to LTE [79] and IEEE 802.15.4 [80]. The contribution in this section is to apply this full methodology to LoRaWAN.

The design of the LoRaWAN Energy Model follows the framework from [77]. The model is integrated into the existing LoRaWAN implementation solely through the tracing of the relevant PHY layer parameters. Pseudocode of the integration of the model is provided in Code Extract 3.1. For each LoRaWAN device, an energy model is created and linked to the energy source of the device, which models the battery (line number 7-14). Then the required PHY layer variables are traced to connect the device PHY and the energy model (line number 17-20).

Code Extract 3.1: Integration of the Energy Model into a LoRaWAN Module

```
1  //create energy model factory
2  ObjectFactory m_radioEnergy.SetTypeId("ns3::LoRaWANRadioEnergyModel"
       );
3
4  Ptr<DeviceEnergyModel> LoRaWANRadioEnergyModelHelper::DoInstall(Ptr<
       LoRaWANNetDevice> device, Ptr<EnergySource> source)
5  {
6    //create the energy model
7    Ptr<LoRaWANRadioEnergyModel> model = m_radioEnergy.Create();
8    // set energy source pointer
9    model->SetEnergySource(source);
10   // and add model to device model list in energy source
11   source->AppendDeviceEnergyModel(model);
12   // set the current model
13   Ptr<LoRaWANCurrentModel> current = m_currentModel.Create<
         SX1272LoRaWANCurrentModel>();
14   model->SetCurrentModel(current);
15
16   // create and register energy model Phy listeners
17   Ptr<LoRaWANPhy> phy = device->GetPhy ();
18   phy->Trace("TrxState",MakeCallback(&LoRaWANRadioEnergyModel::
         ChangeLoRaWANState, model));
19   phy->Trace("TxPower",MakeCallback(&LoRaWANRadioEnergyModel::
         SetTxCurrentA, model));
20   phy->Trace("Bandwidth",MakeCallback(&LoRaWANRadioEnergyModel::
         SetRxCurrentA, model));
21   return model;
22 }
```

Any modification of the traced parameters triggers linked function calls in the energy model. In this way, the energy model can be integrated with minimal changes required to the original LoRaWAN model. The model can then be installed on each node in the ns-3 simulator setup ("scratch") file, as shown in Code Extract 3.2. Note that a battery capacity of 5 Wh is modelled; a choice of different capacity will result in shortened/extended curves of the same shape as those presented (depending on whether the capacity is reduced or increased).

Code Extract 3.2: Installation of the Energy Model into a LoRaWAN Node in ns-3

```
1    BasicEnergySourceHelper sourceHelper;
2    sourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue
         (18000)); // = 5Wh,
3    sourceHelper.Set("BasicEnergySupplyVoltageV", DoubleValue(3.3));
4    EnergySourceContainer energySources = sourceHelper.Install(
         endDeviceNodes);
5
6    LoRaWANRadioEnergyModelHelper radioHelper;
7    radioHelper.Install(lorawanEDDevices, energySources);
```

At the end of the simulation, the remaining energy on the installed source can be accessed as in Code Extract 3.3. Note that in this code extract, *endDeviceNodes* is a *NodeContainer* of the devices, which is a core object type in ns-3 and is the standard method of accessing information related to the simulated devices.

Code Extract 3.3: Accessing Final Energy Remaining at the End of the Simulation

```
1  for (d = endDeviceNodes.Begin(); d != endDeviceNodes.End(); ++d) {
2    std::cout << (*d)->GetId() << " " << energySources.Get((*d)->
         GetId())->GetRemainingEnergy()  << std::endl;
3  }
```

A model of the structure of the implemented system is provided in Figure 3.2, highlighting that the link between the models is only on the PHY layer (with the added model shown in green).

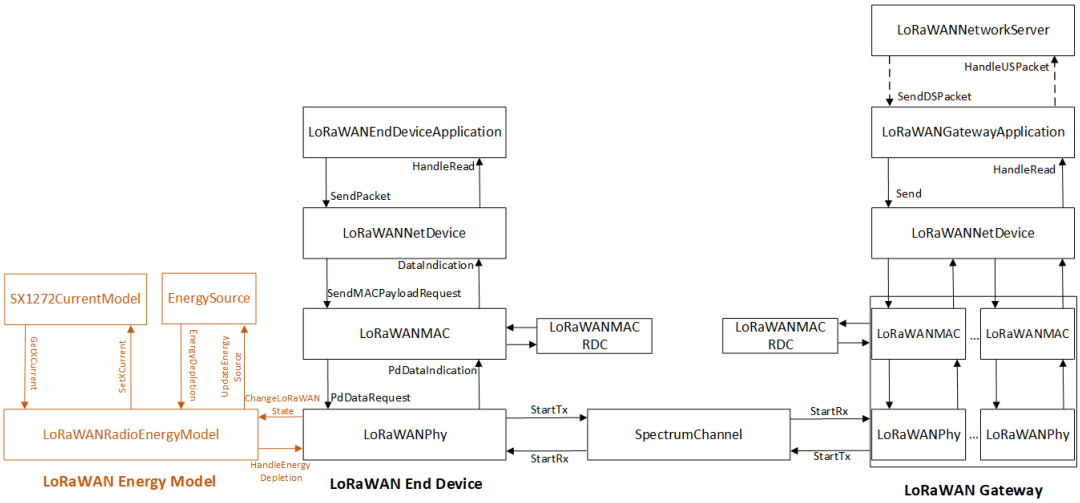


Figure 3.2: LoRaWAN Energy Model Integration

The LoRaWANEnergyModel class subtracts energy from the energy source of the device based on the state of the PHY layer of the device. Pseudocode of the energy depletion function is provided in Code Extract 3.4. This function is called whenever the PHY layer state changes, and the depleted energy is based on the previous state and the length of time the device was in this state.

Code Extract 3.4: Energy Depletion in the LoRaWANRadioEnergyModel

```
1  void LoRaWANRadioEnergyModel::ChangeLoRaWANState(newState)
2  {
3    Time duration = Simulator::Now() - m_lastUpdateTime;
4    double energyToDecrease = 0.0;
5    double supplyVoltage = m_source->GetSupplyVoltage();
6
7    switch(m_currentState)
8      {
9      case PHY_TRX_OFF:
10       energyToDecrease = duration*m_currentModel.GetSleepCurrentA();
11       break;
12     case PHY_IDLE:
13       energyToDecrease = duration*m_currentModel.GetIdleCurrentA();
14       break;
15     case PHY_RX_ON:
```

```
16          energyToDecrease = duration*m_currentModel.GetRxCurrentA();
17          break;
18        case PHY_TX_ON:
19          energyToDecrease = duration*m_currentModel.GetTxCurrentA();
20          break;
21        case PHY_BUSY_RX:
22          energyToDecrease = duration*m_currentModel.GetRxCurrentA();
23          break;
24        case PHY_BUSY_TX:
25          energyToDecrease = duration*m_currentModel.GetTxCurrentA();
26          break;
27        default:
28          NS_FATAL_ERROR("Undefined radio state");
29        }
30      energyToDecrease *= supplyVoltage;
31      m_lastUpdateTime = Simulator::Now();
32      m_source->UpdateEnergySource(energyToDecrease);
33      m_currentState = newState;
34    }
```

The overall structure of the module is shown in Figure 3.3. The core current consumption functions of the node are contained within a LoRaWANCurrentModel object inside the LoRaWANEnergyModel. The actual consumption values are defined in the SX1272LoRaWANCurrentModel class, which inherits from the abstract LoRaWANCurrentModel class and contains the readings described previously. This approach enables the easy implementation of other models in the future (e.g., for the SX1276). The only required changes to the model to integrate it into a LoRaWAN module are the links to the PHY layer of each implementation, which is limited to two model variables and the PHY layer traced values. This integration has been performed for three modules: [73][1], [74][2], [75][3].

[1] https://github.com/ConstantJoe/signtlabdei-lorawan-with-energy-model
[2] https://github.com/ConstantJoe/ku-leuven-lorawan-with-energy-model
[3] https://github.com/ConstantJoe/imec-idlab-lorawan-with-energy-model

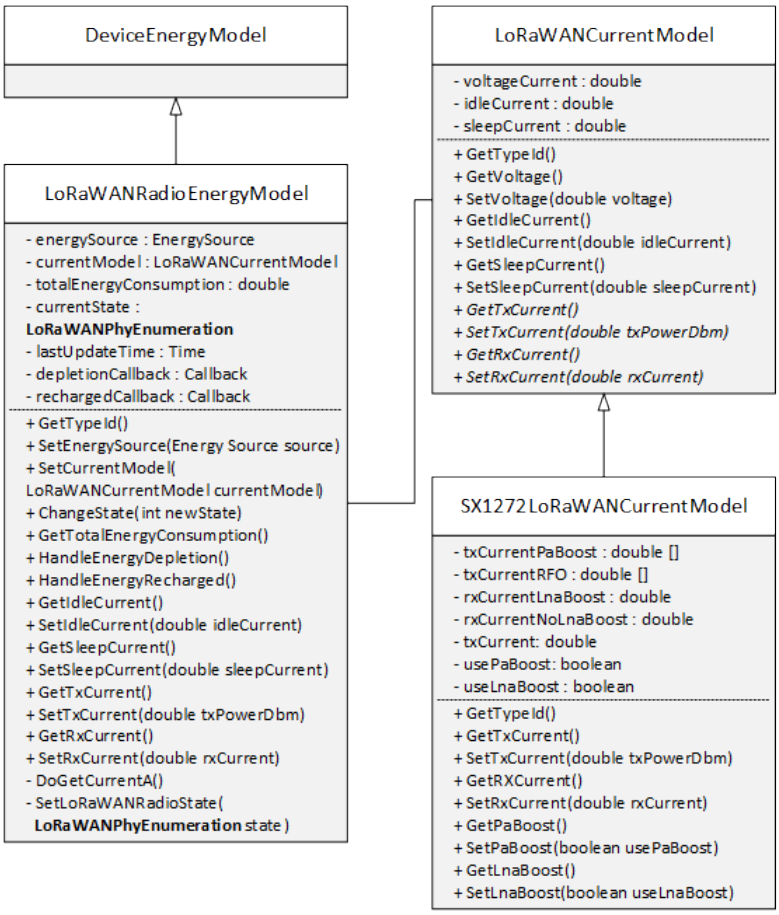


Figure 3.3: LoRaWAN Energy Model Class Diagram

3.1.3 Simulations

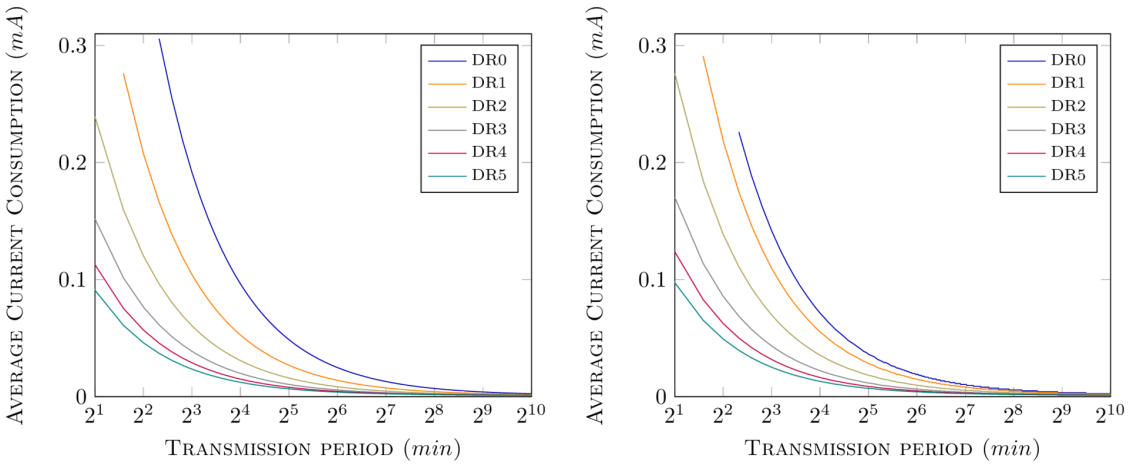
To judge the quality of the model, the energy model is used to calculate the average current consumption of the regular transmission of unacknowledged LoRaWAN packets for each data rate. These results are then compared to equivalent results derived from the analytical model introduced in the previous chapter.

The energy model was integrated into the module from [75]. Simulations consist of a network with 1 Network Server, 1 gateway, and 10 devices placed within a gateway-centred disk with a radius of 1 km. Each device sends 1 unacknowledged Class A LoRaWAN packet randomly within each T_{period}. Each simulation was run for 1 day, and the average current consumption of each device was calculated as:

$$I_{avg} = \frac{(E_{Source\_Initial} - E_{Source\_Final})/Vcc}{T_{day}} \quad [\text{A}] \quad (3.1)$$

3.1.4 Results

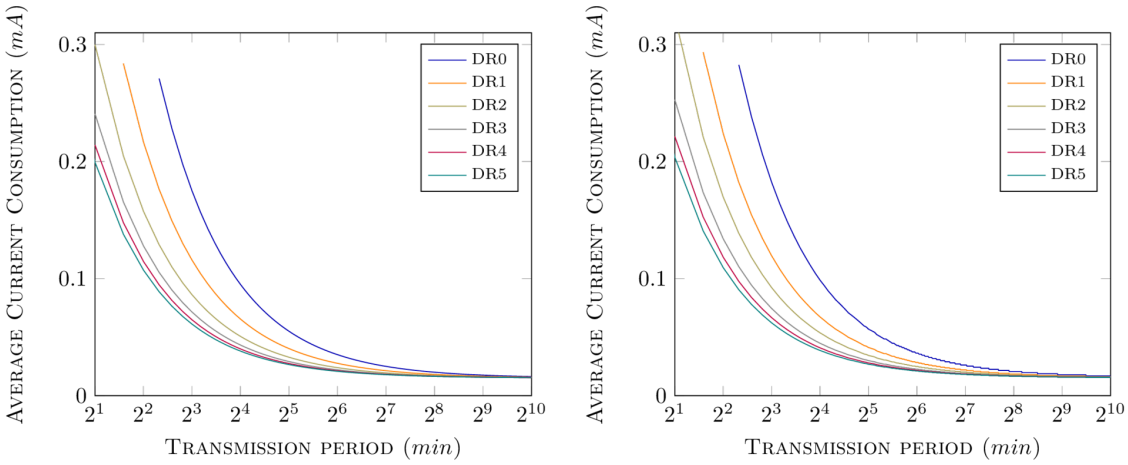
The adapted analytical model and the simulator were then used to generate the average current consumption for each data rate, for an increasing transmission period. The input variables for the model include the set captured in the previous section for the SX1272 transceiver, and also the set used in the previous chapter to model the current consumption of the BSFrance device.



(a) Analytical Model (b) ns-3 Simulations

Figure 3.4: Average Current Consumption (SX1272)

Figures 3.4 (a) and (b) show the average current consumption achieved as a function of the data rate and T_{period} for the SX1272, with (a) showing the theoretical limits calculated using the analytical model, and (b) showing the results generated using the simulator. The ultra-low sleep power of the transceiver enables an extremely low average current consumption for infrequently transmitting devices, for this component.



(a) Analytical Model (b) ns-3 Simulations

Figure 3.5: Average Current Consumption (BSFrance Board)

Figures 3.5 (a) and (b) show the equivalent results when modelling the BSFrance, a full device. The average current consumption for all data rates tends to the sleep current of the device as the transmission period increases, and this sleep current is orders of magnitude greater than for the SX1272. Note however that the transceiver used in this device enables a lower transmission power than in the results for the SX1272, as the PA_BOOST option can remain off.

Table 3.6: Average Difference Between the Model and Simulation Curves

Data Rate	**Average Difference**	
	SX1272	BSFrance
DR0	14.63 %	3.11 %
DR1	11.42 %	2.03 %
DR2	14.03 %	2.15 %
DR3	10.21 %	1.44 %
DR4	7.91 %	1.13 %
DR5	6.09 %	0.93 %

The average difference between the curves for both sets of results, for each data rate is provided in Table 3.6.

Note that the average difference for the chip model exceeds that of the device because of the much lower sleep current of the chip. For both sets of results, the equivalent curves and little deviation in the table show that the ns-3 module does model the LoRaWAN protocol. This work encapsulates the first time that a full model of the described chip has been developed in terms of energy efficiency (as opposed to the subset of results described in the datasheet), and as such enables researchers in the future to accurately model the energy consumption of the LoRa chip.

3.1.5 Findings

In this research, an analysis of the energy consumption of different states of the SX1272 was performed. The results were used to develop a LoRaWAN energy model for use in ns-3, integrated into three of the current publicly available LoRaWAN ns-3 modules. An evaluation of the model through comparisons with the previously-developed analytical model was performed. The results show that the developed simulator does accurately model the energy usage of LoRaWAN Class A. The average difference between the simulated curves and the analytical equivalent is 6.26%. This work permits the evaluation of large scale LoRaWAN networks in terms of energy efficiency, and enables the analysis and evaluation of LoRaWAN MAC layer features in terms of energy consumption.

3.2 Analysis of LoRaWAN Class B in ns-3

Appendix B outlined previous research of simulation of LoRaWAN in ns-3. One such project was [75], a module which was developed to enable the simulation of Class A devices in ns-3. In this

work the module has been extended to support the simulation of Class B devices[4]. The LoRa Alliance mandates that for a device to be LoRaWAN-compliant, only the implementation of Class A is mandatory. However, as Class B provides further downlink feedback possibilities for devices, it will provide support for evolving and reactive IoT applications, and thus requires further analysis. In particular, the simulator's MAC layer (which acts as the driver of the PHY layer and is shared by both LoRaWAN devices and gateways) and the application layers (which implement the Network Server, gateway, and device-specific functionality) have been modified. As Class B is a purely MAC-level mode, minimal changes to the PHY layer were required.

Using the extended module, the scalability of Class B of LoRaWAN is investigated, and the results show that the principle restriction on scalability is due to the duty cycle limits that the gateway must adhere to. An example application is simulated which requires the sending of a single 8-byte packet to every node every two and a half hours. A maximum of 64 devices using the slowest data rate for uplink and downlink can be supported while maintaining a PDR of >95%. Using the fastest data rate, this rises to over 1000 devices. In addition, a limitation in the protocol is identified which in certain configurations allows a gateway node to block the future transmission of its own beacon frames.

3.2.0.1 Research Analysing and Using LoRaWAN Class B

In [81], the delay of Class B downlink communications is studied through the use of a Markov chain model. It is found that the data rate and ping periodicity of devices are the two primary factors on the delay. The number of available channels for uplink transmissions also has a significant impact on the reception of confirmed downlink frames. However, in this work it should be noted that the duty cycle regulations of the European sub-GHz bands are not applied to the gateway nodes (which is unrealistic in an actual European deployment). As a result, every ping slot is available for use, and all downlink communications can be acknowledged with an uplink frame. In [82], an analytical model of Class B is developed, and used to derive a cost function for the optimal value of the ping periodicity. This cost function is based on the average delay, data throughput, and energy consumption of devices. For devices, there is a tradeoff between the delay and energy consumption. As the network scales, the cost function allocates a high rate of ping slots to devices to allocate for colliding slots. However, it should be noted that this work does not include any consideration of duty cycle regulations, as the simulation environment implements the regional parameters of India. A Class B-specific network attack is presented and analysed in [83], where an attacker spoofs LoRaWAN beacon frames. The frames are transmitted with a small offset in time to the regular beacon frames, causing Class B devices to lock to the wrong beacons. The attacker then increases the length of the offset over time, and then eventually stops transmitting beacons altogether. This results in locked devices being unable to synchronise time, and eventually causes all Class B devices to revert to Class A and begin a full beacon search again. Countermeasures to this attack are proposed.

In [84], the scalability of multicast Class B communications is studied, and used to propose modific-

[4]https://github.com/ConstantJoe/ns3-lorawan-class-B

ations to the class which increase the probability of beacon reception and downlink frame reception. In [85] and [86], LoRaWAN Class B is used in a smart grid application because of the availability of multicast communications. Downlink multicasts are used to send coordinated commands to interface protection systems. A scalability analysis of the proposed system is performed, and it is found that if faster data rates are used, a single LoRaWAN gateway can handle over 300 devices for this form of application. This is while maintaining adherence to the regional regulatory frameworks for these applications.

In [87], the beaconing and ping time scheduling concepts of Class B are used to enable a new channel access method for uplink communications. The methodology is effectively the same as described above for Class B, except uplink timeslots are scheduled instead of downlink ping slots. Results derived from a developed analytical model of the scheme show that an increase in the probability of successful reception of independent uplink frames can be achieved in comparison to legacy LoRaWAN Class A. In [88], the core of Class B has also been used to enable a TDMA-like structure for LoRaWAN uplink communications. In this work, the number of allocated timeslots has been reduced to lessen the impact of uplink on-air collisions for neighbouring timeslots, and slots allocated to devices can be used for both uplink and downlink communications. Additionally, a beacon skipping approach is taken to reduce the energy consumption impact of time synchronisation. The proposed scheme is evaluated in LoRaWANSIM, a LoRaWAN simulator, and the results show that the scheme outperforms legacy LoRaWAN in terms of achievable network throughput. Beaconing-based approaches to LoRaWAN uplink scheduling have inherent similarities to LoRaWAN Class B, for example in the work presented in [89] and [90]. These works and others (which are beacon-based but not directly based on LoRaWAN Class B) have been discussed in the earlier section on TDMA over LoRaWAN.

In [91], LoRa is combined with wake-up receivers to enable ping-able LoRa devices without the energy consumption of frequent idle listening [91]. This achieves the functionality of Class B without the overhead of beacon frames and ping slot listening. In [92], multicast over Class B is used to enable firmware updates over the air. In the proposed scheme, devices are selected by the Network Server to transition to Class B for the duration of the update procedure, and return to Class A mode once updated. In [93], an alternative downlink mechanism to Class B is proposed based on sequential polling, which outperforms Class B in terms of energy efficiency.

Overall, in comparison to Class A, there is much less available analysis of LoRaWAN Class B. This is particularly true in regards to the performance of large scale networks of Class B devices, as the primary focus of analysis has been of the delay of downlink frames.

3.2.1 Implementation of Class B in ns-3

A description of the inner workings of LoRaWAN Class B has already been provided in Appendix B. Thus in this subsection, only the changes required to implement the mode are described, and not the system itself.

3.2.1.1 Modifications to the Network Server & Gateway

In the implementation described in [75], the Network Server is a singleton object which encapsulates the functionality of a LoRaWAN Network Server, Application Server and Join Server, as well as performing the generation of downlink traffic. To implement Class B, the information the Network Server stores about each device has been extended to include the ping periodicity, Class B-specific channel, data rate, and code rate of each device. As the join procedure is not yet modelled in the simulator, an initial gateway is also assigned to every device at the start of each simulation, providing the Network Server with a prospective gateway to use to send Class B downlink frames to that device.

To implement this, an additional timer is added to the Network Server, which, if the device is in Class B mode, fires every 128 seconds and causes the Network Server to generate the beacon frame. In the simulator, each gateway has an array of 4096 vectors and the Network Server appends the device ID to the *ith* vector in order to signify that that device is to potentially have a frame sent to it in the *ith* ping slot of the beacon period. A new event is then generated to fire at the time of that ping slot. Pseudocode of this Network Server function is provided in Code Extract 3.5.

Code Extract 3.5: Sending of Beacons and the Scheduling of Class B Ping Slots

```
void LoRaWANNetworkServer::ClassBSendBeacon(){
  Time timestamp = Simulator::Now();
  Ptr<Packet> beacon =  GenerateBeacon(timestamp);
  for (auto gw = gateways.cbegin(); gw != gateways.end(); gw++)
  {
    if ((*gw)->CanSendImmediatelyOnChannel()) {
        (*gw)->SendBeacon(beacon);
    }
  }
  //schedule ping slots for all devices
  for (auto ed = endDevices.begin(); ed != endDevices.end(); ed++)
  {
    if(ed.isClassB)
    {
      uint dAddr = ed.Addr;
      int period = std::pow(2.0, 12) / ed.ClassBPingSlots;
      uint buf[16] = GenerateAesBuffer(timestamp, dAddr);
      AES aes;
      aes.SetKey(ed.key, 16);
      aes.Encrypt(buf, 16);
      uint offset = (buf[0] + buf[1]*256) % period;

      uint beacon_reserved = 2120; // ms
      uint slotLength = 30; // ms

      auto gw = ed.m_lastSeenGW;
      //calculate and schedule ping slots for this device
      for(int i=0;i< ed.ClassBPingSlots; i++)
      {
        uint pingSlot = offset + period*i;
        uint pingTime = beacon_reserved +  pingSlot * slotLength;

        (*gw)->RequestPingSlot(pingSlot, dAddr);
        Simulator::Schedule (pingTime, &LoRaWANNetworkServer::
            ClassBPingSlot, this, dAddr, pingTime);
      }
    }
```

During each event, if the Network Server has a data packet queued for the associated device, and if there is no device ahead of that device in the ping slot vector with a data packet queued, and if the gateway may immediately send on the associated channel (i.e. the gateway is not currently transmitting or receiving, and can send without breaking the duty cycle limits), then the queued packet is transmitted. Pseudocode of this is shown in Code Extract 3.6. By default, in Europe downlink Class B communications are sent at a frequency of $869.525MHz$, the same channel as the beacon frames.

Code Extract 3.6: Sending During Class B Ping Slot

```
void LoRaWANNetworkServer::ClassBPingSlot(uint devAddr, uint
    pingTime)
{
  // get device to send downlink to
  auto ed = endDevices.find(devAddr);
  auto gw = ed.lastSeenGW;

  // ensure packet can be sent
  if((*gw)->IsTopOfPingSlotQueue(pingTime, devAddr) && (*gw)->
      CanSendImmediatelyOnChannel())
  {
    Packet p = ed.ClassBQueue.pop_front();
    p.AddHeader(LoRaWANFrameHeader);
    (*gw)->SendDSPacket (p);
  }
}
```

3.2.1.2 Modifications to the Device

The device application class has been modified to include the ping periodicity of each device, as well as the Class B-specific channel, data rate, and code rate. If a device is in Class B, at the start of a simulation an event is fired using a timer every 128 seconds to put the device into receive mode, in order to receive the incoming beacon frame. The device then schedules events for each of those times, which cause the device to wake up to potentially receive a downlink frame. The events are scheduled in a similar manner to the example given in the major loop of Code Extract 3.5.

3.2.1.3 Modifications to the MAC Layer

The MAC layer of the simulator has also been modified to implement Class B behaviour. Two additional receive states have been added to the device, for the beacon and Class B downlink receives respectively. The MAC layer state changes can be seen in Figure 3.6, with new states shown in red.

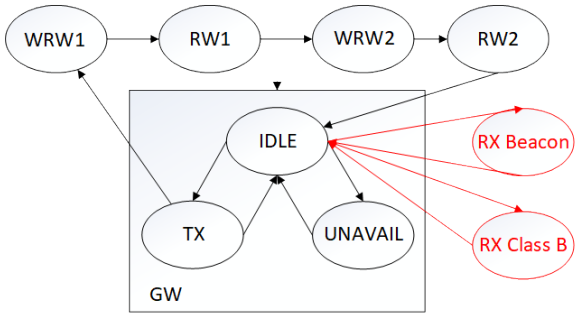


Figure 3.6: Finite State Machine Showing the Required Changes to the MAC Layer

The new finite state machine (FSM) reflects the uncoordinated nature of Class A and Class B communication in a LoRaWAN device. A device may only attempt to receive a beacon or downlink ping when it is in the idle state. Similarly, a device may not attempt to send an uplink packet when currently receiving a Class B downlink or beacon frame. Attempted transitions between states outside of the relations depicted in Figure 3.6 are prevented. The transition to *RX_BEACON* or *RX_CLASS_B* causes the device to attempt to receive the expected length of LoRa symbols as the preamble of the incoming packet, on the pre-set channel and data rate.

3.2.1.4 Modifications to the PHY Layer

No changes have been made to the PHY layer of the simulator, beyond the use of a longer preamble for beacon frames. As the beacon frame has a different and longer preamble than other LoRaWAN frames, the length of the preamble is factored in to all tags in each layer, and on the PHY layer the longer preamble length is included in calculating the expected transmission time of the packet.

Overall, the modifications to the structure of the simulator module are shown in Figure 3.7, in blue.

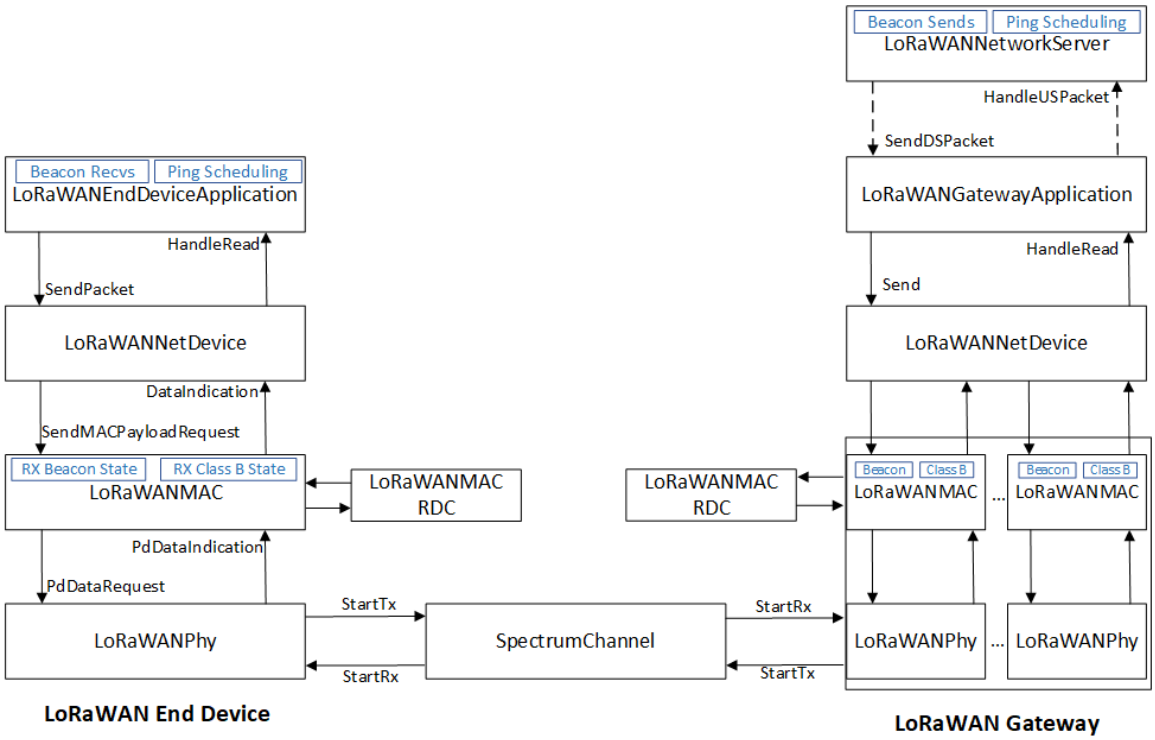


Figure 3.7: LoRaWAN Class B Integration

3.2.2 Scalability Analysis of Class B

The developed simulator was used to assess the performance of a LoRaWAN network with various downlink requirements. Each experiment consists of a LoRaWAN network with one gateway and a virtual Network Server. The simulation parameters are outlined in Table 3.7. A number of LoRaWAN devices operating in Class B mode are created and randomly placed within a gateway-centred disk with a radius of 1500 m (enabling transmissions from devices using any possible data rate to reach the gateway). The analysed throughput requirements were an uplink data generation period of 900 s, and a downlink period of 9000 s. This is consistent with the expected throughput requirements of LoRaWAN applications, where uplink forms the vast majority of network traffic. 8-byte payloads are transmitted, both in the uplink and downlink. Simulations were run for one day.

Table 3.7: Simulation Parameters

Gateways	1
Devices	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096
Disk Radius	1500 m
Uplink Period	900 s
Downlink Period	9000 s
Packet Size	8 bytes (excluding header)
Random Streams	5
Simulation Time	86400 s (1 simulated day)

In the simulations, all downlink communication takes place using the Class B ping slots i.e. Class A receive slots are scheduled for all devices, but not used. In each simulation, the chosen data rate is equal across all devices and remains static for the duration of the experiment. All other LoRaWAN parameters were set to the default in the standard. The coding rate used in transmission was 4/5, and the ping periodicity used was 6, corresponding to two ping slots being assigned to each device per beacon period. These are the default values for these parameters in the LoRaWAN standard. Five random streams for each experiment are run, and so each data point represents the mean of 5 runs of the same simulation with a different random stream. The maximum standard error across all of the described metrics (all of which are percentages) in this section is 0.8.

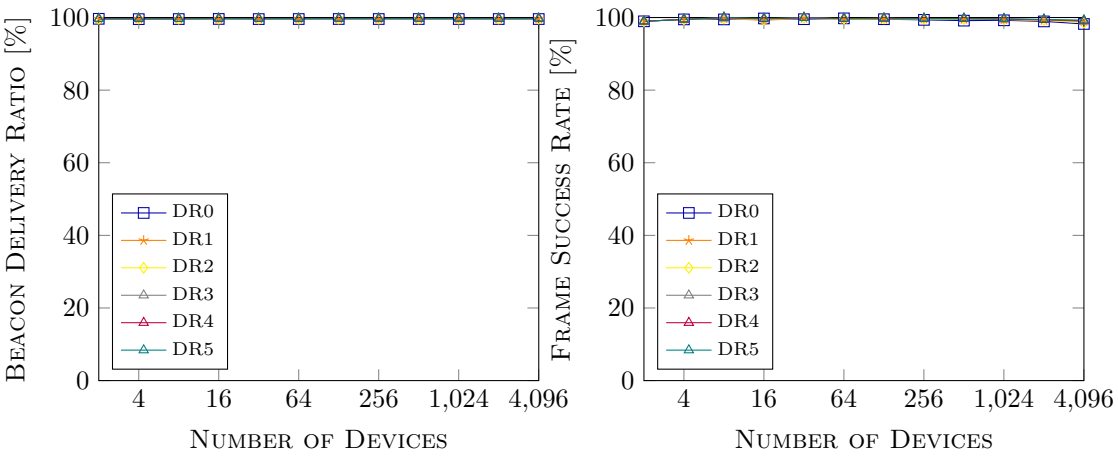
As neither Class B beacon-tracking nor clock drift are implemented in the simulator, in the case where the device fails to receive any beacon frames in the previous two hours, the device transitions back to Class A mode. This is indicated in the next uplink frame from the device. The device stops waking for beacons and ping slots. The Network Server stops scheduling ping slots for the device. However, traffic is still generated for the device, it is just never sent.

The spectrum access and max transmission power of a LoRaWAN channel depend on the subband. In Europe, the LoRa Alliance mandates that 3 channels inside subband *48* must be implemented. In addition, by default beacon frames and Class B downlink frames are transmitted in a channel inside subband *54*. The core of the analysis focuses on how the protocol performs when using these subbands. The particular aspects of LoRaWAN Class B to be analysed were as follows:

- The scalability of a network of LoRaWAN Class B devices, where downlink pings are sent in a channel in a subband with a duty cycle limit of 1% (e.g. subband *48*).
- The scalability of a network of LoRaWAN Class B devices, where downlink pings are sent in a channel in the subband with a duty cycle limit of 10% (i.e. subband *54*), which is shared with the transmission of beacon frames.

3.2.2.1 Transmission in Channel With 1% Limit

Figure 3.8 (a) shows the percentage of Class B beacons that were received by devices. It can be seen that, due to the transmission of beacon frames in a separate channel, these frames are reliably sent by the gateway and received by devices. Any failed reception of beacon frames in this case are due to the simultaneous transmission by a device during the time in which a beacon frame is due to be received. For each data rate, the on-air frame success rate is consistently >95%, as can be seen in Figure 3.8 (b); devices wake up at the correct time to receive packets, and packets sent by the gateway are successfully delivered. Lost packets are due to a "collision" between the ALOHA-based MAC layer of the uplink and the scheduled traffic of the Class B downlink on the device (i.e. a device attempts to transmit an uplink frame during the time in which a Class B downlink packet is due to be received), with a nominal amount due to on-air collisions.

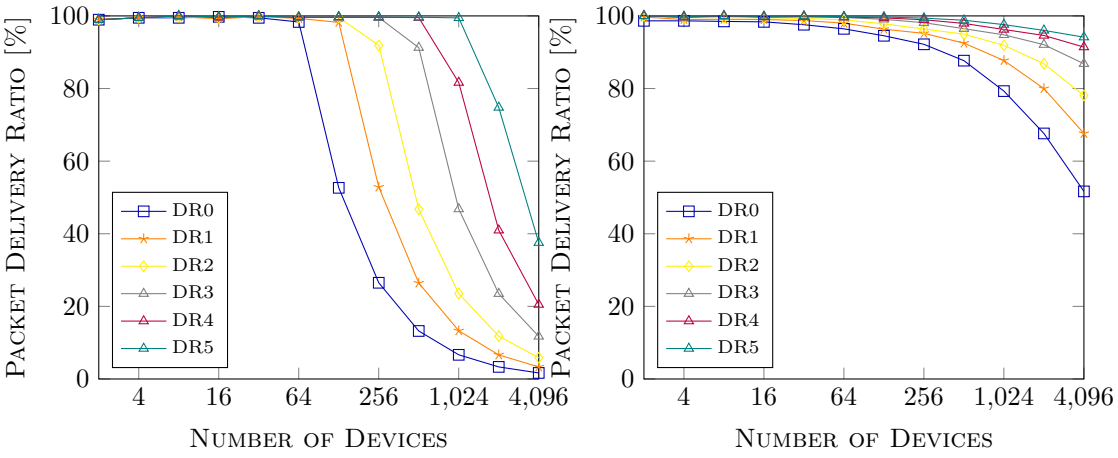


(a) Beacon Delivery Ratio

(b) Frame Success Rate

Figure 3.8: Beacon Delivery Ratio and Frame Success Rate (1% band)

However, as can be seen in Figure 3.9 (a), the Packet Delivery Ratio (including all generated and queued traffic, not just traffic that reaches on-air) of the network drops quickly with scale. As the number of devices increases, the number of downlink packets queued for transmission increases proportionally, but the number of packets actually transmitted remains limited by the duty cycle regulations. By comparing Figures 3.8 (b) and 3.9 (a), it can be seen that the vast majority of lost packets are never actually sent i.e. packets are generated but cannot be transmitted. For this particular traffic pattern, only a maximum of about 64 Class B devices can be supported while maintaining a PDR of >90% when the slowest data rate is used. By comparison, for the same traffic pattern the fastest data rate can support over 1000 devices. As can be seen in Figure 3.9 (b), the PDR of uplink frames remains high, even with ten times the amount of uplink data being sent compared to downlink.



(a) Packet Delivery Ratio (Downlink)

(b) Packet Delivery Ratio (Uplink)

Figure 3.9: Packet Delivery Ratio of Uplink and Downlink Frames (1% band)

To demonstrate the effect of the duty cycle regulations, Figure 3.10 shows the average actual number of received packets per data rate, in the day of simulation. It can be seen that the number of transmitted packets eventually hits a peak for each data rate. This is the maximum number of packets that can be transmitted by a gateway to adhere to the duty cycle limits. The maximum number of ping slots for each beacon period is utilised each period by the gateway, while maintaining adherence to the duty cycle limits.

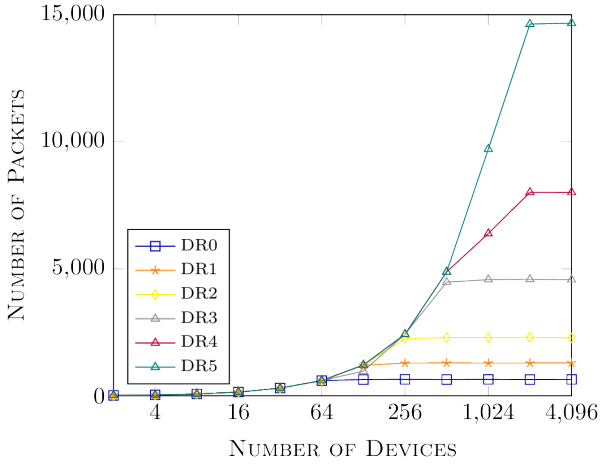


Figure 3.10: Total Number of Received Frames

To demonstrate the behaviour of the scheduling and usage of ping slots in LoRaWAN Class B, Figure 3.11 shows the total number of times each ping slot was allocated by the Network Server in a network of 64 devices.

A full description of LoRaWAN Class B is provided in Appendix B. Briefly, the allocation of slots in Class B operates as follows:

- A beacon period consists of $2^12 = 4096$ 30 ms slots
- Each device requests a number of ping slots in the beacon period, which must be a power of 2: $period = 2^{12}/slots$
- For each device, the initial slot in the period to be used is calculated, first by applying the AES-128 encryption algorithm to a padded block consisting of the beacon timestamp *(Time)* and the device address *(DevAddr)*, using a key of all zeroes: $R = AES128\_enc(16 * (0x00), Time|DevAddr|pad16)$.
- The first two bytes of this calculation are used to generate the index of the slot: $O = (R[0] + R[1] * 256)\%period$. Since the slot calculation is based on the applying the AES-128 encryption algorithm on the beacon timestamp and device address, the probability of use of each slot is equal, and the usage of each slot by an individual device each beacon period is randomised.
- Each subsequent slot is equally spaced in the beacon period: $timings = \{O + x * period \mid x < slots, x \in \mathbb{N}\}$

In the network demonstrated in Figure 3.11, each device transmits and receives frames using DR1, and receives downlink frames in a channel in subband *48* (with a duty cycle limit of 1%). Each device also uses a ping periodicity value of 6 (i.e. two ping slots per beacon period are scheduled). The network is run for 7 simulated days. It can be seen that a pattern emerges in the allocation, as would be expected considering how the allocation is performed, since every device is allocated the same number of slots, and the gap between slots for a device are equally spaced. A red vertical line indicates the 2048th slot, which is the middle slot in the beacon period. An initial offset is calculated each ping period for each device, in the interval [0: $\frac{4096}{p}$], where p is the number of ping slots to be allocated in the period. The remaining $p-1$ slots are then equally spaced in the period by gaps of $\frac{4096}{p}$ slots. Since every device uses the same value of p in the simulation, a pattern emerges in the allocation.

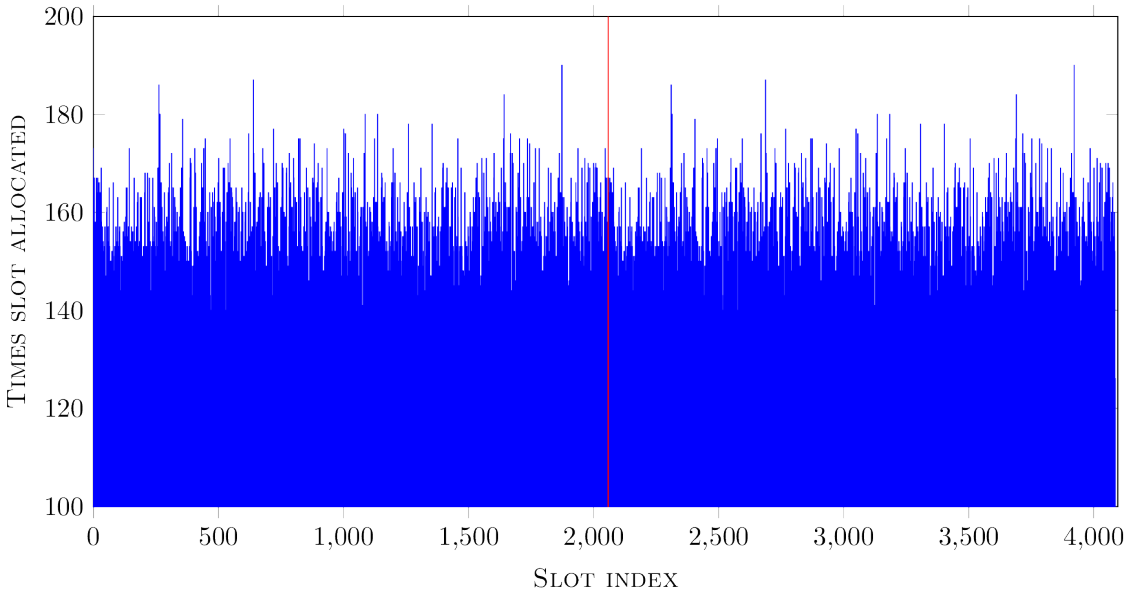


Figure 3.11: Allocation of Slots in the Class B Beacon Period (64 Devices)

However, the usage of the ping slots does not follow the same pattern. Figure 3.12 shows which of these ping slots were actually used. The usage of the ping slots follows a roughly random pattern. and indeed it can be seen that a pattern emerges that not all slots are used equally. The gaps between the peaks correspond to the transmission time plus the amount of time off band.

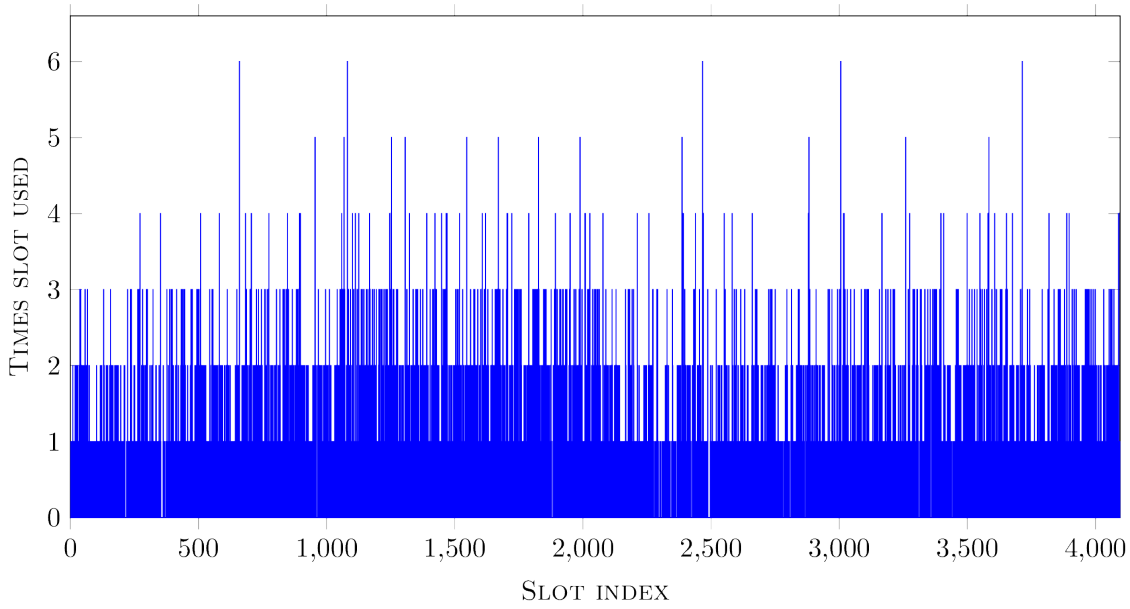


Figure 3.12: Usage of Slots in the Class B Beacon Period (64 Devices)

Figures 3.13 and 3.14 show the equivalent graphs for a simulation of 512 devices. While the allocation pattern follows the same principles as the smaller network, the usage pattern shows more evidently the effect of the duty cycle regulations. In this network, downlink queues for each device build up over time, and the usable slots for the gateway become limited to only a fraction of the available slots. In the simulation, because all devices use the same data rate and packet length, a recognisable pattern emerges in the allocation, because the transmission time plus the amount of time off band is a constant.

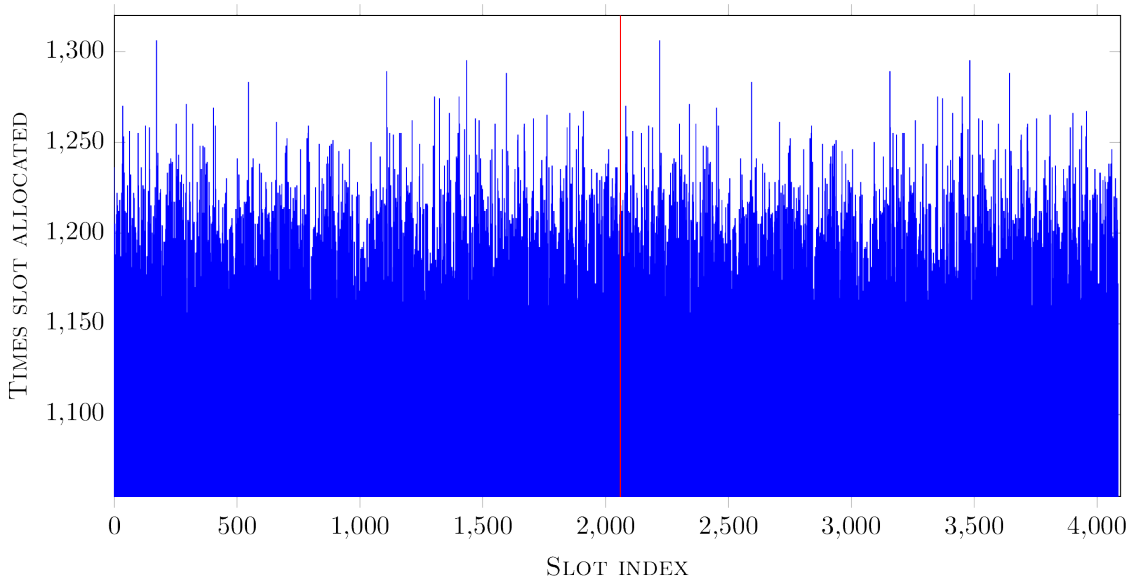


Figure 3.13: Allocation of Slots in the Class B Beacon Period (512 Devices)

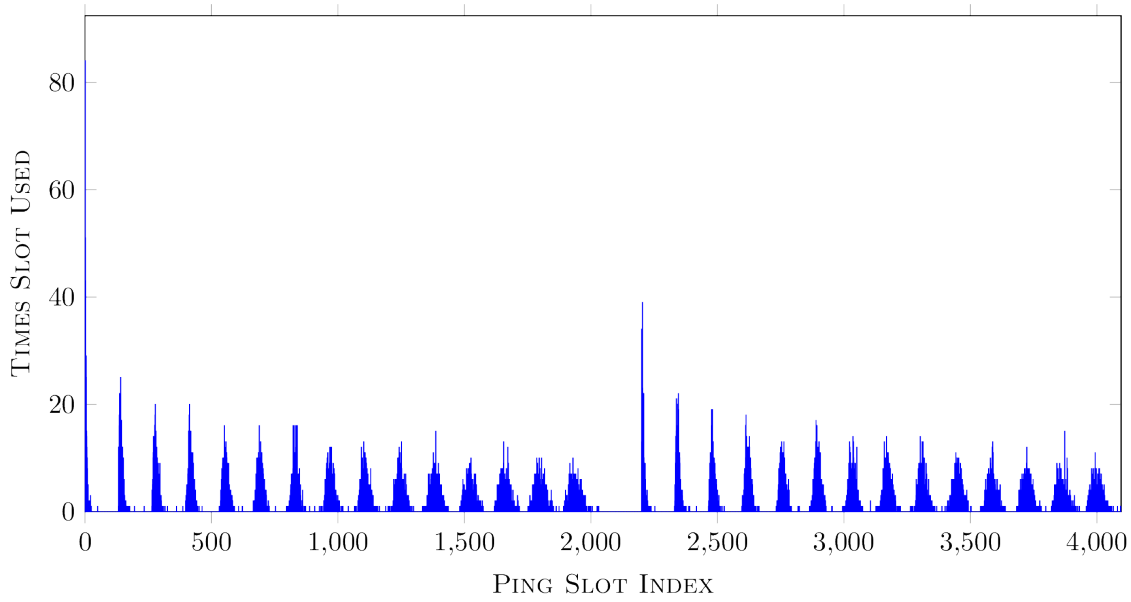
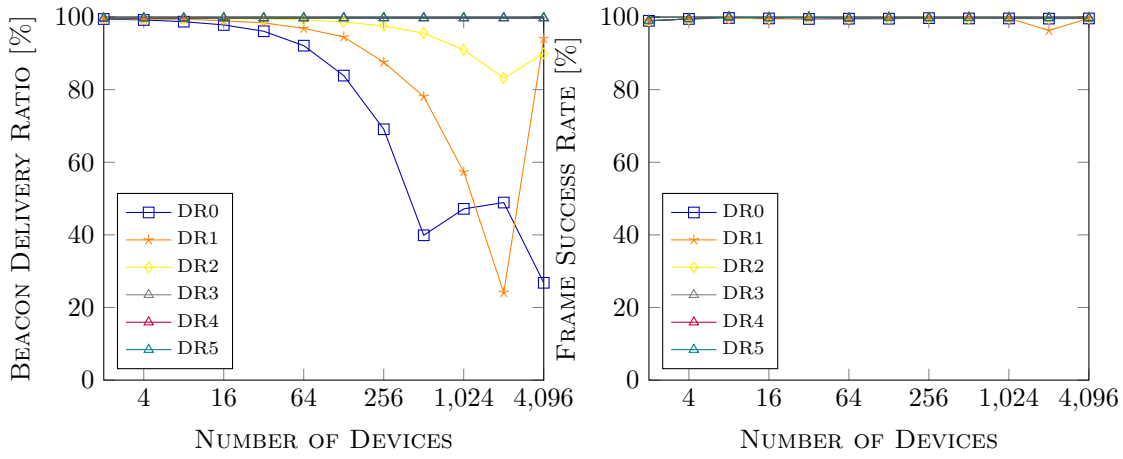


Figure 3.14: Usage of Slots in the Class B Beacon Period (512 Devices)

3.2.2.2 Transmission in Channel With 10% Limit



(a) Beacon Delivery Ratio

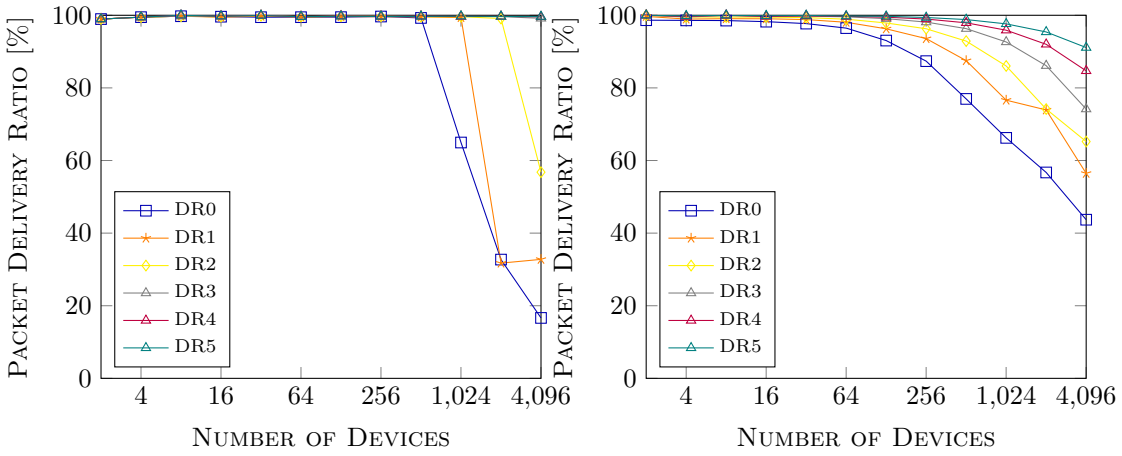
(b) Frame Success Rate

Figure 3.15: Beacon Delivery Ratio and Frame Success Rate (10% band)

Figures 3.15 and 3.16 show the case where downlink frames are transmitted in subband *54*. In this case, the single 10% channel is shared for the transmission of beacon frames and Class B downlink frames, enabling the gateway to spend a longer amount of time transmitting. Note that the dips are caused by Class A transitions performed by poor performing nodes, this is extrapolated on later in this section. When the channel is shared with the beacon frames, in larger networks (where there is regularly data queued at the gateway for transmission) the gateway may transmit a frame that causes a time-off-band period that extends to when the next beacon frame is due to be transmitted. The

beacon guard between beacon periods prevents this for the faster data rates, but is not sufficiently long enough to prevent this blocking of beacon transmissions for the slower data rates. As the network scales, the likelihood of use of a late slot increases, increasing the probability of failure to send a beacon frame (as seen in Figure 3.15 (a)). The Class B definition currently does not prevent a gateway from sending a downlink ping that will block itself from transmitting the next beacon.

Class B devices can generate the expected beacon time during the first two hours of beaconless mode, enabling the successful reception of sent downlink pings as seen in Figure 3.15 (b). However, when enough successive beacon frames are failed to be sent, Class B devices are forced to revert to Class A mode. This, coupled with the duty cycle regulations, results in the reduction of PDR as seen in Figure 3.16 (a). However, despite this, the overall PDR scales further in this case because of the greater amount of time that can be spent transmitting by the gateway. The PDR of uplink frames remains similar to the 1% band case, which is to be expected as the Class A-related parameters remain the same as in that case. Interestingly, the PDR of uplink frames is actually higher in the 4096 DR1 device case than the 2048 DR1 device case. This results in the dip visible in the figure. This is because as so many devices are forced to transition from Class B to Class A, the downlink activity of the gateway decreases enough that a higher percentage of uplink frames can be received, despite the network containing twice as many devices.



(a) Packet Delivery Ratio (Downlink)

(b) Packet Delivery Ratio (Uplink)

Figure 3.16: Packet Delivery Ratio of Uplink and Downlink Frames (10% band)

3.2.3 Findings

The downlink-focused Class B of LoRaWAN has been implemented in ns-3 in this work to enable the simulation of large scale networks that require server-initiated communication. The simulation results indicate that the primary bottleneck in the transmission of downlink frames in Class B mode is the duty cycle regulations. In addition, while the use of the *54* subband with a higher spectrum access limit does provide a greater maximum throughput, it is identified that currently in the protocol there is no mechanism to prevent gateways from transmitting packets in this subband that will block the future transmission of beacon frames. The uplink frame results are consistent

with previous Class A simulation results.

3.3 Analysis of the LoRaWAN Adaptive Data Rate Algorithm

The final part of this chapter focuses on the LoRaWAN Adaptive Data Rate algorithm. Optimal choice of the data rate for transmissions ensures that devices operate in an energy efficient manner and increases the maximum number of devices that can be managed by a single gateway. As such, the ADR algorithm is a key component of the LoRaWAN protocol which controls the performance of a LoRaWAN Network by modifying the data rate parameter of devices based on the current wireless conditions. In this section, the LoRaWAN module in ns-3 [75] is extended by adding ADR, enabling the simulation of realistic LoRaWAN networks. As LoRaWAN networks are not realistically scalable without ADR [94], any thorough simulation of the protocol requires an implementation of the algorithm. A full description of the ADR algorithm is provided in Appendix B. Briefly, the algorithm can be described as follows:

- The ADR is composed of two concurrently running algorithms; one runs on the Network Server, and the other on the device.
- The Network Server-side algorithm is not defined in the LoRaWAN protocol specification, but there is a recommended algorithm provided by Semtech. In this algorithm, the Network Server records the highest SINR value for each incoming packet and calculates the expected most suitable data rate for the device, based on the SINR value, and the current data rate and transmission power. This algorithm only increases the data rate.
- The End Device-side algorithm is defined in the LoRaWAN protocol specification. The algorithm requests a downlink frame every 32 uplink frames, and reacts to a lack of requested downlink feedback by decreasing the data rate (and thus increasing the range). This algorithm only decreases the data rate.

Implementations and analysis of ADR have previously been undertaken in LoRaWANSim [95], MATLAB [96], and OMNET++ [97]. It has been found that network convergence time increases greatly with network size, and that the current ADR algorithm is effective in stable channel conditions but not in highly-variable conditions. In addition, it has been found that the performance of the algorithm is ultimately limited in scale by the duty cycle regulations, and that in a lossy channel, the End Device-side algorithm convergence time to an optimal state is very slow. The current ADR algorithm is only recommended for use with static nodes, and indeed it is found that the performance of the ADR algorithm is inversely proportional to the mobility of the node [98]. In this section, an implementation and analysis of the ADR is performed. In the subsequent chapter, enhancements to the ADR are proposed and evaluated.

3.3.0.1 Research Proposing Alternatives to the LoRaWAN Adaptive Data Rate

There has been a significant amount of research focusing on adaptions and alternative approaches to the ADR. The current NS-side ADR algorithm allocates the data rate to a device based purely on uplink frames received from that particular device; and so, data rate allocation approaches using global network knowledge have been investigated, which overhaul the existing LoRaWAN ADR. Different methods have attempted to achieve greater scalability through aiming to achieve an equal collision probability across nodes [99], through overall throughput maximisation [47], through average system packet success probability maximisation [100], through the balancing of the link load across channels [101], and through the optimisation of the packet error rate of edge nodes [102]. In [103], linear programming models are used to optimise the transmission power and spreading factor of connected devices. The optimisation process minimises collisions in the most overloaded spreading factor, balances collisions across gateways, and finally minimises the energy consumption of devices. In [104], the optimum distribution of spreading factors in a multi-gateway LoRaWAN network which maximises network throughput is derived taking into account the capture effect. Then, an adaptive algorithm is proposed to achieve this distribution based on the adjustment of SINR thresholds.

Other approaches considered are the equal allocation of Time-on-Air to nodes (while also taking into account detected collisions of neighbouring nodes) [105], the use of message replicas to minimise the outage probability while avoiding high collision probabilities and the otherwise use of higher data rates [106], the consideration of inter-SF and co-SF interference in data rate allocation [107], and [108], and the use of mesh topologies in longer range networks to enable the avoidance of use of higher data rates [109]. In [110], a network slicing method is used to allocate network resources based on the Quality-of-Service requirements of individual slices in the LoRaWAN network. A closed expression for the outage expression of LoRaWAN with ADR is derived in [111], and used to develop an optimisation procedure to maximise the number of supported devices under reliability constraints. In [112], spreading factors are adapted based on the retransmission success rate of uplink frames, and on the sensitivity of frames at the gateway during the initial network join.

Alternative beacon-based approaches which remove the need for the ADR algorithm have also been considered. In [90] data rates are allocated to devices based on the RSSI from a previous uplink. Then, devices are grouped by their data rate, and grouped devices transmit simultaneously and are acked in one group acknowledgement. In [89] beacon frames are introduced to indicate the allowed data rates and RSSI limit of each channel for the following beacon period. This effectively groups devices by distance from the gateway, minimising the capture effect. In [113], choice of data rate is chosen based on the signal quality of received nearby beacon frames. A different approach to communications optimisation is the game-theoretical solution used to optimise the use of LoRaWAN as a fail-over communications channel for public safety networks [114].

Other approaches have extended the number of parameters considered in the link adaption, and considered different forms of traffic. These approaches include extending the available data rates to include Frequency-Shift Keying (FSK) modes [115], and extending the ADR policy to include adaption of the coding rate [116], and [117]. In [118], an adaption of the NS-side ADR is proposed to enable a higher quality estimation and conservative selection for the link. In particular, the

estimation is based on the average and standard deviation of the SINR of a larger amount of received frames. In [119], the optimal spreading factor assignment strategies for LoRaWAN communications in the context of bulk data transmissions is examined. An energy-constrained maximum throughput traffic model is considered in [120], and the message time duration of devices is equalised to balance the battery duration of LoRaWAN devices in a network, and to increase the achievable throughput of devices with better quality links. Finally, in [121], the ADR is enhanced to support mobile devices through the integration of the position and trajectory of nodes. A trilateration approach is used to calculate the device position.

Overall, analysis and adaptions of the ADR have formed a significant portion of the research dedicated to LoRaWAN. Realistic and reproducible large-scale simulation of the LoRaWAN ADR is required to fully understand the performance of the algorithm in real-world situations. The limitations of the algorithm in terms of delay and scalability have been noted by many researchers.

3.3.1 Implementation of the Adaptive Data Rate in ns-3

The simulator used in the analysis, as in the previous section, was the ns-3 LoRaWAN module introduced in [75], with the MAC layer modified to include an implementation of the ADR algorithm. The modified module has been released for public use on GitHub [5]. The module, across the stack, has been modified to support the handling of MAC commands, including the ADR-related MAC commands. The modifications to each of the components of the LoRaWAN module will now be outlined.

3.3.1.1 Modifications to the Network Server & Gateway

In order to implement the ADR, the Network Server has been modified to maintain a frame history for each connected LoRaWAN device. This frame history contains the frame counter, number of gateways received from, and SINR of each of the last twenty frames received from a device. After the arrival of every twentieth frame from a device, the NS-side ADR algorithm is run on the contents of the frame history of the device to calculate the most suitable data rate. If the Network Server determines that a data rate increase is required, a MAC command is bundled in the following downlink frame to request the device to do so. Pseudocode of the NS-side algorithm implementation is provided in Code Extract 3.7. The only required changes to the Gateway is to support the passing of SINR details from the lower layers over to the Network Server.

Code Extract 3.7: The Network Server-side Adaptive Data Rate Algorithm

```
1 LoRaWANADRAlgoritmResult LoRaWANNetworkServer::AdaptiveDataRate (
     uint deviceAddr)
2 {
3   auto ed = m_endDevices.find (deviceAddr);
4   //calculate snrMax - the max SNR over the table
5   double snrMax = -128.0;
6   for(auto & row : ed.frameSNRHistory)
7   {
```

[5] https://github.com/ConstantJoe/ns3-lorawan-adr

```
8       if(row.snrMax > snrMax) {
9         snrMax = row.snrMax;
10      }
11    }
12    double snrDr  = adrSnrRequirements[ed.m_lastDataRateIndex].snr;
13    double SNRstep = snrMax - snrDr - ed.marginDb;
14    int nStep = int(SNRstep/3);
15
16    uint8_t dataRate = ed.lastDataRateIndex;
17    uint8_t txPower = ed.lastTxPowerIndex;
18    //calculate most suitable DR and TX Power
19    while (nStep!=0)
20    {
21      if (nStep > 0) {
22        if(dataRate < 5) {
23          dataRate += 1; // increase data rate
24        } else {
25          if (txPower == 7) { // i.e if tx power is the min already
26            break;
27          }
28          txPower += 1; // i.e. drop tx power by 2dB
29        }
30        nStep -= 1;
31      } else {
32        if (txPower > 0) { // i.e. tx power is less than max
33          txPower -= 1;
34          nStep += 1;
35        } else {
36          break; // tx power is already the max
37        }
38      }
39    }
40    LoRaWANADRAlgoritmResult adrRes = {dr, tx};
41    return adrRes;
42  }
```

3.3.1.2 Modifications to the End-Device

The device has been modified to include a counter. Every time an uplink frame is sent, this counter is incremented, and the ED-side algorithm is called. The counter is reset on the reception of any downlink packet that is addressed to the device. If the counter reaches a set value ("ADR_ACK_LIMIT"), all uplink frames request a downlink response from the Network Server. If the counter reaches a further set value ("ADR_ACK_LIMIT" + "ADR_ACK_DELAY"), the data rate of the device is decremented and the counter is reset to the first set value. Pseudocode of the ED-side algorithm is provided in Code Extract 3.8.

Code Extract 3.8: The End Device-side Adaptive Data Rate Algorithm

```
1   void LoRaWANEndDeviceApplication::AdaptiveDataRate()
2   {
3     if(m_txPowerIndex > 0 | m_dataRateIndex > 0)
4     {
5         if(m_adrAckCnt == ADR_ACK_LIMIT)
6         {
7           m_adrAckReq = true;
8         }
9         else if(m_adrAckCnt == ADR_ACK_LIMIT + ADR_ACK_DELAY)
10        {
11          if (m_txPowerIndex > 0)
```

```
12          {
13            m_txPowerIndex = 0; // increase TX power
14          }
15          else if (m_dataRateIndex > 0)
16          {
17            m_dataRateIndex--; // slow data rate
18          }
19          m_adrAckCnt = ADR_ACK_LIMIT;
20        }
21
22    }
23    else
24    {
25      // link range cannot be improved
26      m_adrAckReq = false;
27    }
28  }
```

3.3.1.3 Modifications to the Network, MAC and PHY Layers

The PHY layer has been modified to handle the usage of an assigned transmission power, instead of simply the use of the maximum transmission power for the used channel. Following the ADR scheme, a lower-than-max transmission power can only be selected when the device is transmitting using the fastest data rate. The PHY layer has also been modified to pass the relevant SINR values up to the MAC layer. The only required changes to the MAC and Network layers are to handle the passing of SINR details from the lower layers up to the Application layer.

Overall, the modifications to the structure of the simulator module to implement the ADR are shown in Figure 3.17, in red.

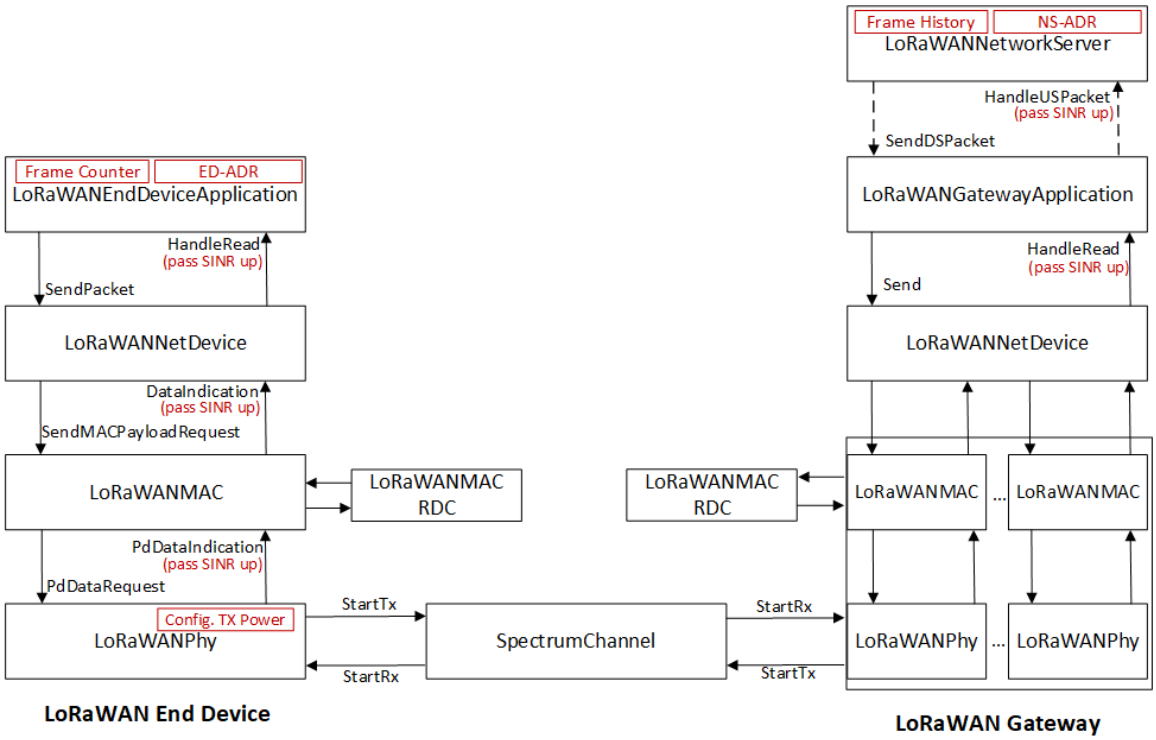


Figure 3.17: LoRaWAN ADR Integration

3.3.2 Data Rate Change Latency

The ADR consists of two concurrently running algorithms. The End-Device side (ED-side) algorithm runs locally and only slows the data rate. The Network Server-side (NS-side) algorithm runs at the network central hub and only increases the data rate. Thus, the time to change down from a data rate drx to a slower data rate dry is based on the ED-side algorithm and the transmission rate of the LoRaWAN device (t_{period}), and in the best case can be calculated as:

$$t_{drx}^{dry} = 64 * [t_{period}] + \sum_{d=dry}^{drx-1} 32 * [t_{period}] \tag{3.2}$$

which is explained as follows: In normal behaviour for a device sending unconfirmed traffic, acknowledgements are not expected to be received by the device. When the ADR is turned on, to ensure that the link is stable, after 32 uplink frames without any acknowledgement the device sets the ADRAckReq bit in the header to request downlink feedback to the subsequent uplink messages. If another 32 uplink frames are transmitted without any acknowledgement, the device concludes that the link quality is poor and drops the data rate by one. This is encapsulated in the first term of the equation. After dropping the data rate by one, the device still continues to send uplink frames with the ADRAckReq bit set. After every 32 uplink frames without any acknowledgement, the device continues dropping the data rate by one. This is encapsulated in the second term of the equation.

The latency is also dependent on the relative activity of the nearest LoRaWAN gateway, as when the device finally begins transmitting at the most suitable data rate, a downlink frame is required from the Network Server to inform the device to maintain this data rate. However, the gateway also has to adhere to the duty cycle limits and as such may not be able to respond to the first request of a downlink frame seen. The state transition diagram is thus Figure 3.18. A device can immediately transition from the lowest transmit power to the highest, and then can only transition directly from one data rate to the next slowest one.

Figure 3.18: Adaptive Data Rate State Transitions, End Device-side Algorithm

The time to change from a data rate drx to a faster data rate dry is based on the NS-side algorithm and in the best case can be calculated as:

$$t_{drx}^{dry} = 20 * [t_{period}] \tag{3.3}$$

Similarly to the other case, this latency is also dependent on the relative activity of the nearest LoRaWAN gateway, as the gateway has to adhere to the duty cycle limits and as such may not be able to send the downlink MAC commands in response to the first uplink frame seen from the device since the running of the algorithm. Note that the minimum value of t_{period} is restricted by the need to adhere to the duty cycle regulations of the EU 868 MHz band, and thus is dependent on the data rate and channel allocation of the device. The state transition diagram is thus Figure 3.19. A device can, in one state transition, change from the current data rate to any faster data rate (or DR5 with a lower transmission power combination).

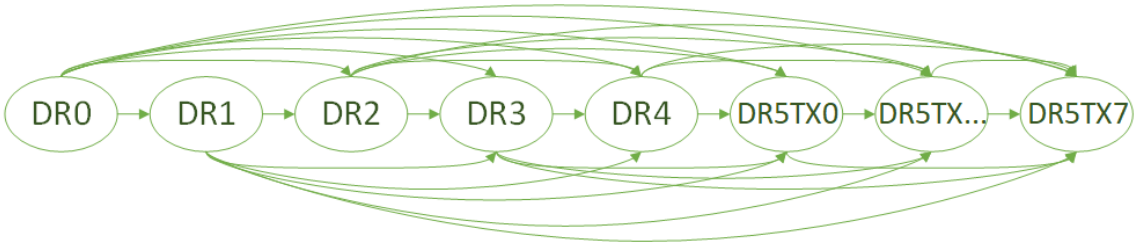


Figure 3.19: Adaptive Data Rate State Transitions, Network Server-side Algorithm

These equations describe the latency behaviour of ADR, and their effect on large-scale LoRaWAN networks is explored next through simulation.

3.3.3 Simulation and Analysis of the Adaptive Data Rate

The simulations consist of a LoRaWAN network, where N LoRaWAN Class A devices are equally distributed across a disk of radius 5 km. The key parameters in the simulations are outlined in Table 3.8. Devices transmit their first frame at a random time between 0 and *Uplink_Period*, and then transmit once every *Uplink_Period*.

Table 3.8: Simulation Parameters

Gateways	1, 4
Devices	100, 500, 1000, 2500
Disk Radius	5000 m
Uplink Period	600 s, 3600 s
Downlink Period	No data plane DL
Packet Size	8 bytes (excluding header)
Random Streams	5
Simulation Time	250 * Uplink Period, or till convergence

3.3.3.1 Effect of Starting Data Rate on Convergence Time

In each simulation, every device begins transmitting using the same data rate. Figure 3.20 shows the amount of time for every device in the network to reach a steady data rate using the ADR algorithm, for each starting data rate case. Data from simulations with a network size of 100, 500, 1000, and 2500 devices is provided. It can be seen that the convergence time for all devices to reach a steady data rate is dependent both on the starting data rate, and the number of devices in the network. As discussed in Appendix B, the ADR is composed of two separate algorithms that run concurrently; the NS-side algorithm only increases the data rate of a device and the ED-side algorithm only decreases the data rate. As such, the simulations where all devices begin using DR0 is the case where the data rate of all devices (at least initially) are manipulated by the NS-side algorithm. Similarly, the ED-side algorithm manipulates the data rate of all devices in the simulations where all devices begin using DR5.

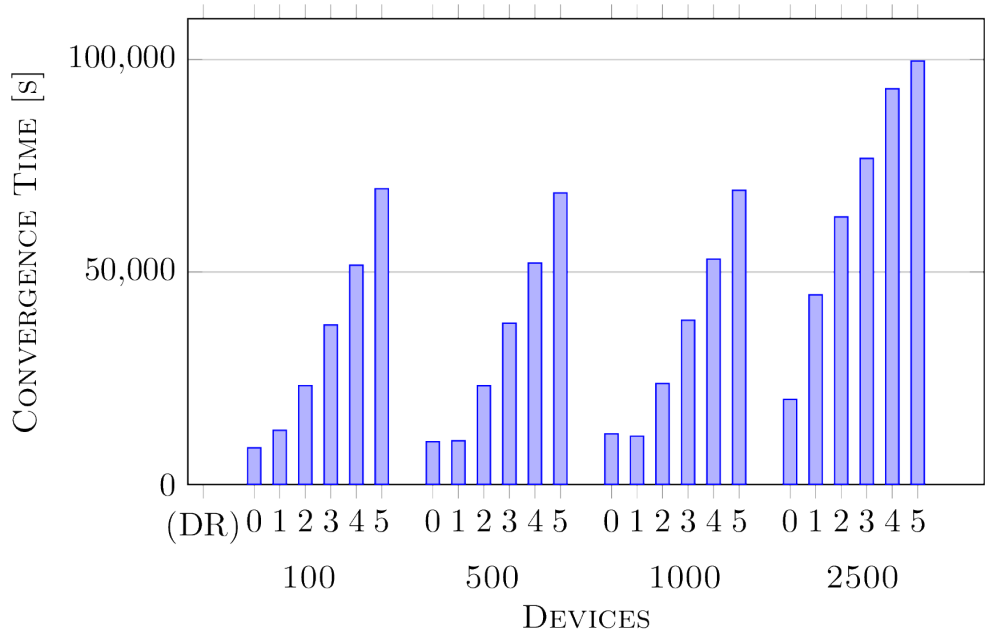


Figure 3.20: Average Node Convergence Time for Varying Starting Data Rates and Network Sizes, Uplink Period=600s

Using this information, from Figure 3.20 it can be seen that the two algorithms do not converge at the same rate; the NS-side algorithm converges much faster than the ED-side algorithm. The starting data rate of a device does have a large effect on the convergence time, with the NS-side algorithm converging to the ideal data rate much quicker than the ED-side. This is intuitive when we consider the structure of the two algorithms (see Algorithms B.1 and B.2 in Appendix B): the NS-side algorithm may modify the data rate of a device to use any faster rate; conversely, the ED-side algorithm only changes the data rate of a device one step at a time.

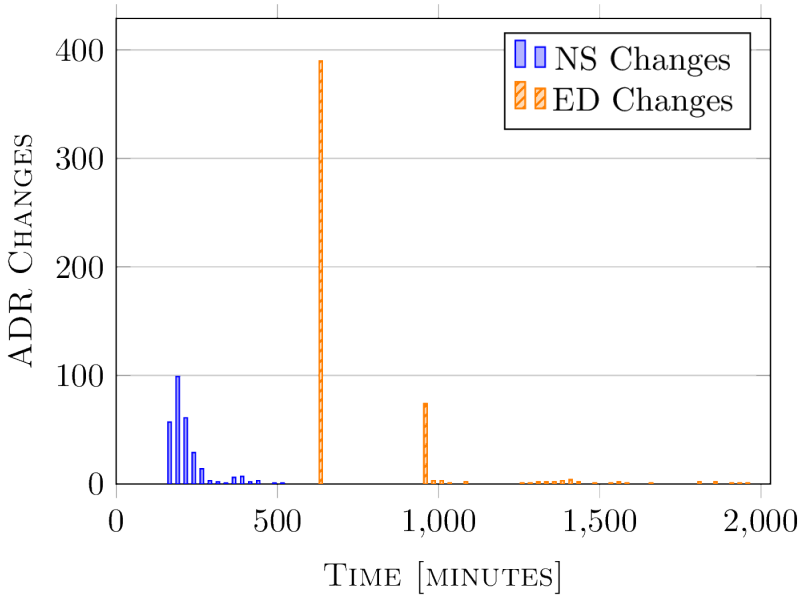


Figure 3.21: ADR Changes Across Time, Starting DR=2, Uplink Period=600s, 1000 Devices

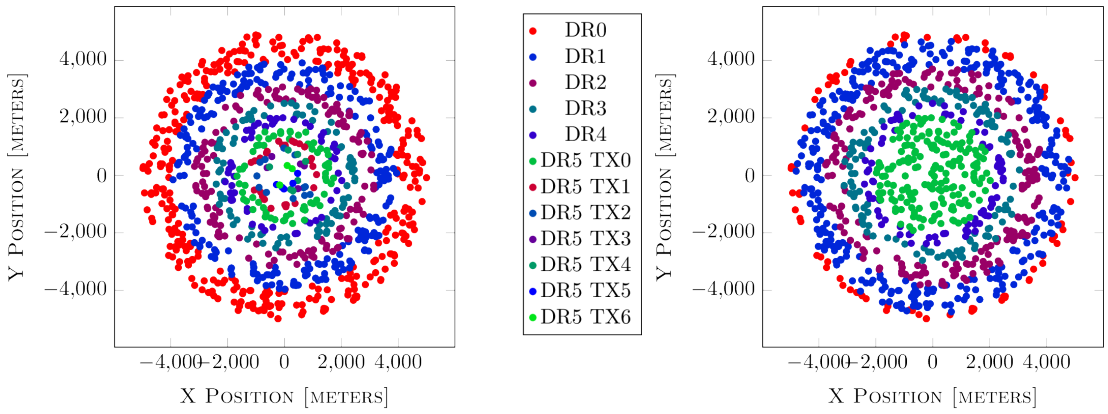
Highlighting this is Figure 3.21, which shows the number of ADR changes across time for an example 1000 device network (where all devices initially transmit using DR2). The bars in blue show the number of devices that changed data rate because of the NS-side ADR algorithm, at each point in time. The bars in orange show the number of devices that changed data rate because of the ED-side ADR algorithm. The ED-side changes occur in time in steps of $ADR\_ACK\_DELAY * uplink\_period$. Conversely, the NS-side changes begin to occur after N uplink frames are received. The choice of a particular N value is quite arbitrary and in some situations it may become clear quickly that the data rate of a device should be increased. One case in particular is when a device first joins the network: in the join procedure, any data rate can be used, and whichever data rate is used in the join procedure becomes the first set data rate of the device. So, when a device first joins the network, it is possible that the device is using a much slower data rate than required. Note that Figure 3.21 just demonstrates a representative example of one run of the algorithm i.e. one run of one simulation.

The ED-side ADR algorithm is slower to converge as it has been designed to minimise the amount, and maximise the flexibility, of control plane downlink traffic, which is limited by the duty cycle regulations applied to LoRaWAN devices, including LoRaWAN gateways. The design of the algorithm enables a greater number of devices to be handled by a single LoRaWAN gateway in the general case (when a device is currently using the most suitable data rate), and enables devices that lose all connection to the gateway to eventually re-establish reliable communication by gradually stepping up the data rate. This flexibility is at the expense of a slow convergence time in the worst case (i.e. when a node is currently using a much lower range data rate than needed to reach the nearest gateway).

Overall, in the general case the NS-side algorithm converges to the ideal data rate for a device at a faster rate. Therefore, a reasonable strategy can be to attempt to join the network using

a conservative data rate and allow the node to converge to the ideal data rate itself, rather than attempting to join the network using the fastest data rate possible and potentially ultimately joining the network using a data rate that cannot allow the node to reliably communicate with the nearest gateway.

3.3.3.2 Effect of Starting Data Rate on Final Data Rate



(a) 1000 Devices, Starting DR=0

(b) 1000 Devices, Starting DR=5

Figure 3.22: Map of Final Data Rate vs Device Location of Different Starting Data Rates

In addition to having a direct effect on the data rate convergence time of devices, the starting data rate has an effect on the final data rate of a device; i.e., the ED-side and NS-side algorithms do not necessarily converge to the same data rate assignment. Figures 3.22 (a) and (b) display a top-down view of the network, with the single LoRaWAN gateway located at the centre and the location of individual devices marked according to the final steady data rate of each device. These figures show the difference in data rate assignment for two simulations, one of which all devices begin transmitting at DR0, and the other at DR5. Table 3.9 shows the percentage of devices using each data rate at convergence for these two simulations. This difference in allocation is primarily due to the margin parameter of the NS-side algorithm, which by default sets devices to a conservative data rate allocation in order to prevent oscillation between data rates.

Table 3.9: Percentage of Devices Settling to each Data Rate, N=1000, Uplink Period=600s

	Starting Data Rate	
Ending Data Rate	**DR0**	**DR5**
0	36%	8.3%
1	23.4%	32.5%
2	14.2%	20.9%
3	8.7%	12.6%
4	5.5%	8.3%
5	12.2%	17.4%

Table 3.10: PDR and Average Current Consumption of Devices After Convergence, N=1000, Uplink Period=600s

Starting DR	**Average PDR**	**Average Curr. Cons.**
DR0	90.79%	73.74 μA
DR1	83.53%	58.60 μA
DR2	80.23%	54.32 μA
DR3	78.95%	52.25 μA
DR4	77.18%	50.96 μA
DR5	75.85%	49.05 μA

These figures show that the NS-side algorithm converges devices to a slower, higher range but less energy efficient data rate than the ED-side algorithm. In the simulation where the starting data rate for all devices is DR0, the mean data rate converged to is "DR" 1.61. For the starting data rate of DR5 case, the mean converged data rate is "DR" 2.32. Table 3.10 shows average PDR and current consumption results from a set of six similar simulations. The energy consumption results make use of the ns-3 extension previously outlined at the beginning of this chapter, and models the energy consumption of the transceiver only (assumed to be the SX1272). The average current consumption for each data rate appears low; note that this is an average across time and even in the worst case LPWAN devices spend $> 99\%$ of the time in sleep mode. The NS-side algorithm converges devices to a data rate that provides a higher PDR, but also a higher energy consumption. The poorer quality of the converged ED-side links is a result of the method used in the ED-side algorithm in maintaining links: the algorithm only requires 1 of a potential 32 uplink frames to be received and responded to; though, as previously discussed, this does enable flexibility and greater scalability, it also does not take into account the quality of the link when decided to maintain that current data rate. Therefore, the ED-side algorithm can converge devices to a data rate that provides a poor quality link, as the device cannot distinguish between a somewhat lossy link and a high quality link. Additionally, as described in the analytical work of the previous chapter and the energy model of this chapter, the data rate of a LoRaWAN device has a significant impact on the energy consumption of a LoRaWAN device, and so these results show that the data rate that a device joins the LoRaWAN network with can have a significant effect on the long-term battery lifetime of a LoRaWAN device.

As the time to transmit the same LoRaWAN packet using different data rates is $\propto 2^{12-DR}$, there is a significant difference in the energy consumption of the transmission of two frames using two different data rates. However, since confirmed LoRaWAN frames are not scalable [122], either the use of a slower-than-necessarily required data rate to maintain a higher PDR, or the sending of redundant frames (i.e. $NbTrans > 1$) is advisable instead of fully confirmed traffic. Naive approaches to this can have a cascading negative effect on the PDR of devices because of the increased probability of collisions, and is complicated by the fact that data rates are quasi-orthogonal [105]. This has led to the proposal of overhauls of the ADR to perform assignment of data rates using global knowledge, as outlined at the beginning of this section.

3.3.3.3 Issue in the Things Network Implementation

As previously mentioned, the NS-side algorithm is not explicitly defined in the LoRaWAN protocol specification. This has led to differences in the available implementations of the Network Server.

In ChirpStack, if there is a MAC layer message queued to be sent to a device, if at the time of one of the device's downlink receive windows there is no application-layer data queued to be sent to that device, a new downlink frame with no application-layer data payload is generated and the MAC layer message is piggybacked in that downlink frame. However, in the Things Network, in the same situation, a new downlink frame is *not* generated, and the MAC layer message is delayed to be sent in the next downlink frame for that device. In devices without downlink application-layer data requirements, this results in MAC layer messages only being sent in downlink frames after the *ADRAckReq* bit is set in a preceding uplink frame, as these are the only downlink frames.

For devices using DR0, this has the consequence that in applications with no downlink data requirements, no downlink frames of any kind will ever be sent, as in the current implementation of the ED-side ADR the *ADRAckReq* bit is never set for devices using DR0 (see line 5 of Algorithm B.1 in Appendix B). As a result, no MAC layer messages are ever sent, and thus no data rate change ever takes place no matter the quality of the link, limiting the energy efficiency and scalability of the network.

3.3.3.4 Effect of Multiple Gateways

Figure 3.23 shows the equivalent ADR change graph for a 2500 device simulation, also with a starting data rate of DR2. The peak of the NS-side changes becomes limited both by the duty cycle regulations that the gateway must adhere to, and simultaneously arriving ADR requests. The final node re-balancing seen in earlier equivalent graphs is extended in time much further as the probability of node transmissions consistently colliding in time and space is increased.

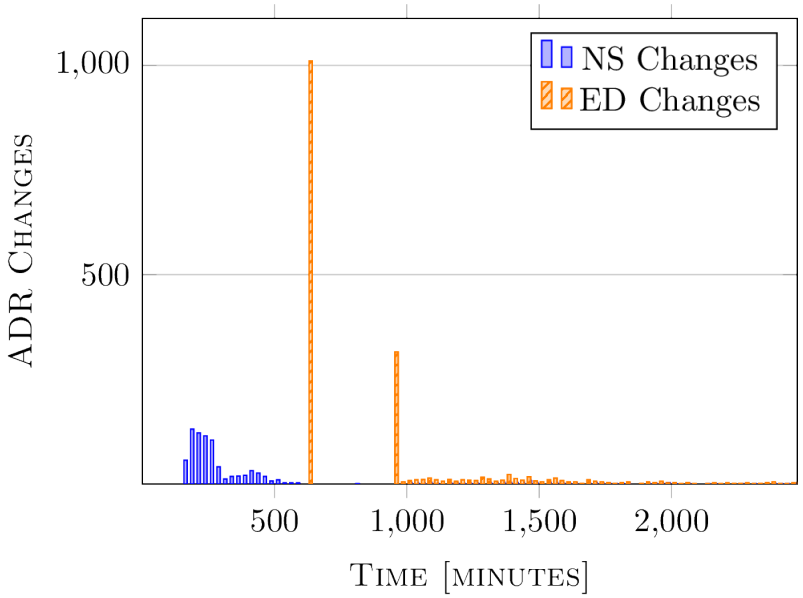


Figure 3.23: ADR Changes Across Time, Starting DR=2, Uplink Period=600s. $N = 2500$, Single Gateway

Figure 3.24 shows the equivalent graph for the multi-gateway case, where instead of one gateway located at (0,0), in the network there exists four gateways at (3000,3000), (3000,-3000), (-3000,3000), and (-3000,-3000). The allocated location for the devices is maintained from the single gateway case. Note that the increase in performance is not purely from reducing the number of devices handled per gateway, but also from enabling a greater proportion of devices to use faster data rates (which also allows more downlink frames to be sent per gateway). It can be seen that the average convergence time drops across each simulation in comparison to the single gateway case. This is because of the dual factors of 1) devices are less likely to be located far away from a gateway, so DR0 and DR1 are used less frequently, and 2) the increased number of gateways increases the amount of time the Network Server can send downlink frames, enabling ADR commands to be sent faster.

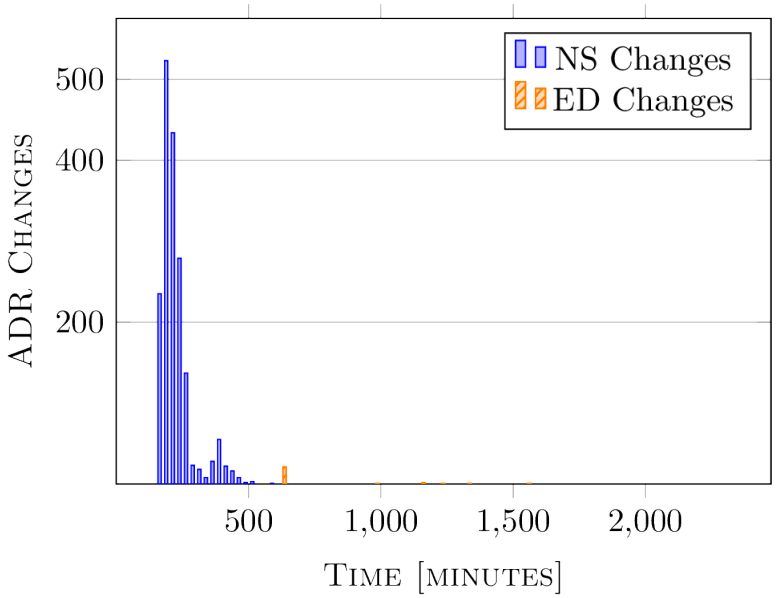
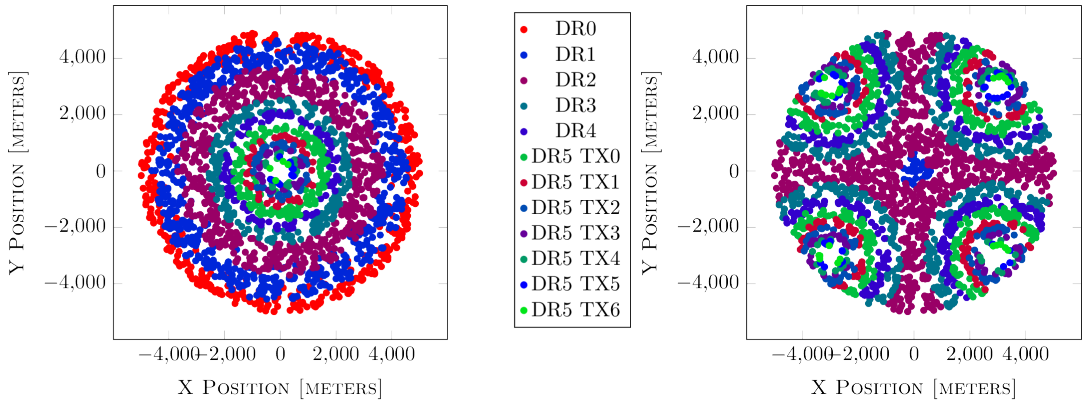


Figure 3.24: ADR Changes Across Time, Starting DR=2, Uplink Period=600s. $N = 2500$, Multiple Gateways

Figure 3.25 shows the final data rate map for the two networks, with (a) showing the single gateway case and (b) showing the multi-gateway case. As the comparison between Figures 3.25 (a) and (b) show, a greater density of gateways enables more devices to use faster data rates. This also enables the gateways to use faster data rates in response, enabling a larger amount of traffic to be sent per-gateway. The mean data rate converged to for the single gateway case is "DR" 1.89, for the multi-gateway case it is "DR" 3.42. Note that in LoRaWAN gateways act purely as relays to the Network Server, so the addition of more gateways does not complicate the management of a LoRaWAN network and thus is the simplest method for increasing the scalability of a network.



(a) Single Gateway

(b) Multiple Gateways

Figure 3.25: Map of Final Data Rate vs Device Location for the Single and Multiple Gateway Case, for a Starting Data Rate of DR2

3.3.4 Findings

Based on the simulations and analysis we can make the following conclusions about the LoRaWAN ADR:

- The ADR consists of two concurrent and interacting algorithms, which do not converge at the same rate. The ED-side ADR algorithm is slower to converge as it has been designed to minimise the amount and maximise the flexibility of control plane downlink traffic, which is limited by the duty cycle regulations applied to LoRaWAN devices (including gateways). The two algorithms do not necessarily converge to the same data rate assignment; the NS-side algorithm generally converges devices to a slower, higher range but less energy efficient data rate than the ED-side algorithm. The ED-side algorithm can potentially converge devices to a lossy link.
- The NS-side algorithm is not explicitly defined in the LoRaWAN protocol specification, and this results in inconsistent behaviour between different implementations of the NS. In the Things Network implementation, the NS-side algorithm fails to function for devices using DR0 and without any downlink data requirements.
- The overall convergence time eventually becomes extended because of the duty cycle regulations applied to the gateway. Since the gateways act purely as relays to the central Network Server the simplest solution to increasing the scalability of the network is to increase the density of gateways.

3.4 Conclusions

As has previously been discussed, LoRaWAN networks are expected to operate on a large scale, with individual gateways serving thousands of infrequently-reporting devices. In this chapter, the simulation of three main aspects of the LoRaWAN protocol were performed. Firstly, the analytical LoRaWAN energy consumption model introduced in the previous chapter is used to develop an energy model of LoRaWAN in ns-3, enabling the prediction of energy consumption of LoRaWAN devices in large scale simulations. The ns-3-based model is directly compared to results from the analytical model and show equivalent results for the same application, showing that the energy model is integrated into the ns-3 module, and enabling the analysis of large scale LoRaWAN networks in terms of energy efficiency.

Simulation results of Class B show the potential scalability of LoRaWAN networks requiring latency-intolerant downlink communications, and identify the primary bottleneck for the performance of LoRaWAN networks containing many Class B devices. Analysis of ADR provides findings on the latency and convergence properties of the system, which is dependent on the starting configuration of devices. Additionally, limitations to both concurrent ADR algorithms are identified. Overall, LoRaWAN simulation tools are extended in this chapter to enable more realistic simulation of the full protocol. For LoRaWAN energy efficiency, Class B and ADR, the research described in

this chapter constitutes the first time the feature has been simulated and analysed in ns-3. These findings stimulate investigation into an improved ADR, which forms the work described in the next chapter.

CHAPTER 4

Enhancement of the LoRaWAN Adaptive Data Rate

In the analysis presented in the previous chapter, a number of characteristics of the ADR algorithm were identified which limit the scalability and reactiveness of a LoRaWAN network. In this chapter, substantive enhancements for the device and Network Server are presented. These modifications fit within the existing mechanism and do not require any overhaul of the existing LoRaWAN protocol. Simulations show that these modifications result in a significant reduction of the data rate convergence time for LoRaWAN devices, and lead to an increased overall PDR for the network in a dynamic network environment. Overall, the contribution is a publicly available implementation of a novel version of the ADR algorithm with enhancements that improve performance in every case while remaining easily integrable into an existing LoRaWAN system.

4.1 Adaptive Data Rate Enhancements

The identified characteristics of the ADR algorithm which impact the performance of the network are as follows:

- The two concurrently-running sides of the ADR algorithm do not converge at the same rate; the Network Server-side converges faster.
- The two sides of the algorithm do not necessarily converge a device to the same data rate.
- The ED-side algorithm can potentially cause a device to converge to a poor quality link, because of the high amount of flexibility provided to the Network Server in reacting to uplink messages which request a downlink response.

- The Network Server-side algorithm is not explicitly defined in the LoRaWAN protocol specification, which results in some cases where LoRaWAN devices fail to converge to a suitable data rate.

Based on these characteristics, enhancements for the ED-side and NS-side ADR algorithms can be identified.

4.1.1 End Device-side Adaptive Data Rate Enhancement

The main identified issue with the ED-side algorithm is the convergence of devices to lossy links; however the Network is actually able to calculate the PDR of devices by analysing the frame numbers included in the LoRaWANFrameHeader. It is proposed that the Network Server takes into account the recent quality of the link when responding to the ADRAckReq from the device. The modified mechanism is displayed in Code Extract 4.1. The modifications are made purely on the Network Server. Prior to sending a downlink frame in response to a set *ADRAckReq* bit, the Network Server calculates the PDR of the device since the device last changed data rate (line number 28). If the PDR is below a threshold value, a *LinkADRReq* MAC layer message is sent to the device to request the use of the next slowest data rate (line number 30). In our simulations, a threshold value of 80 % is chosen. This represents a plausible minimum rate of successful data transfer for a LoRaWAN network with a range of up to 5 km [123]. Note the parameter *totalPacketsSent* is the total number of packets sent by the device since the last change of the data rate. Since the last ADR message had to have been sent in response to a received frame, the frame counter of that received frame will be known by the Network Server. Similarly, since this downlink frame is being sent in response to an uplink frame, the frame counter of that frame will also be known. The PDR calculation in the algorithm is based on the difference between the values of those frame counters, compared with the actual number of frames that were received by the Network Server in the same timeframe. In this way, the PDR of recent frames can be calculated.

Code Extract 4.1: The Modified End Device-side Adaptive Data Rate Algorithm Mechanism

```
1
2  void LoRaWANNetworkServer::HandleReceive(Packet p)
3  {
4      ... // all of the previous HandleReceive code
5      uint deviceAddr = p.GetAddr();
6      auto ed = m_endDevices.find (deviceAddr);
7      if(ed.currentDataRate != p.GetDataRate())
8      {
9          ed.initialFrameCounterSinceDRChange = p.GetFrameCounter();
10         ed.packetsSinceDRChange = 1;
11     }
12     else
13     {
14         ed.packetsSinceDRChange++;
15     }
16     if(p.GetADRAckReqSet()) // handle the MAC command
17     {
18         OnADRAckReqSet(deviceAddr, p);
19         ed.ScheduleDownlink();
20     }
21 }
```

```
22
23 void LoRaWANNetworkServer::OnADRAckReqSet(uint deviceAddr, Packet p)
24 {
25   auto ed = m_endDevices.find (deviceAddr);
26   int totalPacketsSent = p.GetFrameCounter() - ed.
        initialFrameCounterSinceDRChange;
27
28   float recentPDR = ed.packetsSinceDRChange / totalPacketsSent;
29
30   if(recentPDR < pdrThreshold)
31   {
32       ed.currentDLPacket.addMACLayerMessage(LinkADRReq, this.
            currentDR - 1);
33   }
34 }
```

4.1.2 Network Server-side Adaptive Data Rate Enhancement

There are two issues with the NS-side algorithm: 1) the frequency of the running of the algorithm is quite arbitrary (run after N uplink frames are received, where N is usually 20), and 2) the Things Network case mentioned in the previous chapter, where the algorithm fails to run for devices using DR0. It is proposed to introduce a mechanism that enables the timing of the NS-side algorithm based on the SINR values of recently received frames. The pseudocode for this new mechanism is provided in Code Extract 4.2. In particular, an early call of the NS-side ADR algorithm is performed if three factors are true (line number 15):

1. at least 5 uplink frames have been received since the last data rate change, and
2. the received SINR values from the previous uplinks will ultimately cause a change of the data rate of the device during the NS-side ADR algorithm, and
3. the standard deviation across those SINR values is less than 2.5 dB (the difference in signal strength between LoRaWAN data rates).

The first factor is a trade-off between responsiveness and accuracy, and a value of 5 was chosen as a heuristic element as the minimum reasonable value which enables responsiveness without having too little samples for the calculation of the standard deviation. The second factor is included as, without it, there would be no need to calculate the ideal data rate as the device has already converged to it. The third point requires all of the received samples to be relatively similar in value, and so can be considered to requiring the algorithm to converging the device to a particularly suitable data rate.

In addition, after this early fire of the ADR algorithm has been run, if there is no currently scheduled downlink frame for the device, then it is proposed that in this case the Network Server is specifically allowed to create a new empty downlink frame for the MAC command to be sent in, in the next open receive window of that device.

Code Extract 4.2: The Modified Network Server-side Adaptive Data Rate Algorithm Mechanism

```
1
```

```
void LoRaWANNetworkServer::HandleReceive(Packet p)
{
    ... // all of the previous HandleReceive code
    uint deviceAddr = p.GetAddr();
    auto ed = m_endDevices.find (deviceAddr);
    ed.frameSNRHistory.Add(deviceAddr, p.snr);

    // this is the existing ADR call code
    if(ed.UplinksSinceADR == ADR_FREQUENCY) \\ i.e. 20
    {
        AdaptiveDataRate(deviceAddr, scheduleNewDL=false);
    }
    // this is the new early fire mechanism
    else if(ed.UplinksSinceADR >= 5 && StdDevAcrossSnrs(ed.
        frameSNRHistory) < 2.5 && PredictedDRChange(ed.
        frameSNRHistory) != 0)
    {
        AdaptiveDataRate(deviceAddr, scheduleNewDL=true);
    }
}
```

Figure 4.1 shows the difference in the potential data rate change latency for the original and the enhanced versions of the ADR. The enhancements modify the data rate change latency for LoRaWAN devices, which were previously described in Equations 3.2 and 3.3. In the enhanced ADR, the NS-side algorithm, if required, can cause a data rate change after a minimum of five transmit cycles, and the algorithm will run every twenty transmit cycles in any case. For the ED-side algorithm, the Network Server can potentially trigger a data rate change after thirty-two frames, if required. This does require a successful uplink-downlink transaction over the link; if there is heavy packet loss the device can trigger the data rate change itself after 64 frames, which is the same as in the original algorithm. Overall, the enhanced ADR will take advantage of opportunities to improve the link quality faster than the original ADR, and in the worst case will function exactly the same as the original algorithm.

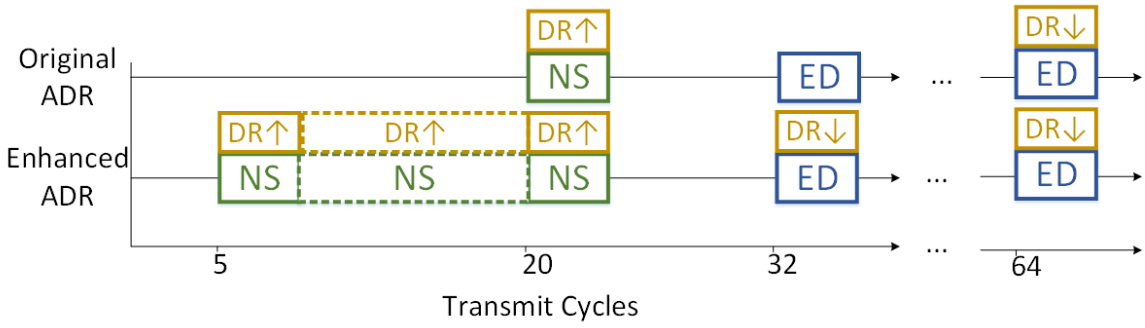


Figure 4.1: ADR Enhancements in terms of Data Rate Change Latency

As indicated in Figure 4.2 in purple, all of the required changes for both mechanisms take place on the Network Server.

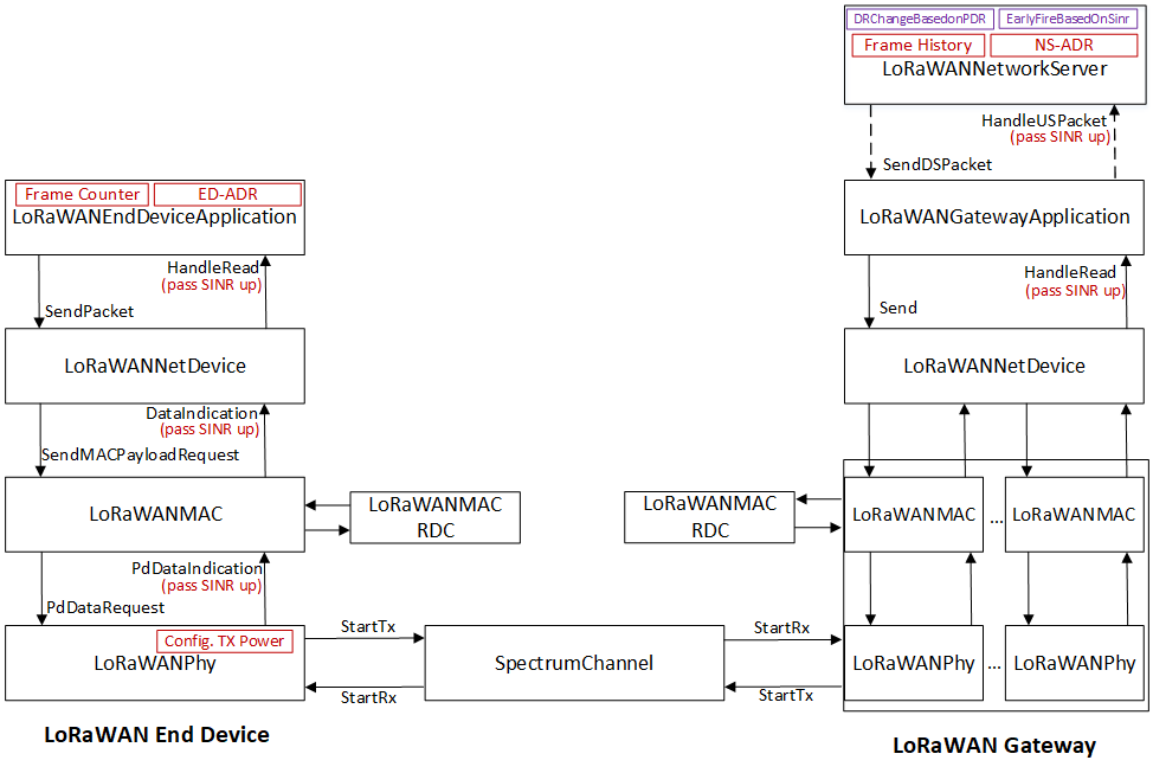


Figure 4.2: Integration of LoRaWAN ADR Enhancements

4.2 Simulation and Evaluation of Enhancements

In this section, simulations of the ADR with the proposed enhancements are performed and compared with the original scheme. All key parameters remain the same as detailed in the simulations in the previous chapter.

4.2.1 Effect of Enhancements on the Single Gateway Case

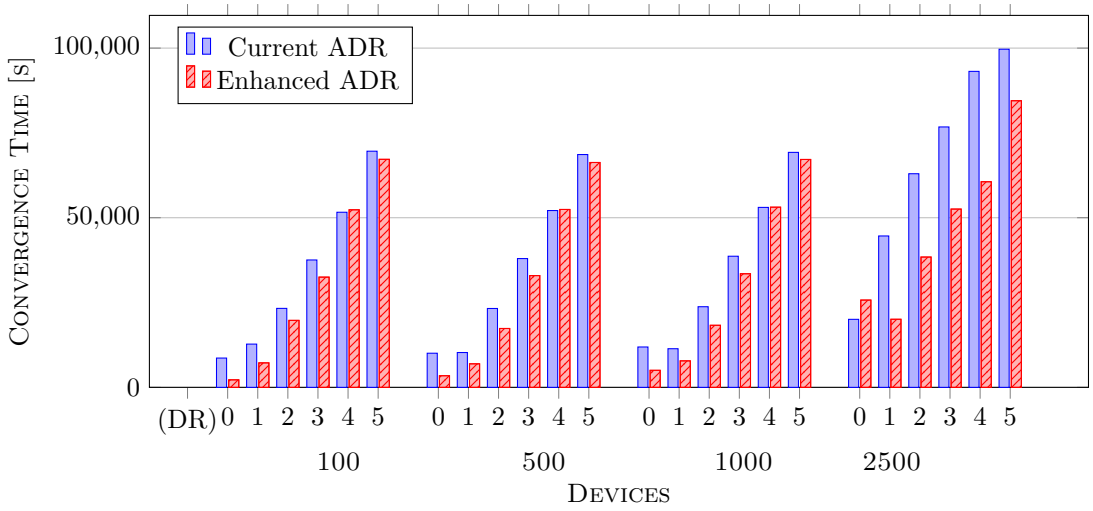
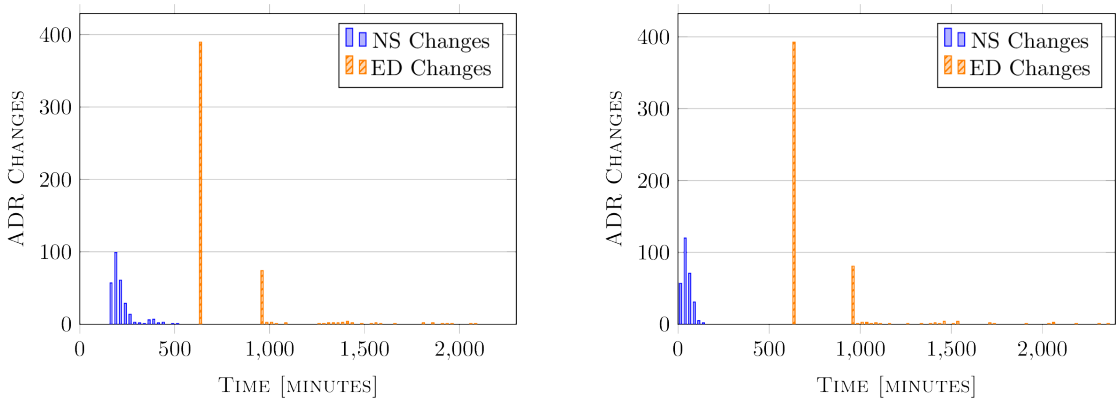


Figure 4.3: Average Node Convergence Time, Uplink Period=600s, Original ADR vs. Enhanced ADR

The proposed NS-side enhancement results in a faster convergence time for devices increasing the data rate. As Figure 4.3 shows, in networks with a small number of devices, an improvement of $> 50\%$ can be seen in the case where the majority of devices would be directly affected by this modification (i.e. the DR0 case). As expected, the benefit of the proposed enhancement is decreased in networks where fewer devices would be affected by the NS-side algorithm (i.e. the DR5 case), but there is an improvement seen in almost all cases. The performance in terms of the convergence time remains the same for smaller networks, where the gateway on-air time is not at the maximum allowed by the duty cycle regulations. As the network scales, there is an increasing benefit as the more flexible NS-side algorithm reduces the delay for a device to transition to a faster data rate, enabling the transmission of more downlink frames. In addition, the ED-side algorithm modification results in devices gravitating away from lossy links faster. A visualisation of the change caused by the enhancements can be seen in Figures 4.4 (a) and (b). The NS-side changes occur earlier, while the ED-side changes happen at the same rate as before.



(a) Original ADR

(b) Enhanced ADR

Figure 4.4: ADR Changes Across Time, for the Original and Enhanced ADR

In addition, the new ED-side enhancement converges devices to better quality links. Table 4.1 shows the equivalent PDR values after convergence for the simulations shown in Figure 4.3; the use of the new link quality-aware mechanism results in a higher average PDR in all cases.

Table 4.1: Comparison of Average Convergence Time and PDR After Convergence of Original and Enhanced ADR Schemes

	Original	**Enhanced**	**Improvement**
Nodes	**100**		
DR0	99.02%	99.05%	+0.03%
DR2	87.95%	98.01%	+10.06%
DR3	85.92%	95.82%	+9.9%
DR5	80.77%	95.87%	+15.1%
	500		
DR0	95.44%	95.59%	+0.15%
DR2	84.22%	94.08%	+9.86%
DR3	82.19%	93.01%	+10.82%
DR5	77.63%	93.97%	+16.34%
	1000		
DR0	90.60%	91.05%	+0.45%
DR2	78.63%	89.60%	+10.97%
DR3	77.53%	88.87%	+11.34%
DR5	74.61%	92.87%	+18.26%
	2500		
DR0	79.29%	79.56%	+0.27%
DR2	78.28%	78.72%	+0.44%
DR3	75.09%	79.97%	+4.88%
DR5	75.60%	79.57%	+3.97%

4.2.2 Effect of Enhancements on the Things Network Issue

The ED-side enhancement also fixes the issue discussed in the Things Network implementation, where devices using DR0 are never affected by the NS-side ADR algorithm. Table 4.2 shows results

from a set of simulations where the Network Server follows the approach from the Things Network (and not ChirpStack, which is followed in all other sets of results). The results show that in the DR0 case, devices can now converge to a stable data rate (i.e. the convergence time is not infinite).

Table 4.2: Average Node Convergence Time, Uplink Period=600s, N=100, Things Network Implementation Case

Data Rate	Current ADR	Proposed ADR
DR0	∞ **s**	**2233.05 s**
DR2	26056.63 s	20958.10 s
DR3	38749.01 s	36031.38 s
DR5	69608.57 s	69349.56 s

4.2.3 Effect of Enhancements on the Introduction of New Nodes

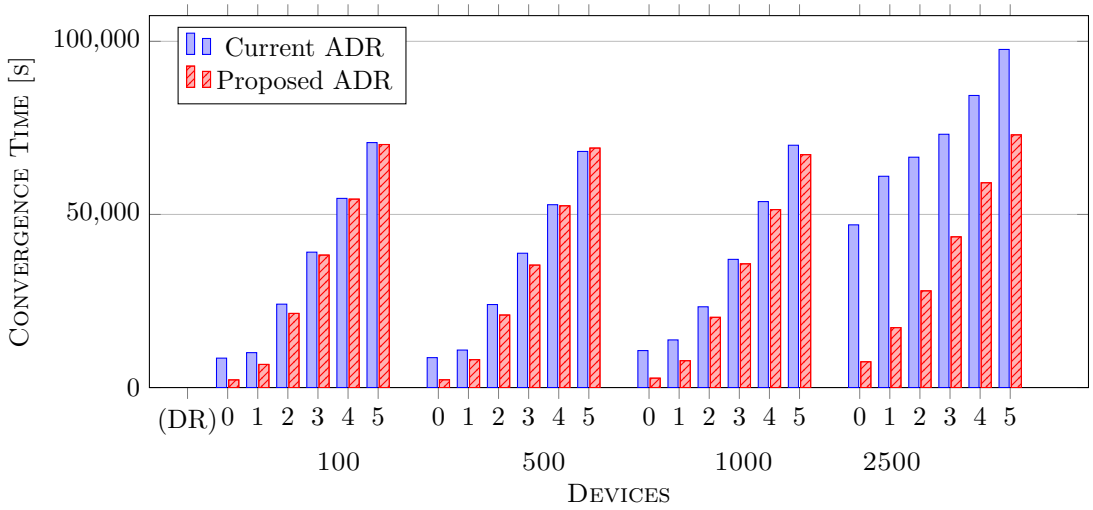


Figure 4.5: Average Node Convergence Time of 100 Newly Introduced Devices, Uplink Period=600s

Figure 4.5 shows the average node convergence time of one hundred newly introduced devices into a network of N already converged devices. In smaller networks, the presence of other devices has some small effect because of collisions. There is a greater effect in larger networks as the gateway begins to reach the duty cycle limit. The proposed enhancements continue to result in a decreased convergence time even in networks already containing many devices.

4.2.4 Effect of Enhancements on the Multiple Gateway Case

Figure 4.6 shows that the proposed enhancements have increasing gains in the multi-gateway case. The multi-gateway configuration in this case is the same as the multi-gateway setup presented in the previous chapter. For the original ADR, the existence of multiple gateways in the same sized area as in the single-gateway case results in a much higher proportional use of faster data rates, as devices

are on average closer to the nearest available gateway. As more devices on average are using faster data rates, and the NS-side algorithm converges faster than the ED-side algorithm, for the original ADR the convergence time for the network decreases with a higher density of gateways. With the enhancements, the convergence time decreases even further as the NS-side enhancement enables the faster allocation of fast data rates to devices. There is a recognisable improvement in all cases.

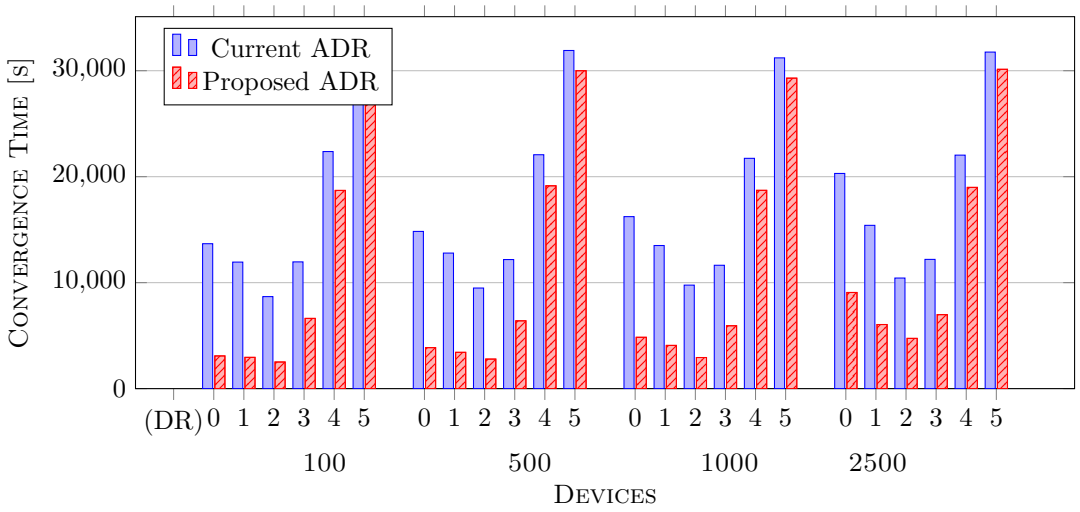


Figure 4.6: Average Node Convergence Time, Uplink Period=600s, Multi-Gateway Case

4.3 Conclusions

In the previous chapter, an extension to a LoRaWAN ns-3 module has was developed to implement ADR, and using this a number of conclusions about the scheme were made. Based on this analysis, modifications for the NS-side and ED-side of the ADR scheme have been proposed in this chapter which are easily integrable into the existing LoRaWAN protocol.

The ED-side algorithm has been enhanced to enable the Network Server to use available link quality estimates to judge whether the data rate for a poor quality link should be decremented. These instructions are piggybacked in a MAC command in the regular ADR downlink frame for a device. This mechanism prevents a LoRaWAN device from settling to a poor quality link. The NS-side algorithm has been enhanced to initiate an early iteration of the previously defined data rate calculation algorithm based on the consistency of SINR value of received frames, instead of an arbitrary set frequency. An early fire of the algorithm also enables the Network Server to specifically create a downlink frame to transmit the MAC command, instead of piggybacking the command on the next available downlink transmission.

These enhancements were implemented in ns-3, and evaluation through simulation shows that the proposed enhancements lead to a faster convergence rate for devices, and the convergence to data rates that result in a higher overall PDR for the network. In comparison to the research outlined in the previous chapter, these enhancements are easily integrated into the LoRaWAN protocol,

and do not require any beaconing mechanism or additional downlink feedback from the LoRaWAN gateways. As well as improving the PDR, these enhancements also focus on the optimisation of the convergence time for devices, which is not a focus in the related work. In the simulations, there is a consistent improvement seen in all cases. These enhancements improve the flexibility of the Network Server, enabling devices to reach the optimal data rate faster.

The response from the LoRaWAN ADR algorithm to an increased probability of collisions is to reduce the data rate of devices (via the ED-side algorithm). However, this can have the opposite effect of what is intended, as the increased time-on-air of subsequent transmissions further increases the probability of collisions. Ideally, in order to maximise the efficient use of spectrum available, devices should be transmitting at the fastest possible available data rate, but the fundamental issue is the limited amount of information available to a device to enable it to make the informed choice of data rate. However, a transition of data rate is not the only potential way of avoiding collisions; the scheduling of transmissions in time would also reduce this probability. In this case, the device will not be avoiding collisions through a transition of pseudo-orthogonal channel (i.e. the data rate), but instead through a transition in time. This underlies the fundamentals of TDMA. While the duty cycle regulations limit the applicability of a standard TDMA scheme for LoRaWAN, mMTC traffic also has inherent properties which may enable a centralised network controller to predict and ultimately control network traffic. These factors motivate the work presented in the next chapter.

CHAPTER 5

Lightweight Timeslot Scheduling Through Periodicity Detection for Increased Scalability of LoRaWAN

As introduced in Chapter 1, the majority of LoRaWAN communications fall under the paradigm of mMTC, where networks feature traffic that is sent by a huge number of low cost, low power, infrequently transmitting devices. The LoRaWAN protocol is ALOHA-based, which enables low power operation but does not include any feature to mitigate the increasing effect of collisions as the network scales. As the research review of this chapter will show, TDMA over LoRaWAN has become a major focus of research in order to increase scalability. The research review shows that a direct implementation of classic TDMA is not feasible for LoRaWAN while maintaining the low power operation enabled through use of ALOHA without carrier sense. Additionally, the restrictions on downlink traffic because of use of the EU 868 MHz ISM band limit the amount of control plane traffic that can be sent by LoRaWAN gateways. Approaches to TDMA over LoRaWAN need to consider these restrictions in order to enable operation at a large scale.

In this chapter, a novel, lightweight timeslot scheduling scheme is introduced that supports the requirements for a subclass of mMTC on LoRaWAN networks. The scheme is based on the traffic periodicity characteristic of mMTC and the multiple channels and quasi-orthogonal data rates of LoRaWAN. The approach is particularly suited for the ALOHA-based LoRaWAN, as extended downlink transmissions from the gateways are not required, and neither is tight time synchronisation between the devices and the LoRaWAN Network Server. The scheme is implemented in the publicly available LoRaWAN ns-3 module [75] that was also used in the previous chapter. Results show that the scheme doubles the number of frequently transmitting devices that can be handled by a single LoRaWAN gateway while providing the same level of performance in terms of successful packet deliveries, and maintaining a reasonable delay for mMTC use cases, without impacting the ability of

the network to send downlink frames or acknowledge high priority packets. Overall, the contribution is a novel approach to TDMA that is suited for periodic traffic over LoRaWAN networks.

5.1 Research Implementing TDMA Over LoRa

ALOHA-based access without collision avoidance provisions enables low power networks with little coordination needed between the devices and the Network Server. However, such networks have inherent limitations in regard to scalability, as the probability of collisions greatly increases with the number of devices in the network. In [124], it is shown that the coverage probability decays exponentially with the expected number of devices, though in reality the decay is not this extreme as this work does not model the LoRa capture effect. In [125], a stochastic geometry-based model is used to compare the performance of pure ALOHA, slotted ALOHA, and Carrier Sense Multiple Access (CSMA) over LoRaWAN. The results show that the capture effect of LoRa has a significant effect in increasing the performance of each of the schemes at scale. It is shown that slotted ALOHA in general outperforms regular ALOHA, and that CSMA outperforms slotted ALOHA, but only when devices are located close to each other.

TDMA over LoRa was suggested as an area for future research in the early stages of investigation into the protocol [126], particularly in regards to decentralised schedulers that take into account the expected traffic patterns of applications using LoRaWAN [127]. However, because of the duty cycle regulations of the EU 868 MHz band, conventional approaches to TDMA cannot be directly implemented in a LoRa-based system. This has led to a significant amount of research in enabling TDMA over LoRaWAN with minimal control plane traffic. In [128], an approach to timeslotting with minimal overhead is proposed. Timing is split into slots based on the maximum packet length, and devices autonomously determine slot positions using a hash function on the device address assigned during the network join. Additionally, acknowledgements are grouped and sent with time synchronisation information to minimise downlink feedback requirements. TDMA and a latency-tolerant packet-aggregating access scheme are implemented in LoRaWAN in [129] in order to increase efficient channel utilisation. In [130] and [131], wake-up radios and a cluster-based topology with LoRaWAN are combined in order to introduce on-demand TDMA, enabling the Network Server to directly request data from individual devices and thus organise scheduling to prevent collisions. A meshed multi-hop protocol for LoRa is proposed in [132] that is suitable for underground networks, in particular for urban drainage. In [133], LoRa-NB-IoT gateways are used to enable machine vibration monitoring applications with a high precision time synchronisation algorithm.

In [134], a MAC layer for LoRa is proposed to enable the support of real-time industrial periodic traffic. The approach taken is similar to IEEE 802.15.4 TSCH, where time is split into beacons, contention-free, contention-access, downlink and acknowledgements sections. During the contention-free period, packets are transmitted after a random delay. During the contention-aware period, timeslots are assigned to devices offline. Results show a benefit over LoRaWAN in terms of scalability and reliability. However, the undefined method for timeslot allocation is a limiting factor of the work. This work is extended in [113] to add support for mobile nodes, channel hopping in the contention-free

period, and slotted ALOHA in the contention-aware period. Additionally, three Quality-of-Service classes are defined to enable a trade-off between reliability and energy consumption.

In [89], a lightweight TDMA for LoRaWAN is proposed. Gateways schedule transmission power and spreading factors on each channel, and devices identify their own parameters in order to group themselves. Within a channel, a random slot is chosen for transmission. In [135], a beacon-less scheduling mechanism for periodic LoRaWAN traffic is proposed. Timeslots for devices are assigned by the Network Server in a synchronisation process based on the uplink traffic rate, clock drift accuracy, and resynchronisation periodicity, which is provided in a synchronisation request frame. Assigned timeslots are provided to the device using a Bloom filter to reduce the packet length. In [136], the reduced latency requirements of mMTC traffic are utilised to delay LoRaWAN transmissions into pre-scheduled timeslots. Beyond reducing collisions, this also enables reduced overhead as packets can be aggregated together. Support for group acknowledgements also reduces downlink feedback requirements. Simulated results show an improvement in terms of energy efficiency and packet delivery ratio. Global Positioning System (GPS) and ultra-wideband location systems are used in [137] to achieve time synchronisation in LoRaWAN devices. Then, a TDMA approach over LoRaWAN without synchronisation beacons is proposed and evaluated on hardware.

In [90], TDMA over LoRaWAN is implemented with group acknowledgements to minimise downlink transmissions. Devices regularly receive beacon frames to maintain time synchronisation, and acknowledgements of frames are aggregated and transmitted on a per-data rate basis. However, it should be noted that in this work use of the US 902-928 MHz ISM band is assumed, which does not have the same regulations as the EU 868 MHz band. A TDMA MAC protocol for LoRa is proposed in [138] that is suitable for networks where the LoRa gateways are energy-constrained and forward data on to low earth orbit satellites. The satellite allocates slots in time to individual gateways, and the gateways in turn allocate slots to connected nodes. TDMA over LoRa is proposed in [139] to enable real-time communications in industrial networks. The system also includes a data rate and channel assignment strategy to reduce network interference. Listen-before-talk is also integrated to reduce interference from external sources. In [140], a TDMA over LoRa approach is proposed, where devices provide an outline of the expected traffic when joining the network. The gateway then uses this to allocate a data rate and suitable timeslots to a device. Devices also wake periodically to receive a synchronisation beacon in order to handle clock drift. In [141], a timeslotted channel hopping mechanism is implemented on top of LoRaWAN, to enable the coverage of real-time applications in industrial settings.

Overall, it is clear to see that alternative approaches to LoRaWAN network access, in particular TDMA approaches, is an active area of research. Approaches which are particularly suited to LoRaWAN systems, taking into account the traffic format, semi-orthogonal spreading factors, centralised network hub, and limitations on downlink feedback are particularly required.

5.1.1 Consideration of Traffic Patterns in MAC layer design

More advanced traffic modelling of MTC than the 3GPP model has been performed by [142], which models individual devices using Markov Modulated Poisson Processes, and geographically co-located event-based through interactions between the processes. [143] evaluate the performance of LoRaWAN using the previously mentioned model which combines periodic and event-based traffic, and show that LoRaWAN is unable to handle significant temporally correlated event-based traffic. [144] analyses LoRaWAN performance under different traffic models based on different IoT use cases, and demonstrates the limitation of scalability for the slowest LoRaWAN data rate. [145] also extend the 3GPP model by integrating an open telecommunications dataset into the MTC traffic model. A comparison of Machine-to-Machine (M2M) traffic models against real-world data sets is performed by [146], who do find that the majority of M2M traffic is deterministic and periodic. An analysis of M2M traffic on cellular traffic is performed by [147]. [148] study the loss rate and delay of non-Poisson M2M traffic in LTE networks. In [44], the characteristics of aggregated periodic IoT data from related work is analysed and compared to a Poisson process as an approximation for the traffic. Finally, [149] propose a model of MTC traffic at the central controlling node of a Wireless Sensor Network, and study packet loss and delay for varying arrival rates.

Consideration of the traffic patterns of mMTC has previously been used in order to enable greater network performance. In [150], categorisation of traffic of a generic LPWAN device is performed to enable reduced latency for urgent and event-driven traffic, by providing one dedicated contention-based channel for event-based and urgent packets, and another dedicated contention-free channel for periodic traffic. In [151], the researchers focus on enabling a reduced latency for event-based traffic in IEEE 802.11ah. This is achieved through the use of a pool of reserved slots which changes in size based on recent reporting demands. Thus, mMTC traffic patterns are considered to enable greater performance, but gains are made through consideration of event-based traffic, not periodic. [152] provide a scheduling method for fast uplink grant transmissions in a cellular system for MTC based on multi-arm bandits. Other research has focused on the graceful support of heterogeneous traffic policies in a single network. [153] introduce and analyse scheduling policies for heterogeneous traffic over LTE, including MTC traffic. In [154], time allocation for a network of heterogeneous devices is formulated as a non-cooperative game, factoring in heterogeneous requirements and capabilities of devices in the network. [67] proposes a system that monitors traffic patterns from networks consisting of devices with multiple wireless access options, and modifies the selected wireless technology of a device to the most suitable option based on the analysis.

Overall, consideration of the traffic format is key in the analysis of realistic LoRaWAN networks. mMTC has considerably different features to HTC; examination and reflection on those features can enable the design of MAC layer features that are particularly suitable for LoRaWAN networks.

5.2 Lightweight Timeslot Scheduling (LTS)

The transmission time of individual LoRaWAN devices (including LoRaWAN gateways) is limited by the duty cycle regulations of the sub-1GHz ISM bands. As a result, LoRaWAN is not suitable for applications with strong real-time or low latency requirements, and downlink traffic and confirmed uplink frames are limited. In addition, this means that control plane traffic, used in the management of a LoRaWAN network, is limited. LoRaWAN manages this limitation through the use of an ALOHA-based channel access method, which does not organise any slots in frequency or time for devices in the network, and thus requires minimal downlink control plane traffic for connected devices.

However, as previously described in Appendix B, mMTC traffic transmitted over LoRaWAN can generally be described as either periodic or event-driven. In this work, the periodicity of LoRaWAN traffic is leveraged to increase the scalability of the LoRaWAN network. As LoRaWAN is already unsuitable for applications with strict low latency requirements, an increased delay to a certain degree is not a major concern. In addition, as LoRaWAN features a minimum of 3 separate channels, and 6 quasi-orthogonal data rates, co-located devices that are programmed to continuously transmit simultaneously will not always interfere with each other. These factors enable the detection of conflicting periodic traffic in LoRaWAN networks.

The scheme in this work uses the LoRaWAN Network Server to learn the characteristics of incoming traffic from individual devices, uses these to make predictions about future network traffic arriving at each gateway in the network, and ultimately reduce expected collisions from periodic traffic of geographically co-located devices. A system is developed that is suited to the characteristics of mMTC and LoRaWAN (in particular, periodicity, and LoRaWAN multiple channels and quasi-orthogonal data rates), which enables lightweight timeslot scheduling that increases scalability without introducing strain on downlink feedback from LoRaWAN gateways. We have named our system Lightweight Timeslot Scheduling (LTS). The system allocates timeslots automatically, and does not require control plane interaction between the Network Server and devices in order to learn per-device traffic patterns. This makes the proposed system particularly suited for LoRaWAN networks, as the energy consumption for devices is not increased over legacy LoRaWAN. The per-device regularity of downlink transmissions from the gateway is also not increased in comparison to legacy LoRaWAN, which is a key factor as these transmissions are limited in LoRaWAN networks. The automatic scheduling of timeslots (as opposed to offline or pre-allocation of slots) enables greater flexibility and adaptability for the LoRaWAN network.

As far as we are aware, this periodic traffic model-based system is a novel approach for TDMA over LoRaWAN. Though TDMA-like approaches have been applied to LoRaWAN before, this work is novel because it takes into account some characteristics of mMTC traffic to provide scheduling that does not require extended downlink transmissions from the gateways, and does not require tight time synchronisation between the devices and the LoRaWAN Network Server. LTS can also be easily integrated into the LoRaWAN specification through the addition of just one new set of MAC commands.

This section describes the step-by-step methodology of our proposed approach, LTS. The method consists of five steps:

A. Data Collection

B. Periodicity Detection

C. Collision Prediction

D. Collision Avoidance

E. Decision Propagation

LTS is run on the Network Server of the network, for each individual LoRaWAN gateway. The method is run once every m seconds, which is a configurable parameter. In this work, it is assumed the method is run once every hour, and the focus is on frequently transmitting LoRaWAN devices (i.e. every device transmitting several times per hour); however, the method could equally be applied in a network with seldomly transmitting devices (e.g. once every six hours, or once per day) by instead running the method less frequently and thus over a longer timeframe (e.g. once per day, or once per week). All of the processing for the method takes place on the Network Server, and thus the method is not dependent on the processing abilities of individual LoRaWAN devices or gateways.

5.2.1 Data Collection

The initial step is to record the necessary information required to identify periodicity. In particular, for each device in the network, the Network Server maintains a record of the time received, gateway, data rate, packet length, and channel of each received uplink frame.

5.2.2 Periodicity Detection

The next step is to analyse the recorded data to classify each recorded frame as periodic or event-based. For the time period being analysed, the activity of each LoRaWAN device is represented by a separate discrete bit sequence $S = \{t_0, t_1, ...t_{n-1}\}$, with size n, where the ith bit of the sequence represents the activity of the device in the time between $i * q$ and $(i + 1) * q$, where q is the time taken to transmit a packet of a set length using the current data rate of the device. This timeslot length m for each data rate is a configurable parameter, but should be long enough to prevent the transmission time of a LoRaWAN packet exceeding the length of two timeslots. As previously mentioned the frequency that the method is run is also a configurable parameter; thus the value of n is calculated based on this value m and the timeslot length q:

$$n = m/q \tag{5.1}$$

Then, for each timestamp, $t_i = 1$ if a packet has been received by the Network Server from this device in timeslot i, and otherwise $t_i = 0$.

Code Extract 5.1 shows pseudocode of the data collection and sequence generation mechanism. The retrospective timeslot for an incoming packet can be calculated based on the arrival time, the time since the newest timeslotting iteration began, and the size of the timeslots (line number 24). If the data rate for the device has changed since the last iteration of the full algorithm, then the algorithm is not run for the device this iteration as the generated sequence is likely to be incomplete or unreliable (line number 8).

Code Extract 5.1: LTS Data Collection and Sequence Generation

```
1
2  void LoRaWANNetworkServer::HandleReceive(Packet p)
3  {
4      ... // all of the previous HandleReceive code
5      uint deviceAddr = p.GetAddr();
6
7      auto ed = m_endDevices.find (deviceAddr);
8      if(ed.currentDataRate != p.GetDataRate())
9      {
10         //the data rate has just changed
11         ed.currentDataRate = p.GetDataRate();
12         // don't run the periodicity detection algorithm on this
               node this time, unless this is the first time this device
                has been seen
13         if(ed.uniqueUSPackets != 1) {
14             ed.m_timeslotsDrChanged = true;
15         }
16
17         // and change the size of the vector holding the timeslots,
               and reset the timeslots delay
18         ed.timeslotsRecorder = std::vector<unsigned char>(
               timeSlotsPerDataRate[ed.currentDataRate].m_slots, 0);
19         ed.timeslotDelay = 0;
20     }
21
22     float slot_time = Simulator::Now ().GetSeconds() -
           currentTimePeriodStart.GetSeconds();
23
24     float slot_index = floor(slot_time / LTSperiodInSeconds *
           timeSlotsPerDataRate[ed.currentDataRate].m_slots);
25
26     ed.timeslotsRecorder[int(slot_index)] = 1;
27 }
```

The goal is thus to identify the periodic patterns in the generated sequence, where event-based traffic effectively represent false positive readings, and missed uplink frames (due to collisions or otherwise) represent false negative readings.

Autocorrelation of the timeslot sequence is used to find candidate solutions C across the entire search space. Pseudocode of the autocorrelation procedure is provided in Code Extract 5.2. Autocorrelation of the entire sequence reveals candidate periodicity values (line number 7), and then the correlation of a sequence with that found periodicity (and an offset of 0) with the original sequence reveals candidate offset values (line number 17).

Code Extract 5.2: LTS Data Collection and Sequence Generation

```
1
2  void LightweightTimeslots::FindCandidateSolution(vector<char>&
       sequence)
3  {
4      Correlation(sequence, sequence, correlation_output);
5
6      //choose the max value found
7      int periodicity = GetIndexOfMax(correlation_output);
8
9      // generate a sequence with that periodicity with the offset 0
10     vector<char> new_sequence(sequence.size(), 0);
11     for(uint i=0;i<new_sequence.size();i+=periodicity) {
12         new_sequence[i] = 1;
13     }
14
15     //then get the correlation of that sequence with the original
           sequence, and find the angle between them
16     Correlation(sequence, new_sequence, correlation_output);
17     int offset = GetIndexOfMax(correlation_output);
18
19     return tuple<int, int>(periodicity, offset);
20 }
21
22 void LightweightTimeslots::Correlation(vector<char>& x,  vector<char
       >& y, vector<float>& z)
23 {
24   //enter data into the pre-allocated matrices. Put y in backwards
         as this is correlation, not convolution
25   int ysize = y.size();
26   for(int i=0;i<ysize;i++) {
27     inX[i][0] = x[i];
28     inY[i][0] = y[ysize - 1 - i];
29   }
30
31   fftw_execute(inX, outX, forward);
32   fftw_execute(inY, outY, forward);
33
34   //perform element-wise multiplication of x and y
35   //(x + yi)(u + vi) = (xu - yv) + (xv + yu)i
36   for(uint i=0;i<x.size()*3 - 1;i++) {
37     inZ[i][0] = (outX[i][0] * outY[i][0]) - (outX[i][1] * outY[i
         ][1]);
38     inZ[i][1] = (outX[i][0] * outY[i][1]) + (outX[i][1] * outY[i
         ][0]);
39   }
40
41   //get ifft of result
42   fftw_execute(inZ, outZ, backward);
43
44   //and output the absolute value
45   for(uint i=0;i<z.size();i++) {
46     z[i] = std::abs(complex<double>(outZ[i][0], outZ[i][1])) / (x.
         size()*3 - 1);
47   }
48 }
```

Autocorrelation is a well known method for periodicity detection in binary sequences, but struggles to accurately detect exact period and offset values [155]. Thus, the approach presented by [155] is then used to score the subset of solutions in the vicinity of found candidate solutions. This approach is suitable for sequences with multiple periodic patterns and low sampling rates, and is less sensitive to noise than approaches based on the Fourier transform or autocorrelation. The system is based

on a score function:

$$score = (1 - \alpha)(\frac{|S_T C_T|}{S_T}) - \alpha(\frac{|S_F C_T|}{S_F}) \tag{5.2}$$

where S_T is the number of 1s in S (total number of positives), S_F is the number of 0s in S (total number of negatives), $|S_T C_T|$ is the number of elements of the sequences S and C that are both 1 (total number of true positives), and $|S_F C_T|$ is the number of elements of the sequences S and C where the value in C is 1, but the value in S is 0 (total number of false positives).

This function thus scores potential periodic patterns by their closeness to the dataset in terms of false positives and negatives. Candidate solutions found using the autocorrelation method are used to direct a local search. Neighbouring solutions are scored, and the highest scoring neighbour, if it scores above a threshold α, is filtered out of the next iteration of the search. Each time a solution is filtered out of the sequence, the global search begins again through the initial use of the autocorrelation function. The search continues until the highest scoring candidate solution has a score less than the threshold. With a properly chosen α, periodic patterns with some missing transmissions will still be detected, and event-driven traffic will result in poorly scoring periodic patterns and thus will not be detected as periodic data.

This detection method is computed on the data collected for each device in the network, and provides an output of a set of tuples of candidate solutions in the form (p, o, id), where p is the periodicity, o is the offset of the sequence, and id is a unique identifier for the device.

5.2.3 Collision Prediction

The previous step results in a set of tuples for each device indicating the expected periodic traffic. Each sequence of a device using the same data rate will be the same length. The next step is to identify devices that have periodic transmissions that overlap in time and data rate. i.e. for two devices that overlap in data rate and location, calculate the overlap in their generated sequence. If the overlap exceeds some threshold, then the offset of one of the device's traffic will be modified to prevent later collisions.

Namely, across the periodicity set $\{(p_0, o_0), \ldots (p_N, o_N)\}$, integer solutions are found for multiples of p_i and p_j that are equal to one another when considering the relative offset i.e. when $ap_i + bp_j = o$, where a and b are integers and $o = |o_i - o_j|$. This is the simplest form of the Diophantine equation, and is known to have a solution [156].

Firstly, we define $d = gcd(p_i, p_j)$. If o is a multiple of d, then if the sequences (p_i, o_i) and (p_j, o_j) are generated there will be an eventual shared member of the sets. If o is not a multiple of d i.e. $o\%d \neq 0$, then the two sequences will never overlap.

Example 1: For the tuples (50, 1) and (20, 2), which correspond to the sequences $\langle 50n + 1 | n \in \mathbb{N} \rangle$ and $\langle 20n + 2 | n \in \mathbb{N} \rangle$ respectively:

$$d = gcd(50, 20) = 10$$

$$o = |2 - 1| = 1$$

$$1\%10 \neq 0$$

Therefore o is not a multiple of d. If we generate the sequences:

$$\langle 50n + 1 | n \in \mathbb{N} \rangle = (1, 51, 101, 151, 201, ...)$$

$$\langle 20n + 2 | n \in \mathbb{N} \rangle = (2, 22, 42, 62, 82, ...)$$

It can be seen that indeed the two sequences will never overlap. A graphical representation of the sequences is shown in 5.1.

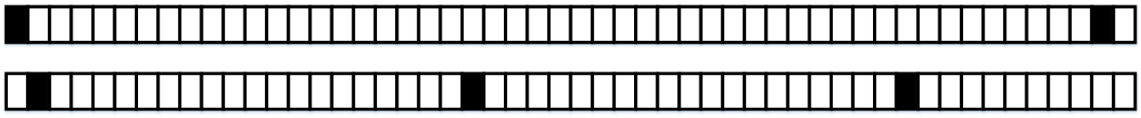


Figure 5.1: Graphical Representation of the Sequences of Example 1

Example 2: For the tuples (50, 1) and (20, 11), which correspond to the sequences $\langle 50n+1 | n \in \mathbb{N} \rangle$ and $\langle 20n + 11 | n \in \mathbb{N} \rangle$ respectively:

$$d = gcd(50, 20) = 10$$

$$o = |1 - 11| = 10$$

$$10\%10 = 0$$

Therefore o is a multiple of d. If we generate the sequences:

$$\langle 50n + 1 | n \in \mathbb{N} \rangle = (1, 51, 101, 151, 201, ...)$$

$$\langle 20n + 11 | n \in \mathbb{N} \rangle = (11, 31, 51, 71, 91, ...)$$

It can be seen that the two sequences indeed do overlap. A graphical representation of the sequences is shown in 5.2.

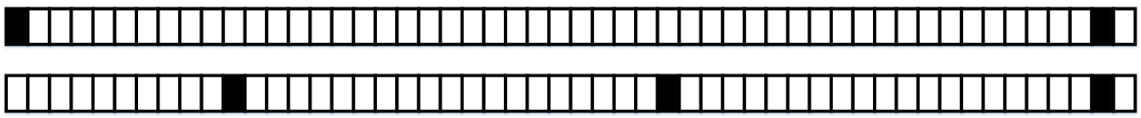


Figure 5.2: Graphical Representation of the Sequences of Example 2

Then, if two overlapping sequences are found, the percentage of each sequence's members that will overlap with the other sequence can be calculated using $\frac{p1}{lcm(p1,p2)} * 100$ and $\frac{p2}{lcm(p1,p2)} * 100$

respectively.

From Example 2 it can be seen that the first overlap of the sequence occurred at 51. Then if $lcm(50, 20) = 100$ is calculated, this shows that the sequence of overlapping members of the sets is given by the equation $51 + 100k$, and that the percentage of members of the $(50, 1)$ sequence that overlap is $\frac{50}{lcm(50,20)} * 100 = 50\%$.

5.2.4 Collision Avoidance

Now, the principles outlined in the previous step are used to change the offset of periodicities to prevent future collisions between devices. Note that in (p_i, o_i) periodicities, the p values cannot be changed. Therefore, the *gcd* and *lcm* values between pairs of sequences also cannot be changed i.e. two sequences either 1) do collide, and collide by a certain percentage of elements, or 2) never collide. Thus the goal is to modify the o values between pairs of sequences to minimise the number of o values that are a multiple of the pair's d value. The collision avoidance algorithm is described in Algorithm 5.1. The heuristic algorithm works by building a vector for each data rate describing the usage of each timeslot, and then modifying o_i values to balance the usage of timeslots.

Algorithm 5.1 Collision Avoidance Algorithm

```
1:  once every period:
2:  for each LoRaWAN data rate do
3:      sort (p,o) tuples by p, then o
4:      create a vector of size timeslots(DR), of all 0s
5:      for each (p,o) do
6:          increment used slots in the vector
7:      for each (p,o) do
8:          count number of overlapping slots with this (p,o)
9:          (not including own transmissions)
10:         divide by number of transmissions
11:     if overlapCount ≥ 1 then
12:         repeat count for new o values of {o ≤ x ≤ o + max_change}
13:         (i.e. the potential slots that could be used)
14:     if lowestOverlap < overlapCount then
15:         modify the o value of this (p,o) to fit new slot
16:         modify the initial vector to reflect this change
```

Such a timeslot scheduling approach introduces delay for all future transmissions from a device. In the general case, an acceptable latency for mMTC is 10 seconds [157], and thus this is maintained as the maximum delay which the algorithm can impose. The *max_change* variable reflects the maximum number of timeslots that a device can be shifted while still maintaining below this level of delay. This variable thus is relative to the size of the timeslot for the data rate; an example of *max_change* values is provided in the next section.

In addition, the constraint is added that a device can only be assigned a timeslot which is after its initial o_i value, with the logic that the reason for a transmission may be time-specific, and thus

this approach enables devices to wake to take a sensor reading, then return to sleep and wake once again to transmit at the assigned time. Note that the changes to the o_i values are maintained across time, and that the *max_change* value is relative to the initial o_i value that the device had at the very first time the algorithm was run for this device. The output of the algorithm described in this section is a change of o_i value, designated c, for a subset of (p_i, o_i) that will reduce future collisions across devices.

In the best case scenario, where all transmissions are periodic and there is no packet loss, future collisions from devices will be reduced. In a system with periodic traffic and significant collisions, not all of the periodicities will be detected successfully. However, a subset of future collisions will be mitigated, enabling those other periodicities to be potentially detected in future runs of the algorithm. In the worst case scenario, where all traffic is aperiodic, it is possible for the system to detect periodicities and potential collisions where there in reality is none. However, in this case the worst end result is a delay to all of the transmissions of a device with purely event-based traffic. Since the traffic is entirely aperiodic, this does not increase or decrease the probability of collisions for future transmissions from the device.

5.2.5 Decision Propagation

Finally, a LoRaWAN MAC command and response are defined which can be piggybacked onto the next downlink frame for a device, which instructs the device to delay all future uplinks by c timeslots, which corresponds to a set amount of time dependent on the LoRaWAN data rate of the device. The original channel selection method of LoRaWAN is maintained i.e. channel selection is performed on a random basis from any channel that is available for immediate transmission without breaking the duty cycle regulations of the EU 868 MHz band.

The format of these new MAC commands are provided in Figures 5.3 and 5.4. These MAC commands can be included in the FrameOptions field of the LoRaWAN Frame Header when required, as seen in Figure B.11 in Section B.2.3. The TimeslotDelayReq MAC command, sent only in downlink frames, contains a single byte which represents the number of timeslots a device is to delay all future transmissions by. A check is performed both on the Network Server and the device to prevent the device from choosing a timeslotDelay value that will result in an introduced delay that is greater than the accepted maximum tolerable delay for mMTC communications of 10s. The TimeslotDelayAns MAC response, sent only in uplink frames, contains a single byte, the low order bit of which indicates whether the device could successfully change the timeslot delay. The remaining seven bits are reserved for future use.

Size (bytes)	1
TimeslotDelayReq Payload	timeslotDelay

Figure 5.3: Format of the TimeslotDelayReq MAC Command

Size (bytes)	1
TimeslotDelayAns Payload	Status

Figure 5.4: Format of the TimeslotDelayAns MAC Command

Note that each new downlink frame generated does put additional strain on the gateway (and can impact the packet delivery rate as current LoRaWAN gateways operate in half-duplex). However, the LoRaWAN ADR scheme for maintaining optimal data rate (use of which is highly recommended by the LoRaWAN specification) does already require the transmission of 1 downlink frame every 32 uplink frames. The generated MAC commands from the described algorithm can be piggybacked on those frames.

Runs of LTS are computed only on data gathered since the last iteration. Clock drift of devices will thus be handled automatically based on the received time of recent frames, with new *timeslotDelay* values allocated as required. Thus, there is not an accumulative effect from clock drift, and time synchronisation of devices is not required.

A toy example showing the collision avoidance of LTS is shown in Figure 5.5. The example network contains three devices which are sending periodic traffic with the same periodicity. The traffic of two of the devices collides continuously in time. However, as the traffic will not continuously collide in frequency, enough of the traffic from both devices can be successfully received in order to accurately determine the periodicities and identify that future collisions will occur. A MAC command is then sent to device C to indicate to the device to delay all transmissions by an amount of time that is equal to the length of one slot. There are then no more collisions in the network.

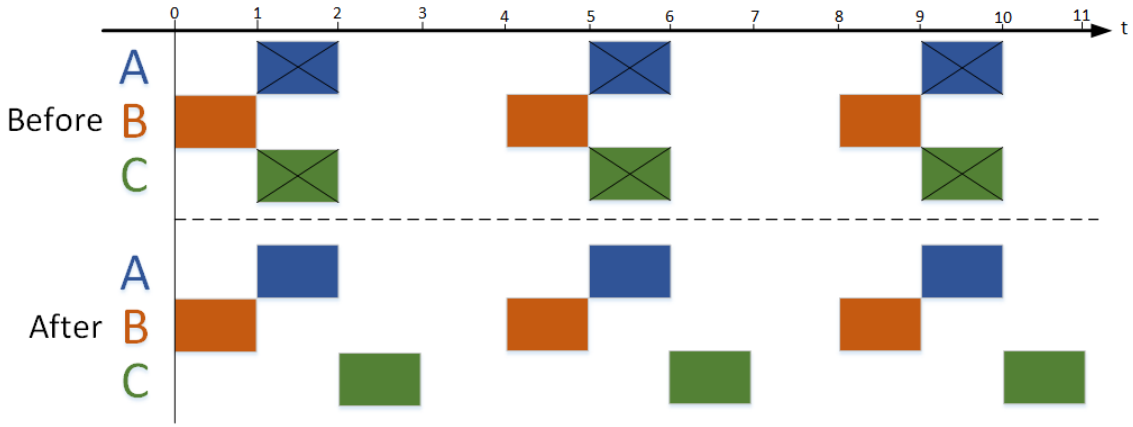


Figure 5.5: Toy Example of the Collision Avoidance of LTS

The structure of LTS is shown in Figures 5.6 (in green) and 5.7. The Network Server is modified to maintain the recording of timeslot sequences, and both application layers are modified to handle the additional MAC command. All other functionality is extracted out to the LTS object, which as a singleton handles all of the functions related to the method and is directly linked to the sole Network Server of the simulation.

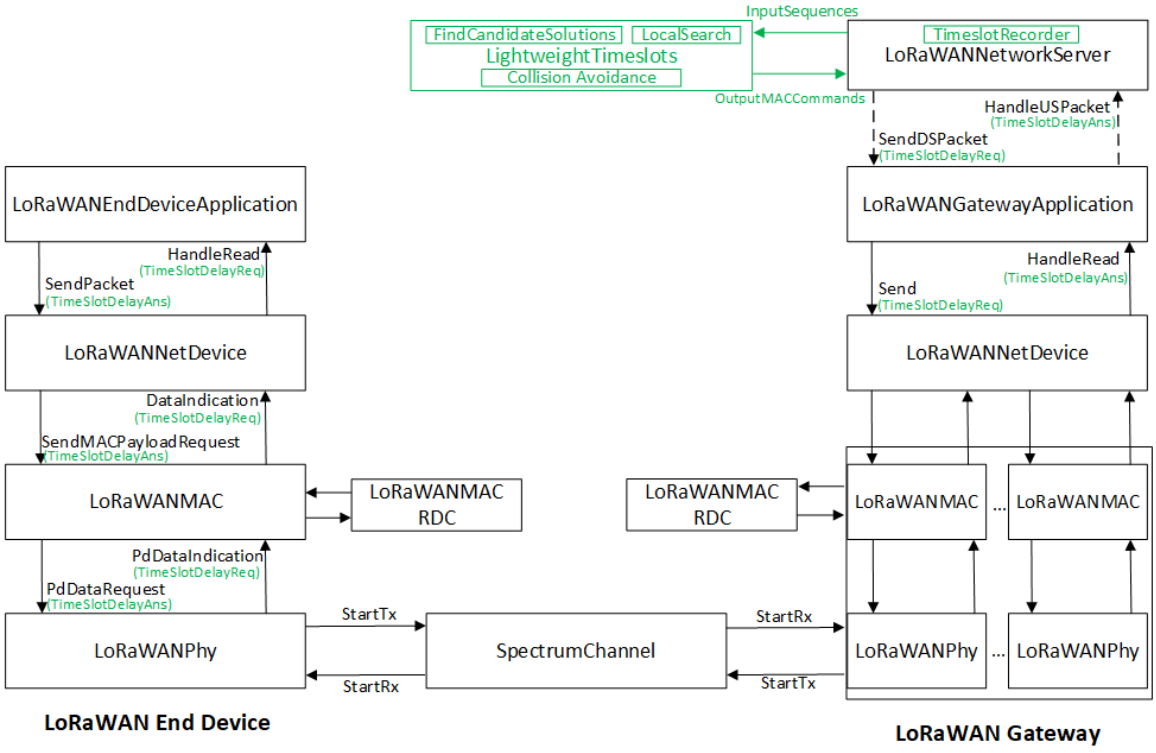


Figure 5.6: Integration of LTS into the LoRaWAN ns-3 Module

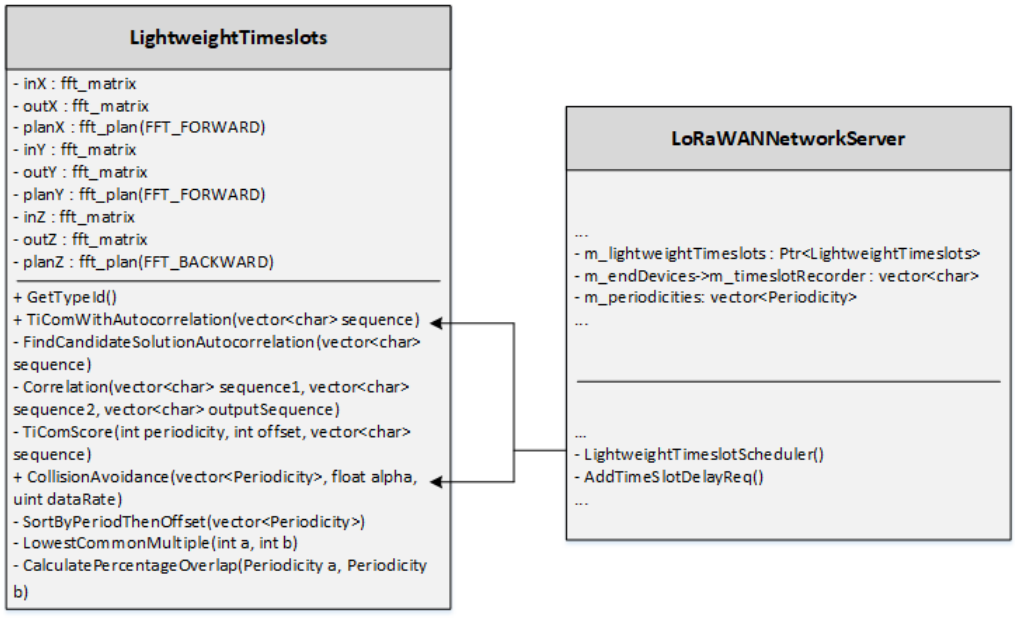


Figure 5.7: LTS Object Structure

5.3 Evaluation

LTS is implemented in ns-3, and compared to a previously released implementation of LoRaWAN [75]. The modified module has been released for public use on GitHub [1]. The simulations consist of a single LoRaWAN network, where N LoRaWAN Class A devices are equally distributed across a disk of radius 4km. Devices are assigned an appropriate LoRaWAN data rate based on the distance from the gateway, representing an overall topology which would be reached as a result of each device using the LoRaWAN ADR. The key parameters in the simulations are outlined in Table 5.1. Each simulation is run five times, using a different random stream. Devices transmit initially at a uniformly random time between 0 and *Uplink_Period*, and then transmit once every *Uplink_Period*. This is equivalent to the periodic, deterministic traffic modelled in [158]. Note that the periodicity of a transmission every 600s is actually very frequent for mMTC use cases [157], and thus in terms of number of devices supported the results shown can be considered a difficult case scenario. The standard error for the listed metrics never exceeds 0.4 for any of the results provided.

Table 5.1: Simulation Parameters

Gateways	1
Devices	100, 200, 400, ... 5600
Disk Radius	4000m
Uplink Period	600s
Downlink Period	No data plane DL
Algorithm Frequency	once per hour
Packet Size	20 bytes (excluding header)
Random Streams	5
Simulation Time	150 * Uplink Period
α	0.8

As Equation 5.2 shows, α represents the sensitivity of the periodicity detection approach to false positives and false negatives. Offline tests were conducted to find a generally suitable α value of 0.8, which can account for collisions (false negatives) and event-based traffic (false positives).

For simplicity, it is assumed that all devices are sending frames of the same length, and that the network features only one gateway node. Every uplink packet sent is 33 bytes long, corresponding to 13 bytes for the LoRaWAN header and 20 bytes of application-layer payload. The length of a timeslot for DR0 is calculated to be the length of time to send a frame of this length. The length of a timeslot for each other data rate is calculated to be half the length of the transmission time of the next slowest data rate, which is an approximation of the difference in time taken to transmit two equivalent LoRaWAN packets using two different data rates. As previously mentioned, we assume the algorithm runs once per hour. Thus, the length of a timeslot, and thus the number of timeslots and *max_change* value for each data rate is provided in Table 5.2.

[1] https://github.com/ConstantJoe/ns3-lorawan-lts

Table 5.2: LoRaWAN Timeslots (33 Byte Transmission)

Data Rate	**Timeslot Length**	**Timeslots In An Hour**	**Max Slot Change**
DR0	1.812s	1986	5
DR1	0.906s	3972	11
DR2	0.453s	7944	22
DR3	0.227s	15888	44
DR4	0.113s	31776	88
DR5	0.057s	63552	176

mMTC devices only infrequently report data and as such metrics such as total network data throughput are not relevant, as there is a maximum amount of required transmitted data. In addition, as frame confirmation is not a scalable approach for all frames [159], and blind retransmissions are not scalable [159], the key metric in performance evaluation in this case is PDR. Delay is also measured to ensure the introduced delay by the system never exceeds the mMTC tolerable delay of 10s. In these simulations, the PDR during the initial two runs of the algorithm (first two hours of simulation) are not factored in, as a regular LoRaWAN network would not have so many devices initially connecting to the network at the same time.

5.3.1 Results

Figure 5.8 shows the PDR of LoRaWAN with LTS implemented compared to a regular implementation of LoRaWAN, for an increasing number of devices. The figure shows that the proposed algorithm enables a single LoRaWAN gateway to handle the traffic of approximately twice as many devices, while providing the same level of PDR and maintaining the restriction on delay. Note that the proposed system does not achieve the increase in scalability provided by classic TDMA in comparison to ALOHA, However, the approach is dynamic and does not pre-assign slots. Additionally, it responds to the data patterns of the available devices, instead of preemptively defining the full available schedule of slots for devices. Finally, it requires a very small amount of control plane traffic to implement (just the addition of one MAC command to a downlink frame that would still be otherwise transmitted to each device). A classic implementation of TDMA cannot be reliably implemented over LoRaWAN because of the duty cycle limits on the 868 MHz band, but our proposed algorithm achieves a form of TDMA that can greatly improve network performance while handling these limitations.

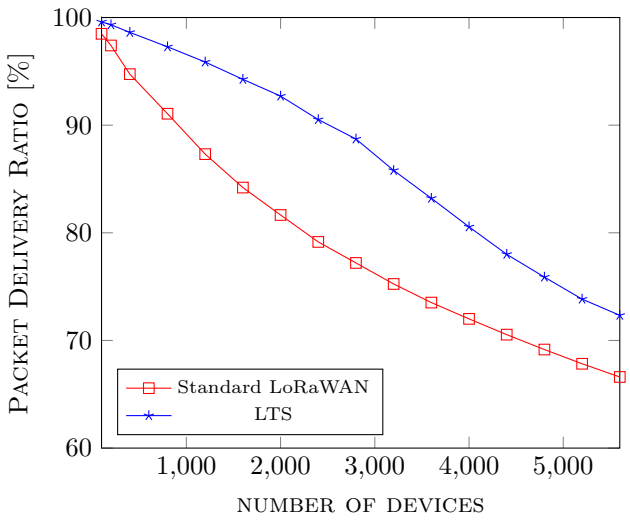


Figure 5.8: PDR of a LoRaWAN Network with LTS Implemented vs Standard LoRaWAN, for an Increasing Number of Devices.

A note about the potential scalability of the system: in these simulations devices are equally dispersed around the gateway in a disk of radius 4 km. However, as well as being more energy-efficient, the faster LoRaWAN data rates also enable the splitting of time into smaller timeslots, allowing the handling of approximately twice the number of devices for each faster data rate. In addition, a faster data rate enables a greater *timeslotDelay* change to be allocated by the Network Server while still maintaining a tolerable delay, providing more flexibility and greater potential efficiency of the system. The number of devices allocated to each data rate in a sample simulation of 5600 devices (equally dispersed) and the resulting average PDR for devices using each data rate, versus the proportion of timeslots available for each data rate is provided in Table 5.3. If devices were dispersed in such a way to enable devices to use a faster data rate on average, there would be increasing gains using our scheme. This could be achieved by deploying a greater density of gateways in the network, while the LoRaWAN ADR algorithm is enabled on all devices.

Table 5.3: LoRaWAN Devices Allocated per Data Rate vs. Proportion of Timeslots (in a Simulation of 5600 Devices)

Data Rate	**Devices Allocated**	**PDR**	**Timeslots in an hour**	**% of available timeslots**
DR0	23.14%	48.34%	1986	1.58%
DR1	21.05%	66.63%	3972	3.17%
DR2	16.32%	79.11%	7944	6.35%
DR3	13.13%	84.53%	15888	12.70%
DR4	11.39%	87.46%	31776	25.40%
DR5	14.96%	88.56%	63552	50.79%

Figures 5.9 and 5.10 show the max and mean of the delay of transmissions for each data rate, respectively, for an increasing number of devices.

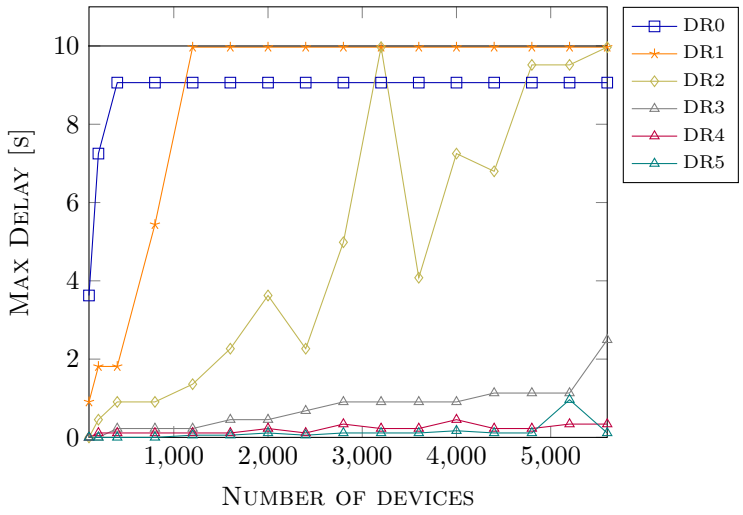


Figure 5.9: Max Delay of an Uplink Frame of each Data Rate, for an Increasing Number of Devices

Note that the mean delay never exceeds 0.06s for DR4, DR5, and DR6. As regular LoRaWAN uses ALOHA without any channel sensing, the delay for regular LoRaWAN is always $0s$. In LTS, the mean delay increases with the number of devices, as a higher proportion of filled timeslots results in the algorithm having to allocate timeslots to devices farther away from the initial projected transmission time. Note that the max delay represents the maximum delay introduced to any device of the particular data rate (and number of devices in the network), across any simulation run, and so this metric is prone to influence from outliers. However, the delayed transmission time for any device never exceeds $10s$, and so the restriction on the acceptable delay for mMTC communications is maintained.

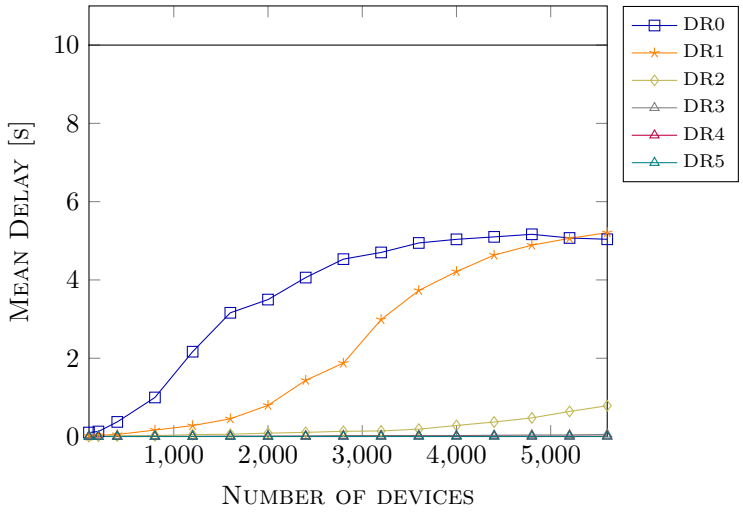


Figure 5.10: Mean Delay of an Uplink Frame of each Data Rate, for an Increasing Number of Devices

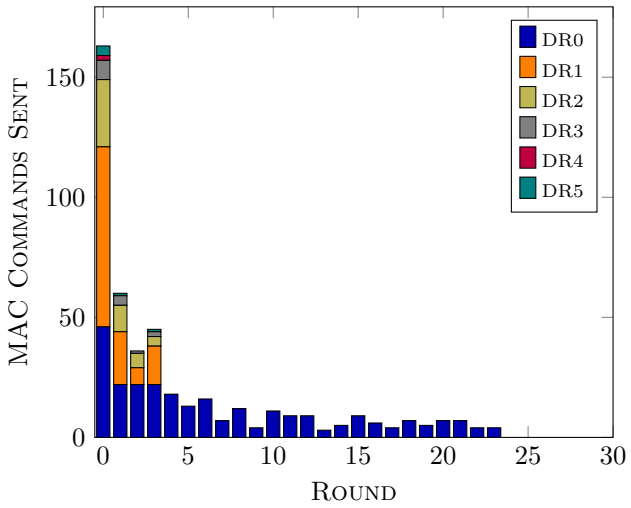


Figure 5.11: Number of MAC Commands Sent After Each Run of the Algorithm, for each Data Rate (1600 Devices)

Figure 5.11 shows the number of MAC commands that are sent after each run of the algorithm, using each data rate, in an example simulation consisting of 1600 devices. Unsurprisingly, with a roughly equal number of devices allocated each data rate and a transmission using one data rate taking approximately twice the amount of time of the next fastest data rate, the number of collisions increases roughly by two for each slower data rate. The initial run of the algorithm detects and handles the majority of colliding periodicities; the next few runs detect colliding periodicities that were not detected initially because of heavy packet loss in early rounds. The next few rounds also propagate MAC commands that were not able to be sent by the gateways in the initial run because of the duty cycle restrictions of the EU 868 MHz band. Eventually for each data rate collisions are avoided and later MAC commands are not required. This simulation was run for the equivalent of 100 rounds of the LTS algorithm. While DR0 in this simulation is slow to converge to a situation where no MAC commands are needed per round because the number of devices using DR0 is high enough that nearly all of the timeslots are used, it does eventually converge; after 23 rounds, no more MAC commands are sent.

5.4 Conclusions

In this chapter, LTS, a novel approach to TDMA over LPWAN has been presented, which is particularly suited to the characteristics of devices transmitting periodic data over LoRaWAN. The scheme does not require extended downlink transmissions from the gateways, and does not require tight time synchronisation between the devices and the LoRaWAN Network Server. Results from simulations show that the scheme doubles the number of frequently transmitting devices (once every 10 minutes) that can be handled by a single LoRaWAN gateway while providing the same level of PDR and maintaining a tolerable delay for mMTC use cases, without impacting the ability of the network to send downlink frames or acknowledge high priority packets. Although the energy usage

has not been simulated in detail, the energy usage for devices will not change significantly as, in the default LoRaWAN Class A unconfirmed mode, frames are not retransmitted whether they are received successfully or not.

In comparison to previous work (described in Section 5.1), the proposed scheme is lightweight, fully automatic, and easily integrated in the existing LoRaWAN protocol. As far as we are aware, this is the only proposed scheme that factors in traffic characteristics of mMTC in the TDMA design. The system is particularly suited to LoRaWAN as the channel access and data rate allocation of LoRaWAN enables the detection of future colliding frames. Additionally, the system is especially appropriate for transmission over the sub-GHz ISM bands as the introduced overhead is minimal and the transmission of additional control plane frames is not required. Overall, the system enables greater network performance for large scale networks of devices transmitting periodic traffic.

CHAPTER 6

Conclusions and Future Work

The aim of this research and the objective of this thesis was to improve the performance and scalability of LPWAN networks. This was achieved through analysis of the behaviour of realistic LPWAN devices in terms of energy efficiency and reliability, as the network scales to a density to be expected of an urban environment. With reference to the research objectives presented in Chapter 1, the main contributions and conclusions from this thesis are:

Permitting the prediction of the battery lifetime of LoRaWAN devices: A analytical model of the energy consumption of LoRaWAN was developed, based on the PHY and MAC layer states of the protocol and the energy profile of real LoRaWAN modules and transceivers. This model was used in a study of the comparative energy consumption rate of different LPWAN technologies. The analytical model was also used in the verification of the integration of a LoRaWAN energy module for ns-3. Finally, the model enables the analysis of the feasibility of the powering of LoRaWAN applications fully through the harvesting of ambient RF energy.

Enabling analysis of realistic large scale LoRaWAN networks in ns-3: Through the work presented in this thesis, the assessment of more realistic LoRaWAN networks in simulation has been enabled, through the development of novel extensions to existing work on simulation of LoRaWAN in ns-3. Firstly, a LoRaWAN energy model has been developed to enable the estimation of device lifetimes in simulation. Additionally, Class B of LoRaWAN has been implemented in ns-3 to allow the analysis of reactive and evolving LoRaWAN applications with stricter requirements on downlink traffic. Finally, the ADR of LoRaWAN has been implemented, enabling the evaluation of adaptive and realistic LoRaWAN networks. Each of these extensions have been released open-source, enabling their use by other researchers in the future. Additionally, each of these extensions are the first time that the features have been implemented in ns-3.

Evaluating large-scale LoRaWAN systems through simulation: Each of the developed extensions to LoRaWAN simulation in ns-3 have been used to evaluate the performance of LoRaWAN in large scale systems, primarily in the case of a single gateway handling a large amount of devices. Simulations of LoRaWAN networks under a variety of conditions have been performed, with devices running realistic applications with mMTC traffic and with networks consisting of the number of devices per gateway which is the target of Semtech. This analysis can be considered state-of-the-art as it constitutes the first time the analysed LoRaWAN features have been implemented in ns-3. Novel findings related to LoRaWAN Class B and ADR have been presented, including edge cases which limit the adaptability and functionality of each feature, and the behaviour of each feature as the network scales. Note that ns-3 is just a singular network simulator. The support of simulation of LoRaWAN in ns-3 and other simulators is discussed in Appendix B. However, it should be noted that, as a result of this research, ns-3 is now the only simulator with an implementation of both Class B and ADR.

Providing increased adaptability and reliability in LoRaWAN networks: Through the previous analysis, bottlenecks to performance for LoRaWAN networks were identified. Proposed changes to ADR have been developed which increase the scalability of the network and fix issues which can currently limit the network adaptability in some cases. A typical easy solution to overloaded LoRaWAN networks is to increase the density of gateways; our proposed enhancements show an increased effect from doing this as the increase in performance is not purely from reducing the number of devices per gateway, but also from enabling a greater proportion of devices to use faster data rates (which also allows more downlink frames to be sent per gateway).

Increasing the scalability of LPWAN MAC through lightweight and automatic TDMA: Using the properties of LoRaWAN transmission and mMTC traffic, a lightweight collision prediction algorithm has been developed which mitigates the effects of collisions from periodic traffic. The results demonstrate that the scheme doubles the number of frequently transmitting mMTC devices that can be handled by a single LoRaWAN gateway while providing the same level of PDR and maintaining a tolerable delay for mMTC use cases, without impacting the ability of the network to send downlink frames or acknowledge high priority packets.

The work presented in this thesis has been disseminated through presentations at a PhD doctoral conference and four IEEE international conferences, and through two IEEE journal publications. All developed code has been released open-source on GitHub.

6.1 Threats to Validity

The threats to external validity are the differences between the simulated networks and a real world environment. A simulation relies on a developed model of the real world, and not the real world itself, and as such the reliability of the results are dependent on the accuracy of the model to the real world for the phenomena being analysed [160]. At a certain point in the network simulator ns-3, the PHY layer performance becomes an abstraction of the real world. How and when this becomes abstracted is dependent on the simulator module implementation; a discuss of the different

approaches to the modelling of the PHY layer performance in different LoRaWAN simulators is provided in Appendix B.

Threats to internal validity have been minimised by the careful design of simulations, where minimal factors are changed between each simulation set. This enables the identification of the real effect of individual parameters on the network performance. To maintain construct validity, standard well-defined metrics are used in the simulations. Finally, to ensure conclusion validity, for each simulation a set of reproducible random streams are used, as is standard in ns-3 simulations. The design and validity of our results have been legitimised through the acceptance of publication in international IEEE conferences and journals.

6.2 Broader Applicability of the Contributions

Though the exactly scalability and reliability is dependent on the PHY layer performance of individual links, the key performance metrics will remain broadly similar across the majority of unlicensed LPWAN technologies. This is because of the many common factors of these technologies, as described in Appendix A. The introduced approach to TDMA is broadly applicable to any technology under which mMTC traffic is transmitted, though it will be most useful in technologies with similar channel access mechanisms to LoRaWAN.

6.3 Future Work and Research Directions

Based on the work presented in this thesis, there are a number of identifiable areas of future research:

Hardware prototypes of ultra-low power LoRaWAN devices with RF EH capability: The potential for LoRaWAN devices to be powered solely by RF energy harvesting has been demonstrated in this thesis. This work could be further strengthened through the development of a hardware prototype. Such a device should be developed to enable ultra-low power operation by the use of an external clock to allow an extremely low sleep current, and by the transition to low power or even sleep mode in the waiting states of a LoRaWAN transmission.

EH-aware LPWAN MAC: EH-aware MAC is a currently active area of research, primarily for IEEE 802.11 and IEEE 802.15.4 wireless sensor networks. The low power operation requirements of LPWANs mean that these protocols could also potentially be adapted to factor in EH-aware MAC functionality. Such approaches could include probabilistic polling for Class B-based LoRaWAN networks, and harvesting-aware duty cycle adaption for Class A-based networks.

Evaluation of LoRaWAN Class B over 2.4 GHz: As has been shown in the analysis presented in this thesis, the primary bottleneck on the performance of LoRaWAN networks of primarily Class B devices is in the duty cycle regulations applied to the gateway. The new class of SX128* LoRa transceivers operate over the 2.4 GHz band, and as such do not have any restriction for time-on-band. While this bottleneck is removed in the 2.4 GHz band, transmission at a higher frequency

and in a busier band would have other implications for the potential performance of such a network. The overall performance of such a network would be worth exploring. The challenges of operation in this band would be making efficient usage of the increased bandwidth while operating in a more congested part of the radio spectrum and with reduced range.

Assessment of intra-LPWAN interference at 868 MHz in simulation: The Spectrum module of ns-3, which is now integrated into the Wi-Fi, IEEE 802.15.4, and LoRaWAN implementations in the simulator, enables potential analysis of intra-network interference. The development of suitable error models for LoRaWAN with other co-located networks (e.g. IEEE 802.15.4, Sigfox, IEEE 802.11ah, etc.) would allow the analysis of the impact of intra-LPWAN interference in high density urban networks.

Adaptive Data Rate with mobility support using accurate localisation data: As has been demonstrated in this thesis, optimum performance of the ADR of LoRaWAN is essential in ensuring the scalability of the network as a whole. The current LoRaWAN ADR does not support mobile LoRaWAN devices. An extension or adaption of the ADR for mobile devices would enable energy-efficient communications for a much broader range of LoRaWAN applications. This could be achieved using the newly released LoRa Edge LR1110, which combines passive GNSS and Wi-Fi scanning with LoRa Time-Difference-of-Arrival to enable ultra low power localisation.

Hardware prototype of LTS: LTS has been introduced in this thesis as a mechanism to mitigate predictable collisions in LoRaWAN networks. The performance benefit of the algorithm has been demonstrated in simulation using ns-3. These results would be strengthened through the addition of results generated through real hardware experiments.

Generalisation and further analysis of LTS: The value of simulated results of LTS would be further improved through the integration of additional network features. Further simulation featuring event-based traffic (both independent and geographically correlated), simulated clock drift, and the existence of multiple gateways would strengthen the presented results. Additionally, a thorough exploration of α could be performed to enable an adaptive approach to choice of this parameter based on expected network traffic flows. Finally, the effect of variable radio conditions and a variable amount of active devices should be explored, along with a full analysis of the parameters which impact the convergence of the algorithm.

Diversity Combining for LPWAN: The extended simulation of large scale LoRaWAN networks presented in this thesis has revealed a number of potential opportunities for increasing the network performance in dense environments. One such approach that is particularly suitable for LPWAN networks is diversity combining, where multiple copies of a single failed frame can be recombined to retrieve the lost message. In LPWAN networks, this can be achieved by taking advantage of the existing network architecture. For LoRaWAN, multiple copies of the same frame can be received by different gateways, and redundancies are filtered out by the Network Server. In LoRaWAN diversity combining, the Network Server can take the responsibility of recombining the received copies. In Sigfox, each uplink frame is transmitted three times by a device. Diversity combining can be used in Sigfox to recombine these three different frames. In both cases, reliability in the LPWAN network can be increased without extra transmissions or control plane traffic.

APPENDIX A

Background - The Emergence of the LPWAN Paradigm

The key characteristics of LPWAN technologies are energy efficiency, scalability, and long range [161]. LPWAN devices typically need to operate autonomously for years using only affordable power sources, motivating ultra-low power operation and the use of energy harvesting. To reduce infrastructure costs, individual gateways are required to handle the traffic of thousands of devices. LPWAN technologies enable a transmission range in the kilometres, allowing wide areas to be covered with reduced infrastructure.

In this appendix, there is a detailed description and comparison of the more open LPWAN standards: LoRaWAN, Sigfox, NB-IoT, LTE-M and EC-GSM-IoT. Other competing standards are also discussed, namely Nwave, Telensa, Weightless-P, Ingenu, the Dash 7 Alliance Protocol, and WAVIoT. However, a question still remains: why did so many different independent protocols begin to develop in parallel, instead of the adoption of existing standards? To answer this, firstly there will be a discussion of the existing wireless landscape that eventually led to these developments.

A.1 The Motivation for New Standards

Wireless standards can generally be categorised based on the development group: IEEE, 3rd Generation Partnership Project (3GPP), and private companies developing fully proprietary standards. In addition, active research in academia provides alternative solutions and proposals. A number of protocols emerged in (or around) 2015, to target the mMTC component of 5G. For example, the LoRa Alliance was founded in 2015 [162]. Sigfox was founded in 2009 but raised $115m in Series D funding in early 2015 [163]. NB-IoT, LTE-M and EC-GSM-IoT were first defined in 3GPP Release

13 in 2016 [164]. To explore the questions of why so many standards started to develop at this same time, leading to the LPWAN paradigm, the development of the 3GPP, IEEE 802.11 and IEEE 802.15.4 standards up to this time is now addressed.

A.1.1 IEEE 802.11 (Wi-Fi)

The Wi-Fi standards form the core work of the IEEE 802.11 working group, and from the beginning targeted the general use case of a Wireless Local Area Network (WLAN), requiring a range of up to 50 m and acting as a general wireless alternative to Ethernet [165]. From the main precursor of Wi-Fi, WaveLAN, up to the latest standards, the majority of proposed amendments of the standard have focused on this remit, with some notable exceptions that will be discussed. Figure A.1 shows the development history of Wi-Fi and the major amendments of the standard, starting from the precursor **WaveLAN** in 1991 and continuing to **IEEE 802.11ba**, which is still in development. Amendments are grouped together into the eventual version of the standard each was first officially amalgamated into. Note that standards generally follow each other directly, with individual chipsets typically able to transmit several different standards of the protocol (for example, a mobile phone with an IEEE 802.11a/b/g chipset) to enable the graceful obsolescence of devices using older standards.

The amendments that can be considered to focus on wireless connectivity for IoT applications are highlighted in Figure A.1 in red. IEEE 802.11p focuses on connectivity for vehicular networks, which as an IoT application falls into the category of uRLLC, not mMTC. The **IEEE 802.11ah** amendment (i.e. HaLow), approved in 2016, defines modifications to both the PHY and MAC to enable transmission in the sub-GHz ISM bands, providing a minimum data rate of 100 Kb/s and a range of up to 1 km. This amendment is the most similar to the paradigm of LPWAN, and developed at the same time as the major LPWAN protocols. Finally, the currently in-development **IEEE 802.11ba** extends new features from IEEE 802.11ah and introduces an auxiliary Wake-Up Radio, enabling the use of Wi-Fi in low power IoT devices. However, the development of this amendment only began after the proliferation of LPWAN protocols. Overall, though the diversity of IEEE 802.11 PHY and MAC options has increased over time, before the start of the development of IEEE 802.11ah (and now IEEE 802.11ba) there was little focus on providing a suitable standard for energy-constrained devices. This, combined with developments from other wireless standards, led to the development of competing non-IEEE LPWAN standards.

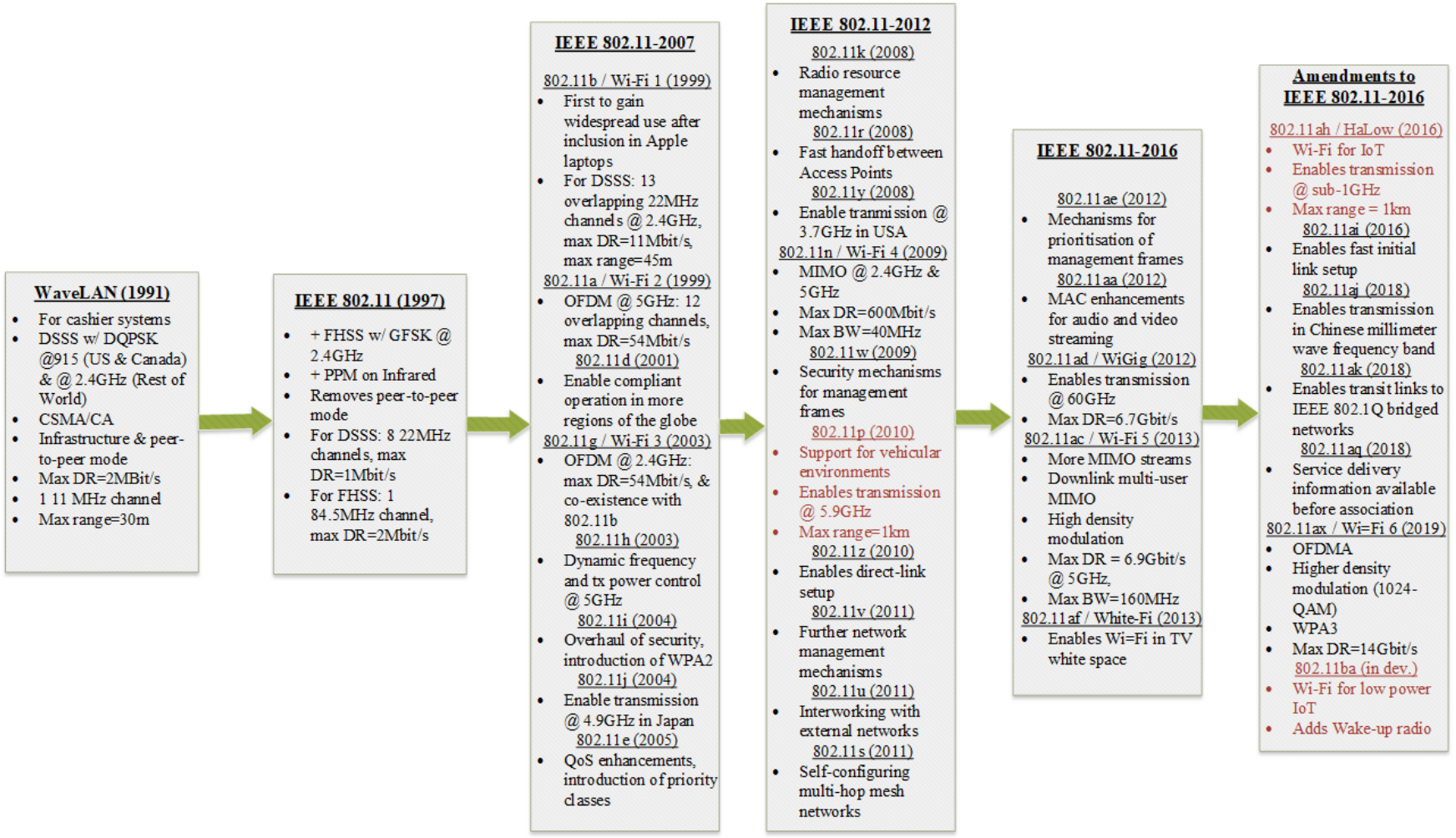


Figure A.1: The Development of Wi-Fi

A.1.2 IEEE 802.15.4

The IEEE 802.15.4 working group primarily develops options for low complexity, low data rate wireless connectivity for energy-constrained devices, both stationary and mobile [166]. Similarly to IEEE 802.11, IEEE 802.15.4 defines one official standard which consists of different PHY and MAC layer options, all of which are designed for low rate, low power wireless networking. IEEE 802.15.4 only defines the PHY and MAC layer of the wireless stack. Upper layers are defined by separate bodies, and these upper layers do not have to support all PHY and MAC features of the IEEE 802.15.4 standard. The most well known upper layer standard for IEEE 802.15.4-based networks is Zigbee. Figure A.2 shows the development history of IEEE 802.15.4, with amendments which are particularly relevant to LPWAN highlighted in red.

In 2012, the **IEEE 802.15.4e** amendment introduced five new behaviour modes in order to overhaul the MAC layer, and define a general protocol with specific modes more suitable for particular industrial applications, beginning an approach to amendments which provide particular PHY and MAC layer options targeting specific applications.

The **IEEE 802.15.4g** amendment added support for smart utility networks (SUN) in 2012. Metering devices in a SUN would typically use one of the SUN PHY layers with some mesh-based approach in order to reach the nearest access point. The SUN PHY layers were later extended in amendment **IEEE 802.15.4x** in 2019 to enable use of additional sub-GHz frequency bands and increase the maximum data rate to 2.4 Mb/s, and to introduce new MAC-related functions for spectrum resource management and spectrum resource usage information exchange, enabling coordination of co-located devices to mitigate interference.

The **IEEE 802.15.4k** amendment introduced low energy critical infrastructure monitoring (LECIM) networks. LECIM networks operate in a star topology, with a mains powered central coordinator and energy-constrained leaf nodes. Uses of LECIM networks include oil and gas pipeline monitoring, water leak detection, soil monitoring, and building monitoring. Finally, the **IEEE 802.15.4w** Task Group is currently developing an extension of the LECIM PHY layer that is suitable for LPWAN applications, enabling a coverage cell radius of 15 km in rural areas.

As can be seen, there are a number of IEEE 802.15.4 standards targeting applications that could be considered LPWAN use cases. However, these standards target specific use cases, and had not reached widespread adoption by 2015, leading to the parallel development of different LPWAN standards. The IEEE 802.15.4 standard which is most similar to LPWAN, IEEE 802.15.4w, is only currently in development. In addition, LPWAN technologies ease the difficulty in deployment through the use of network operators, which enable developers to access a widely available and robust network instead of developing and maintaining their own infrastructure, both in terms of hardware and software. This is of particular importance in applications with mobility requirements.

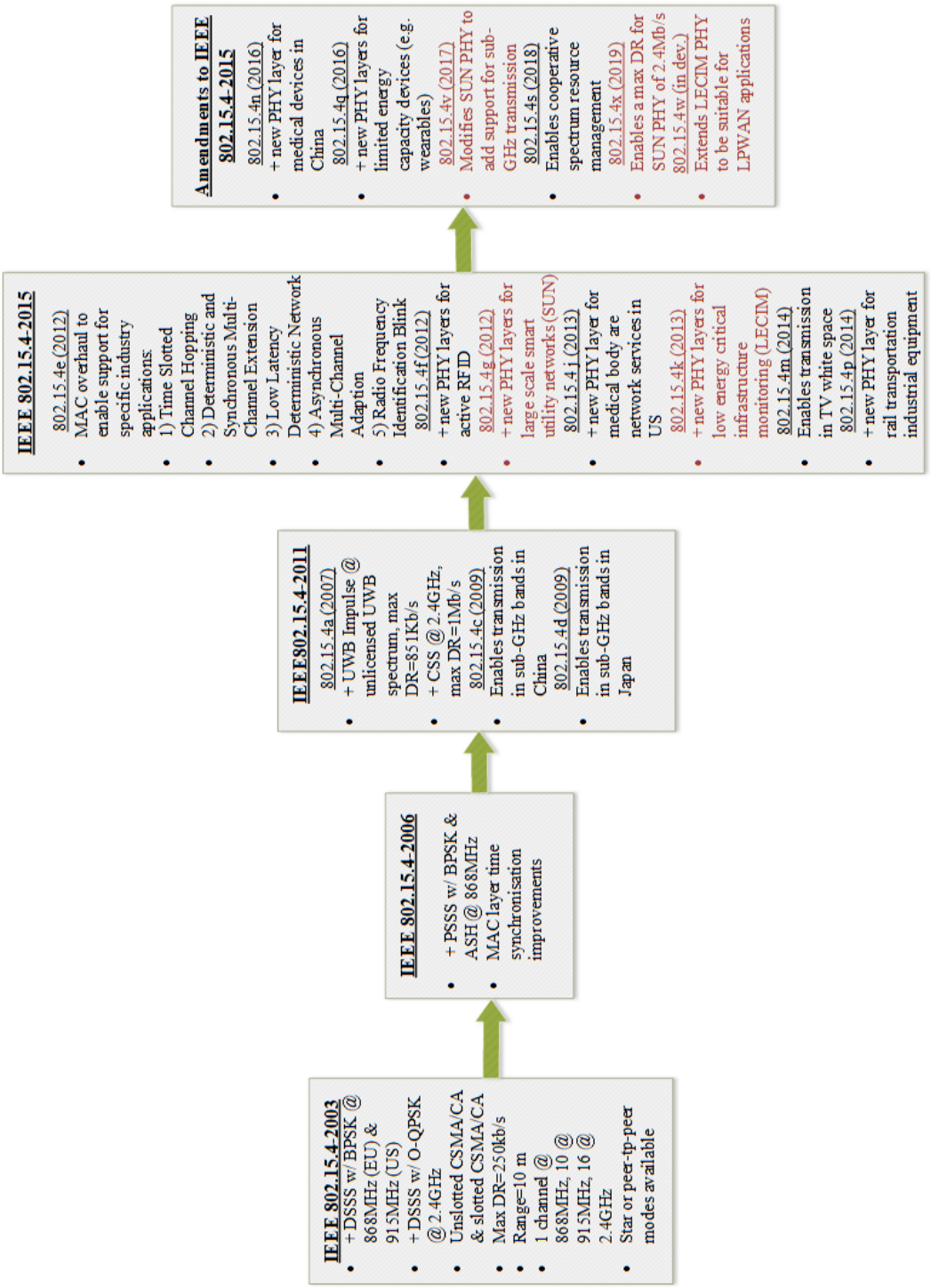


Figure A.2: The Development of IEEE 802.15.4

A.1.3 3GPP

Cellular technologies provided the range required for LPWAN applications but not the energy efficiency requirements. Global System for Mobile Communications (GSM) and the improvements General Packet Radio Service (GPRS) and Enhanced General Packet Radio Service (eGPRS) are the source of connectivity for IoT applications with coverage requirements, and have been used for such applications in the past, such as point-of-sale terminals. However, GSM, which was originally standardised in the early 1990s, was not designed for devices that have such strict requirements on energy consumption as LPWAN-suited applications, and for networks with so many connected (but infrequently transmitting) users. Thus, 3GPP have in development three Cellular-IoT (C-IoT) technologies designed to suit mMTC applications. One of these standards, EC-GSM-IoT, is a direct update for GSM, designed to improve performance for LPWAN-suited applications. GSM networks are slowly now being phased out throughout the world, having already ceased to operate in the United States, Japan, South Korea, and Australia. 2G spectrum will be reused in Europe for 5G IoT services.

Overall it can be seen that, while LPWAN-suitable applications could theoretically have been deployed using IEEE or 3GPP standards (and in fact were, primarily using GSM or IEEE 802.15.4, depending on range and mobility requirements), the lack of widespread adoption of later IEEE 802.15.4 amendments, coupled with the phase-out of GSM and the general reduction in the cost of components required for LPWAN-suitable applications, led to the development in parallel of numerous LPWAN protocols. Newer amendments and protocols from IEEE working groups and 3GPP, such as IEEE 802.11ba, IEEE 802.15.4w, and NB-IoT also target LPWAN-suitable applications, but these began development either alongside or after the initial development of independent LPWAN protocols.

A.2 Defining LPWAN

LPWAN technologies are characterised by a particular focus on energy efficiency, scalability, and long range [161]. LPWAN devices are typically required to be able to function autonomously for extended periods of time (typically a target of 10 years) using only a cheap power source, such as a pair of AA batteries or a button cell. In addition, battery replacement is often unpractical because of the scale of applications in terms of number of devices, and the potential for devices to be deployed in hard to reach areas. This motivates the attainment of ultra-low power operation, and the adoption of energy harvesting approaches where applicable. Typical range requirements are in the kilometres, enabling the coverage of a city or rural area without an unreasonably expensive infrastructure of backhaul nodes. Long range is achieved by substantially reducing the bit rate of transmission, enabling the receiver to receive frames at a very low sensitivity. The receiver sensitivity in LPWAN technologies is typically lower than -128 dBm, compared to -90 to -110 dBm in Wi-Fi [167]. Sub-GHz bands are also used in many LPWAN technologies, to enable transmission with less attenuation and multipath fading, as well as transmission in a less congested band [168].

Low power operation is achieved through the design of a lightweight MAC layer, which typically minimises control plane traffic, and simplifies the network topology to a single-hop star. ALOHA-based MAC enables a connection without frequent synchronisation with neighbouring devices, or the reception of beacon frames from gateway nodes. When no data is being sent or received (which for these applications is over 99% percent of the time), the device reverts to a very low power deep-sleep mode. Additionally, LPWAN devices typically operate on a sense-and-transmit cycle and offload complex operations to a non-local application server, reducing the complexity, cost, and energy consumption of devices in the field [168].

LPWAN technologies can be categorised into unlicensed and licensed protocols. Licensed protocols, such as those in development by the 3GPP, transmit in licensed spectrum and thus do not have to follow stringent regulations on transmissions. Unlicensed LPWAN protocols such as LoRaWAN and Sigfox typically operate in the sub-GHz ISM bands, which are shared by all "Short Range" devices[1]. Use of ISM bands typically requires devices to adhere to regional regulations. In the EU 868 MHz ISM band, the regulations limit devices from communicating more than a particular percentage of the time during a day, fundamentally limiting the potential throughput of the device [12]. As an example, Table A.1 outlines the spectrum access available in the EU 868 MHz ISM band for subbands that support wideband modulation. Devices in these subbands may alternatively adhere to the regulations through the following of a "polite spectrum access" policy[2], whereby the device uses a Clear Channel Assessment (CCA) and random backoff approach to spectrum access.

Table A.1: ETSI Spectrum Access per Subband

Band Number	**Spectrum Access**	**Edge Frequencies**	**Max ERP**
46a	0.1 %	863-865 MHz	14 dBm
47	1 %	865-868 MHz	14 dBm
48	1 %	868-868.6 MHz	14 dBm
50	0.1%	868.7-869.2 MHz	14 dBm
54	10 %	869.4-869.65 MHz	27 dBm
56a	No Requirement	869.7-870 MHz	7 dBm
56b	1 %	869.7-870 MHz	14 dBm

Assessment of the overall duty cycle per subband for a device is made for a representative 1-hour long period, representing the most active period in the normal usage of the device, where normal usage is considered to be the behaviour of the device during transmission of 99% of frames generated during the operational lifetime [169]. Additional restrictions on the band may also be mandated by the national telecommunications regulator. The EU 868 MHz ISM band are shared by all general purpose devices, and so there is potential interference not just from other LPWAN transmissions, but also transmissions over IEEE 802.11ah, the IEEE 802.15.4-based Z-Wave and Zigbee, IO Home Control, amongst others, which can impact LPWAN performance [170]. A graphical representation of the EU 868 MHz regulations is shown in Figure A.3.

[1]Short Range devices are defined by ETSI [169] as devices at low risk of interference, including alarms, identification systems, radio-determination, telecommand, telemetry, RFID, and detection, movement and alert applications.

[2]Note that previous ETSI regulations previously specifically referred to the use of Listen Before Talk – Adaptive Frequency Agility for this policy, but this has been generalised in newer regulations [169]. Band numbers have also recently been renamed.

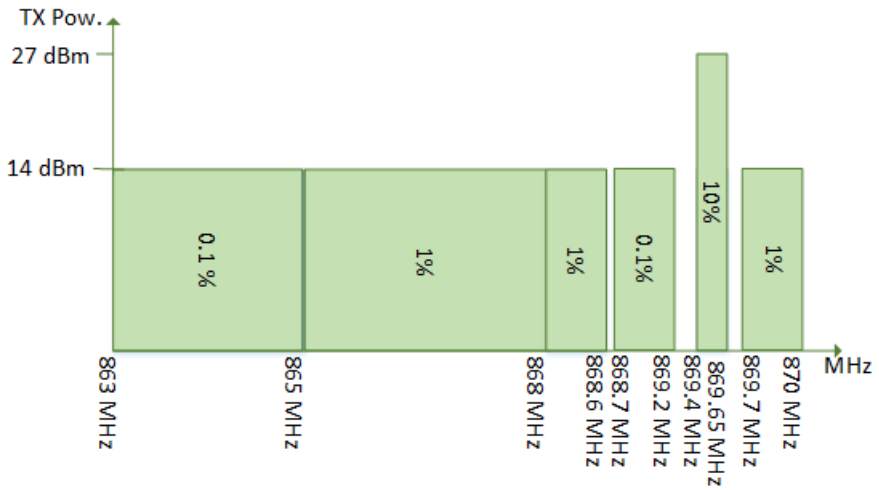


Figure A.3: Duty Cycle Regulations of the EU 868 ISM Band

LPWAN protocols can also be grouped into wideband or ultra-narrowband (UNB) technologies, where wideband techniques utilise a larger bandwidth than what is needed and use controlled frequency diversity to retrieve data, and UNB techniques compress data into ultra-narrow bands and use high stability Radio Frequency (RF) crystals and digital signal processing techniques to recover the data [171, 172].

An analysis of the literature has been performed to study the suitability of particular IoT applications for LPWAN. This was achieved through the identification of real implementations of particular use cases in the literature. The results of this are provided in Table A.2. The categorisation of use cases has been adapted from surveys and discussions of LPWAN [4, 11, 161, 173, 126]. The mapping of use cases to implementations described in published work is original to this thesis. For clarity, the references provided in the table have been separated from the main reference list for the thesis.

Table A.2: Implementations of LPWAN Use Cases, from the Literature

LPWAN Use Case	As in
Transportation	
Smart Parking	[243, 244]
Traffic Monitoring	[245, 246, 247]
Street Lighting Control	[248, 249, 250]
Self-Service Bike Rentals	[251]
Trash Collection	[252]
Fleet Tracking	[253, 254]
Vehicle Maintenance	[255, 256, 257]
Railways	[258]
Structural Monitoring	
Structural Health Monitoring	[259, 260, 261]
Heritage Preservation	[262, 263, 264]
Manhole Cover Monitoring	[265, 266]
Environmental Monitoring	
Air Quality Monitoring	[267, 268, 269] [270, 271, 272]
Water Quality	[273, 274, 275] [276, 277]
Temperature Monitoring	[272]
Animal Tracking	[278, 279]
River Monitoring	[280]
Smart Buildings	
Smart Lighting	[281]
Data Centre Monitoring	[282]
Indoor Air quality	[283]
Heating Control	[86]
Medical	
Health Monitoring	[284, 285, 286]
Telemedicine	[287]
Fall Detection	[288, 289]
Assisted Living	[290]
COVID-19	[291, 292]

LPWAN Use Case	As in
Utilities	
Waste Management	[252, 293] [294, 276]
Gas Usage Metering	[295]
Gas Pipeline Monitoring	[296]
Water Grid Management	[297]
Electricity Metering	[298, 299]
Smart Grids	[85, 300] [301, 302]
Wind Turbine Monitoring	[303]
Agricultural	
Livestock Monitoring	[25, 304] [305, 306]
Smart Greenhouses	[24, 307] [308, 309]
Crop Monitoring	[310, 311] [312, 47]
Soil Monitoring	[307, 306, 276]
Irrigation Systems	[313, 314]
Viticultural Monitoring	[307, 315, 316] [317, 318]
Meteorological Stations	[319]
Insect Monitoring	[320, 321, 322]
Emergency Services	
Earthquakes	[323, 324]
Forest Fire Detection	[325, 326] [327, 328]
Flood Monitoring	[329, 330, 331]
Landslide Detection	[332, 333, 334]
Smart Business	
Asset Tracking	[335, 336, 337]

Each of the technologies in the LPWAN paradigm will now be described, highlighting their key features and core similarities.

A.3 LoRaWAN

Amongst all LPWAN technologies, LoRaWAN has attracted the most attention in research and industry [174]. LoRaWAN is an upper layer protocol developed by the LoRa Alliance which operates on top of Long Range (LoRa), a physical layer technology developed by Semtech. LoRa is effectively a form of Chirp Spread Spectrum (CSS) with integrated Forward Error Correction (FEC). LoRaWAN networks are deployed in the sub-GHz ISM bands, and so must adhere to regional access regulations. Three different device classes enable a lower latency for downlink communications, at the expense

of a higher energy consumption rate. LoRaWAN functions on an operator-free model, simplifying and cheapening the management of infrastructure, and helping support a simple "out-of-the-box" deployment. Deploying a LoRaWAN network requires a NetID issued from the LoRa Alliance, or alternatively paying for use through a network provider with a private LoRaWAN network. Figure A.4 shows a map of countries with public LoRaWAN Network operators [162]. Note that this map does not include community-driven LoRaWAN public networks, such as the Things Network, which provides free access to LoRaWAN infrastructure by relying on volunteers to set up and deploy LoRaWAN gateways in their locality. This map does also not include networks maintained by public-private partnerships. For example, the largest LoRaWAN network in Ireland, Pervasive Nation, is publicly owned and accessible to researchers and businesses in the country.

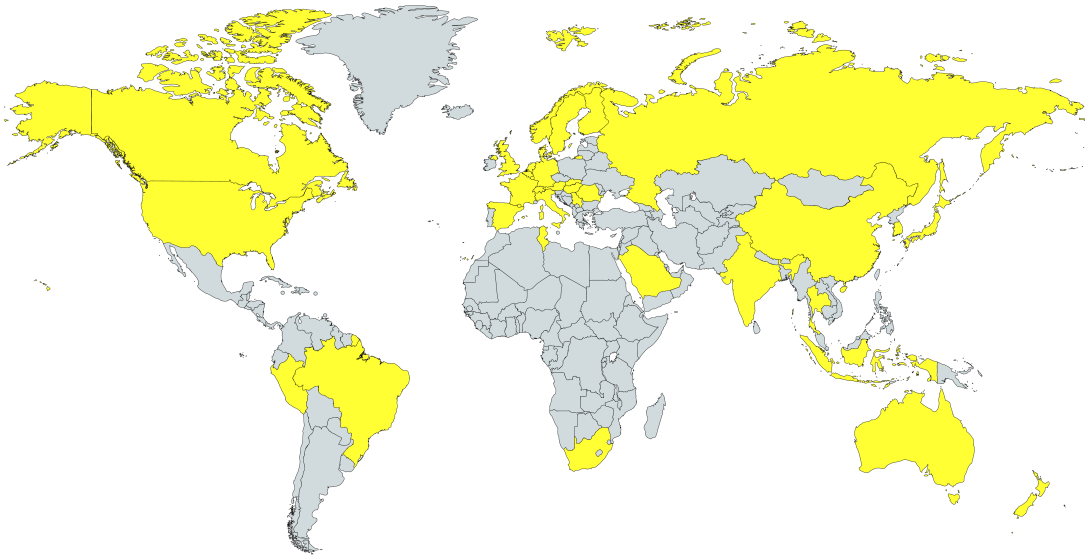


Figure A.4: Countries with Public LoRaWAN Network Operators

LoRaWAN supports secure, mobile, bi-directional communication for payloads up to 251 bytes. The LoRaWAN overhead per packet is 12 bytes. LoRa range depends on the link budget, which can be modified through choice of channel bandwidth, coding scheme, transmission power, carrier frequency, and spreading factor. A higher required link budget for successful transmission also results in higher energy consumption in the transmission per bit, meaning that the device lifetime and daily throughput are directly dependent on the distance from the device to the nearest gateway. LoRa range is up to 5 km in urban environments and over 10 km in rural environments [123, 175]. The transmission channel is chosen on a pseudo-random basis, based on current duty cycle allowances. The choice of spreading factor is optimised by the device based on the delivery of specifically (but infrequently) requested downlink frames, maximising the battery life of individual devices and increasing the overall network capacity. LoRaWAN is the LPWAN technology that has garnered the most interest in research and in community-driven networks, but questions remain about the ability of LoRaWAN networks to handle the traffic of the predicted range of LPWAN applications, at the scale per-gateway predicted by Semtech.

A.4 Sigfox

Sigfox's eponymous technology [176] is a proprietary, ultra-narrowband (UNB) approach, operating on the unlicensed sub-GHz ISM bands. Sigfox functions on an operator model where users subscriptions for each device to regional Sigfox-supported networks operated by network providers, who manage the network of gateways. A particular level of subscription defines the maximum number of uplink and downlink packets that that device can send each day. Transmitted data can be accessed by the customer through a web portal and rerouted through an Application Programming Interface (API) to a customer's own system [177]. Sigfox devices are typically cheaper to purchase than LoRaWAN or NB-IoT [178]. Sigfox organise the roll-out of infrastructure across the world in partnership with regional operators. Figure A.5 shows a map of countries with significant Sigfox coverage [179].

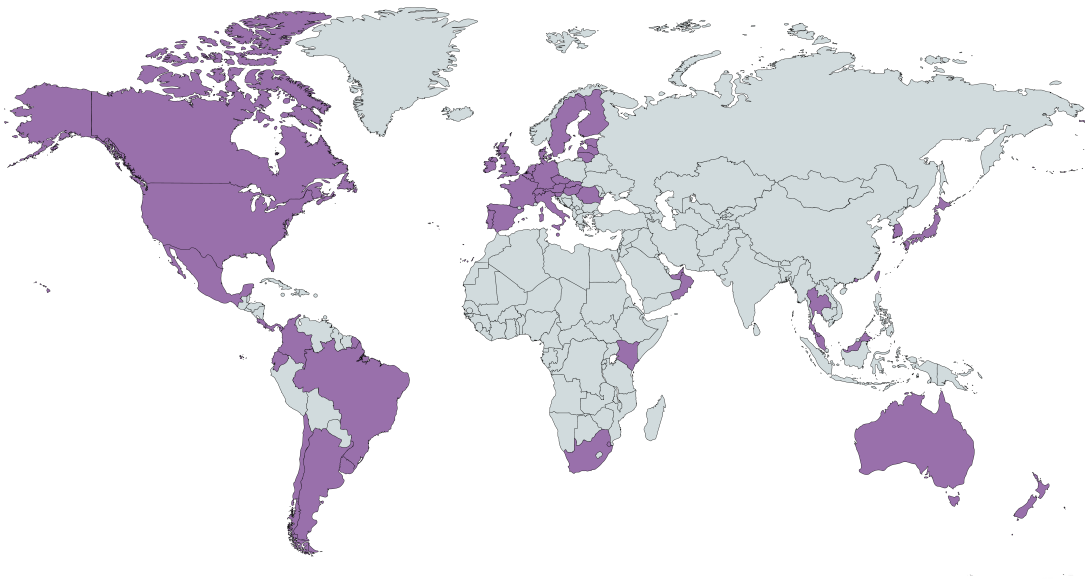


Figure A.5: Countries with Significant Sigfox Coverage

For uplink, Sigfox define 400 orthogonal 100 Hz channels in the 868 MHz band (from 868.180 MHz to 868.220 MHz) [180]. Forty of these channels are reserved. The noise level in each of these narrow bands (hence "ultra-narrowband") is very low, enabling the decoding of signals at the receiver. The sharing of the frequency space in this manner also increases the number of devices that can be supported. However, it also decreases the data rate [161]. For uplink, a Differential Binary Phase-Shift Keying (DBPSK) scheme operating at a fixed 100bps is used [171]. The receiver is capable of demodulating a very low received power signal (-142 dBm). A Sigfox base station can cover a range of 20-50 km in rural areas and 3-10 km in urban areas.

The channel access method of Sigfox is Random Frequency-Division Multiple Access (RFDMA) unslotted ALOHA with no channel pre-transmission sensing. When sending an uplink packet, a device randomly chooses three of the unreserved 360 channels and sends the packet to any base station in

range; the delivery of any of the three defines successful reception [11]. This redundancy helps to ensure delivery, as the very limited downlink traffic prevents the regular use of acknowledgement messages. Similarly to LoRaWAN, an unslotted ALOHA-based approach is taken to minimise control plane packets and enable low power operation [172]. The maximum payload size for a Sigfox uplink frame is 12 bytes, and the protocol overhead is 14 bytes.

Sigfox frames can be sent unidirectional, as described above, or bidirectional, where a receive window is also scheduled after transmission. The receive window opens 20 seconds after the device sends a message, and lasts for between 20.1 seconds and 44.5 seconds [36]. For downlink, a Gaussian Frequency Shift Keying (GFSK) scheme operating at 500 bps on a 600 Hz spectrum segment is used. Downlink transmissions are sent in the subband with the highest spectrum access limits (10%), providing further transmission opportunities for Sigfox gateways [181]. After the reception of a downlink frame, the recipient responds with an uplink confirmation frame. Sigfox provides much wider coverage and network accessibility than any other fully proprietary LPWAN solution, but questions remain of the long term future of the technology because of slow growth, losses in personnel, and the revenue-sharing approach taken with national operators [182].

A.5 NB-IoT

Narrowband Internet of Things (NB-IoT) is one of three solutions, along with EC-GSM-IoT and LTE-M, forming 3GPP's C-IoT, in anticipation of the development of the Internet of Things [37]. Whereas the other newly defined cellular technologies can be considered advances on previous work, NB-IoT can be considered a new track, with good co-existence performance but not fully backward compatibility with existing 3GPP technologies [183]. A clear distinction between cellular approaches and other LPWAN technologies is that these cellular approaches operate on licensed bands and so do not have to deal with the same duty cycle regulations as options transmitting in the sub-GHz ISM bands.

Essentially, NB-IoT is built from Long-Term Evolution (LTE), reducing functionalities to the minimum while enhancing IoT-related aspects. This takes the form of a modified acquisition process (different cell search process to LTE), reduced bandwidth requirements (180 kHz of bandwidth, in comparison to 1.4-20 MHz used in LTE), and a modified random access scheme. Enhanced coverage and reduced power consumption are achieved in exchange for relaxed latency, a lower data rate (around 250 kbps downlink and 20 kbps uplink [184]), and lower spectral efficiency. The price of the chip is also reduced, through the use of a narrower band [185]. Deployment of NB-IoT can be provided through a software update. Countries with NB-IoT operators, in various stages of national coverage, are shown in Figure A.6 [186].

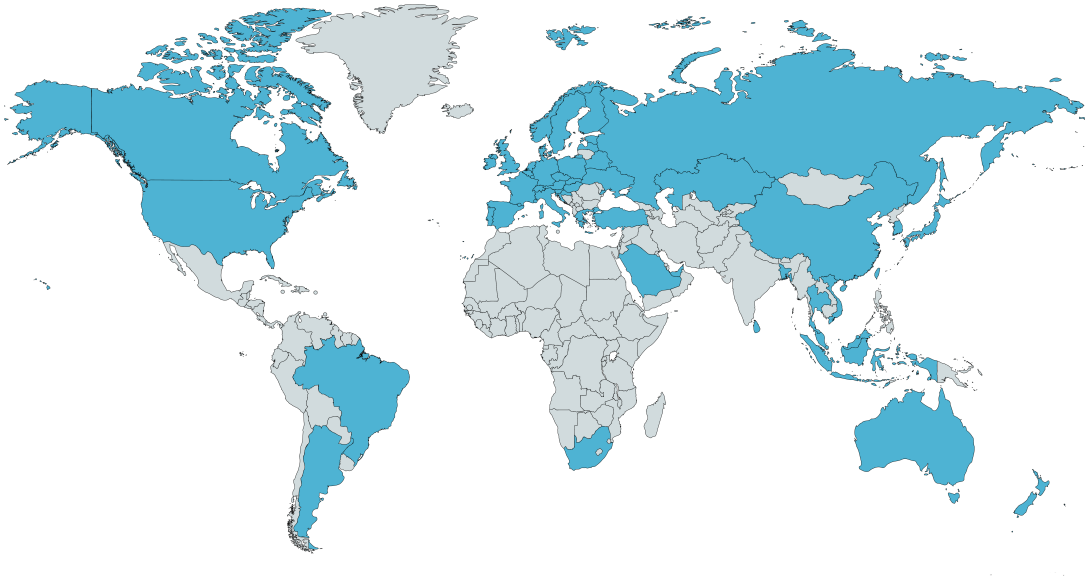


Figure A.6: Countries with NB-IoT Operators

NB-IoT supports 3 different deployment scenarios:

- *In-band operation*: deployed within a LTE wideband system, comprising 1 or more of the LTE Physical Resource Blocks (180 kHz). The transmit power at the base station is shared between wideband LTE and NB-IoT, and both technologies can be supported using the same base station hardware, without compromising the performance of either [183, 187].
- *Standalone*: deployed in a standalone 200 kHz of spectrum. All transmission power at the base station is used for NB-IoT, increasing coverage. Typical usage of this mode would be as replacement of GSM carriers.
- *Guard-band operation*: co-located with a LTE cell, placed in the guard band of a LTE carrier. This shares the same power amplifier as the LTE channel, and so shares transmission power [188].

The downlink of NB-IoT is based on OFDMA, with 15 kHz subcarrier spacing, and reuses the same OFDM numerology as LTE [183]. Both single-tone and multi-tone are supported in the uplink. Multi-tone is based on Single-Carrier Frequency-Division Multiple Access (SC-FDMA) with 15 kHz subcarrier spacing. With single-tone, sub-carrier spacing can be 15 kHz or 3.75 kHz [189]. NB-IoT achieves a 20 dB improvement over GPRS, giving a maximum coupling loss (MCL) of 164 dB [164]. NB-IoT targets covering 52 thousand devices per channel per cell. This is based on an estimation of 40 devices per household, in an area with the density of London [164, 190]. NB-IoT aims to enable a typical device lifetime of over ten years, on a battery capacity of 5 Wh. NB-IoT, like LTE, uses discontinuous reception (DRX), which avoids monitoring the control channel continuously in order to conserve energy. LTE has DRX cycles up to 2.56s. Release 13 introduced extended DRX (eDRX)

cycles for both idle and connected modes, which extend the cycles to 43.69 minutes and 10.24 seconds respectively [164], further increasing energy conserved. NB-IoT modifies LTE to provide a much more suitable wireless technology for LPWAN applications while extending coverage, but may not be able to provide the necessary super low power operation and low deployment and maintenance costs required by a subset of LPWAN applications.

A.6 LTE-M

LTE-Machine Type Communication (LTE-M), another 3GPP C-IoT technology, is also at its core a simplified version of LTE, reworked to reduce power consumption, simplify hardware, and increase coverage, while maintaining mobility [184]. LTE-M consists of a set of device categories, each with different achievable data rates and levels of low power operation. This section focuses on LTE-M CatM1, as this is the device category most suited to mMTC applications with low data rate requirements [191]. LTE-M CatM1 operates in licensed LTE spectrum, with the required bandwidth reduced from 20 MHz in LTE to 1.4 MHz, corresponding to six LTE resource blocks for transmission and two remaining as guard bands [192]. Countries with LTE-M operators, in various stages of national coverage, are shown in Figure A.7 [186].

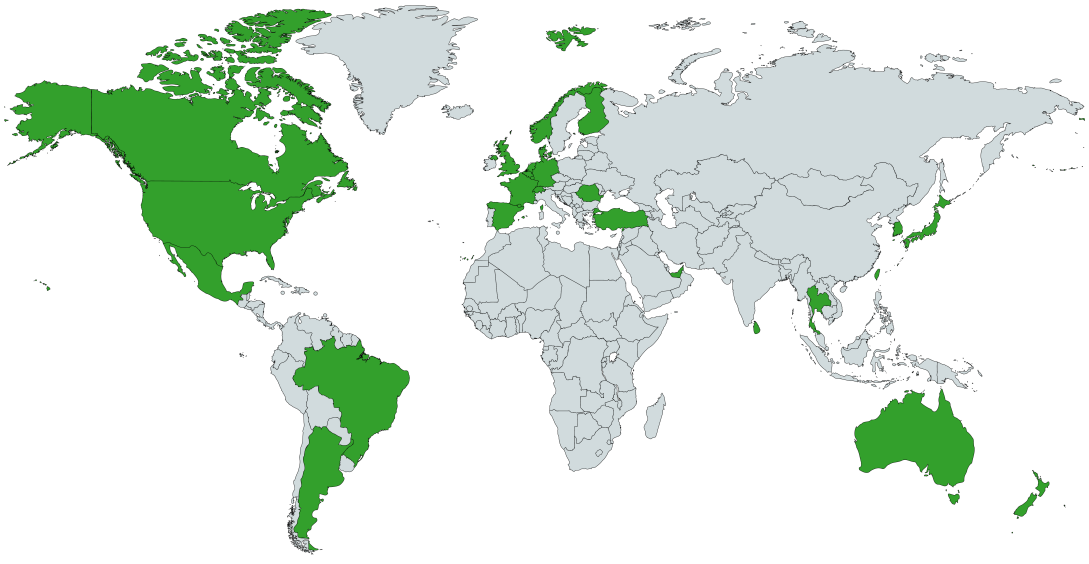


Figure A.7: Countries with LTE-M Operators

A maximum range of 11 km can be achieved with LTE-M CatM1, and the data rate is up to 1 Mbps. This high data rate in comparison to other LPWAN options enables the feasibility of firmware over-the-air-updates [193]. OFDMA for downlink, SC-FDMA for uplink, and the LTE transmission structure are maintained for LTE-M CatM1 [194]. eDRX is also supported, as well as a Power Saving Mode which allows devices to revert to an inactive low power state, enabling a

potential device lifetime of ten years. Also unique to LTE-M in the LPWAN space is the support of Voice over LTE [184]. A capacity of up to 100 thousand devices per base station is supported, depending on the throughput requirements of the underlying applications [4]. Deployment is through an infrastructure software upgrade, with no hardware modifications required [184]. LTE-M CatM1 achieves an MCL of 155.7 dB, in comparison to 140.7 dB for LTE, ensuring extended coverage for IoT applications [192]. LTE-M CatM1 provides a higher data rate as well as potential coverage wherever LTE coverage is already available, enabling LPWAN applications with stricter requirements on latency and data throughput. However, similarly to NB-IoT, LTE-M CatM1 may not be able to compete with unlicensed LPWAN technologies when super low power operation and low deployment and maintenance costs are a primary factor.

A.7 EC-GSM-IoT

Extended Coverage - GSM - Internet of Things (EC-GSM-IoT), aka EC-GSM, is the third C-IoT technology in development by 3GPP [37]. It is designed as an enhancement to GSM, and re-uses the current GSM design (eGPRS) whenever possible, only making changes that are necessary in order to enhance LPWAN-related requirements, that is, high capacity, long range, and low energy. The re-use of GSM design means that upgrades to GSM networks can be provided with a software upgrade, and support for new devices can be achieved in existing GSM deployments. In addition, already deployed GSM units will not be adversely affected with the deployment of EC-GSM-IoT devices, as traffic from legacy GSM devices and EC-GSM-IoT devices can be multiplexed on the same physical channels, since the multiplexing principles from GSM are carried over to EC-GSM-IoT [195]. EC-GSM-IoT uses 200 kHz of bandwidth per channel, for a total system bandwidth of 2.4 MHz.

On the downlink physical layer, the design is for the most part the same as current GSM. The primary difference is that a new packet control channel format has been designed to limit the amount of control signalling required. On the uplink physical layer, this new control channel format is also used, along with an overlaid Code-Division Multiple Access (CDMA) (on the uplink data, uplink control signalling, and extended coverage random access channels) to increase capacity, enabling multiple devices to transmit on the same physical channel simultaneously [37]. Additionally, a separate random access channel is defined for devices that require extended coverage. Beyond this, the design follows GSM principles.

Extending the coverage of GSM is achieved through the use of blind repetitions. Different coverage classes are defined, with different numbers of total blind transmissions for different logical channels. 50,000 devices can be supported per cell. The data rate of EC-GSM-IoT varies from 350 bps to 70 kbps, depending on the coverage class. All power classes available for GSM devices are available for EC-GSM-IoT. The typical power class used is 33 dBm. An additional lower power class of 23 dBm has also been defined, enabling the integration of the power amplifier onto the chip, providing longer lifetime and reducing cost in exchange for a shorter range. The coverage achieved in EC-GSM-IoT is an MCL of 164 dB [195] for the 33 dBm power class and an MCL of 154 dB for the 23 dBm

power class. Power Saving Mode, which was defined in Release 12, and eDRX are also supported on EC-GSM-IoT devices, further increasing energy efficiency. In addition, EC-GSM-IoT supports a relaxed idle mode behaviour, where no cell measurements are performed while in a Power Saving State [37].

The battery life of EC-GSM-IoT nodes is estimated at 10 years with a 5 Wh battery, depending on several factors including the distance of the device from the base station, the number of bytes required to send per day, and the power class used. As an example, a device using the 33 dBm power class, providing a coverage of 154 dB, and sending 50 bytes every 2 hours is predicted to last over 14 years. Though EC-GSM-IoT could prove useful for enabling IoT applications in areas using GSM, interest in EC-GSM-IoT is considerably less than for the other two 3GPP LPWAN technologies, because of the recent and continuing decommissioning of GSM networks by various network operators around the world [184]. There are currently no operational EC-GSM-IoT networks deployed.

A.8 Other LPWAN Solutions

A number of other LPWAN solutions are also in development, which are described here. The distinction of the "primary" and "other" technologies in the LPWAN paradigm is subjective; in this work the distinction is made based on the availability of information sources and verifiable results. For each of the technologies described in this section, there is comparatively little technical information and published comparative analysis about each protocol.

A.8.1 Nwave

Nwave's [196] eponymous protocol is, like Sigfox, based around UNB communications in the sub-GHz unlicensed ISM bands, and operates in a star topology. The primary target application for the protocol is smart parking. According to the developers, Nwave nodes can cover 10 km in urban environments, and 30 km in rural, primarily as a result of advanced de-modulation techniques at the receiver. Additionally, devices can operate for 20 years on a single AA lithium battery, providing a data-rate of 100bps. However, little technical information is available about the protocol.

A.8.2 Ingenu

Ingenu's Random Phase Multiple Access (RPMA)-based [197] protocol is a Direct-Sequence Spread Spectrum (DSSS) solution which operates on the 2.4 GHz ISM band. Use of the 2.4 GHz band results in competition with Wi-Fi and IEEE 802.15.4 standards and increased propagation loss, but also enables a higher available maximum transmission power. In addition, there are no duty cycle regulations to be followed in the 2.4 GHz band, enabling a much greater time-on-air and increased opportunities for downlink communications. Ingenu operate on a public network business model. RPMA is a variation of CDMA, and optimises the choice of spreading factor based on the downlink

signal strength [184]. The base station is capable of receiving at all spread factors and delay times simultaneously. All current Ingenu infrastructure is located in the United States. As with Nwave, little technical information is available about the protocol and no scientific study featuring the protocol is available.

A.8.3 Weightless-P

Weightless [198] are a set of LPWAN technologies defined and managed by the Weightless Special Interest Group (SIG). Three different standards have been proposed by the group: Weightless-N, which is focused on ultra-low cost, Weightless-W, which occupies part of the spectrum formerly used by TV whitespace, and Weightless-P, which focuses on high performance. This section will focus on Weightless-P, as it is the most newly defined standard and is most similar to the other LPWAN technologies covered in this section. Like Sigfox, Weightless-P is a narrowband approach on the sub-GHz ISM bands. Weightless-P splits the spectrum into 12.5 kHz channels. On the physical layer, standard Gaussian Minimum Shift Keying (GMSK) and offset-Quadrature Phase Shift Keying (QPSK) modulation are used. According to the SIG, flexible channel assignment, adaptive data rates (from 200 bps to 100 kbps), and time-synchronised base stations enable the efficient use of spectrum, minimisation of transmit power usage, and prior scheduling of resources, optimising the battery life of individual devices as well as network resources. The support of acknowledgement of all transmissions, FEC, and Automatic Repeat Request (ARQ) help to maintain reliability and Quality-of-Service (QoS). Weightless-P can support a typical range of 2 km in urban environments and all traffic is encrypted using AES-128/256. However, there is no available analysis or direct comparison to LoRaWAN or Sigfox available in the literature. Weightless claim low latency in both uplink and downlink, enabling the support of over-the-air firmware upgrades. Weightless is supported by the core members of the Weightless SIG: Accenture, ARM, M2COMM, Sony-Europe, and Telensa.

A.8.4 Telensa

Telensa [199] also provide an UNB solution in the sub-GHz unlicensed ISM bands. According to the company, a Telensa base station can connect to up to 5000 nodes, and cover 2 km in urban areas and 4 km in rural. Individual nodes continue to function as programmed (in smart lighting applications) even if the connection to the base station is lost, and have an estimated lifetime of 20 years. Additionally, unlike most LPWAN technologies, the protocol can provide fully bi-directional communication, and so is suitable for control as well as monitoring. However, no analysis of the protocol in the literature is available.

Telensa's solution is the most mature available, the company having been founded in 2005. Telensa have already deployed millions of nodes in over 50 smart city networks worldwide, mostly in the United Kingdom but also in cities such as Shanghai, Moscow, and Sao Paulo. Telensa networks are deployed in over 30 countries worldwide. The company themselves deploy smart lighting and smart parking applications, and also provide a platform that companies can leverage in creating their own

smart city applications, enabling authorities to invest in and control the smart city platform for their own city. Telensa is also a member of the Weightless SIG board. Telensa is currently aiming to standardise the protocol through ETSI [168].

A.8.5 WAVIoT

WAVIoT [200] is an Infrastructure-as-a-Service LPWAN-solutions provider from Houston, Texas. Their solution, NB-Fi (Narrowband Fidelity) is a narrow band protocol which communicates on the sub-GHz ISM subbands. NB-Fi separates the 500 kHz band into 5000 channels, and each signal is transmitted in 50 Hz of bandwidth with a minimum bit rate of 50 bod using DBPSK. According to the company, WAVIoT gateways can provide -154 dBm of receiver sensitivity, and cover over 1 million nodes. On WAVIoT-developed devices, short bursts of data use 50 mA of current, and in idle mode, a few μA are used. Devices have a lifetime of up to 20 years, and a 176 dBm link budget. NB-Fi is an open standard, in that WAVIoT will work with interested parties to develop custom devices that utilise the NB-Fi protocol. WAVIoT support three different network types: public, private (city-wide deployment), and enterprise (campus-wide deployment).

NB-Fi operates on a star topology, and can achieve a coverage of over 16 km in an urban environment and over 50 km in a rural environment. The average uplink latency is 30 s, and the average downlink latency is 60 s. NB-Fi is a full-stack technology, covering the PHY layer up to the application layer. Similarly to Sigfox, data sent through the gateways is stored on a cloud server, and can be accessed from an IoT platform and easily rerouted and manipulated through use of an API. All data is encrypted bidirectionally from the device to the server using an XTEA-256 bit key. As with most of the other technologies in this section, there is no analysis of the protocol available in the literature.

A.8.6 Dash 7 Alliance Protocol

The **Dash 7 Alliance Protocol** [201, 202] (known as Dash7 or D7AP) is a protocol designed for wireless sensor network applications being developed by the Dash 7 Alliance. The PHY and MAC layer basis, D7A, originated from the ISO 18000-7 standard [4]. D7AP defines a full-stack protocol for D7A, including the application and presentation layers. D7AP is deployed in the unlicensed sub-GHz ISM bands, primarily in the 433 MHz band [203]. The presentation layer forms a file system; data transmission is in the form of writing to or reading a remote file, and nodes are described with and can be assigned properties, which can be used along with identifiers in the grouping of requests of remote data for different applications. An API is provided to enable interaction with D7AP networks over any interface.

D7AP networks are formed from endpoints, subcontrollers, and gateways [203]. Gateways remain active continuously, collecting data from endpoints and relaying it back to the server. Subcontrollers have the functionality of gateways but are designed to operate at a lower power and have sleep cycles, with the main function being to relay data from endpoints to gateways. In this way, D7AP networks utilise a tree topology, or, without the use of subcontrollers, a star. Endpoints can send data directly

to a gateway or subcontroller, or alternatively send an all-cast or any-cast, where the device waits for acknowledgements from all or at least one gateway respectively. In this way, mobile applications are supported as devices can communicate with any available gateway [204]. Endpoints also have the ability to send data to each other, and gateways can also query data from endpoints. Endpoints can transmit (asynchronously) to the gateway at any time, and wake up periodically to listen for downlink transmissions. D7AP provides three different defined data rates: 9.6 kbps, 55.555 kbps, and 167 kbps, and the achievable range for D7AP is up to 10 km, depending on the environment and data rate [205]. The channel access method is Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA). The modulation scheme used is 2-GFSK, and 1/2 FEC encoding is available. The maximum packet size is 256 bytes. Similarly to IEEE 802.15.4, AES-CBC is used for authentication and AES-CCM for authentication and encryption.

A.9 Direct Comparison

Tables A.3 and A.4 provide a direct comparison of the previously outlined LPWAN technologies. Where available, a value for the range is provided for both the urban (U) and rural (R) case. Overall, some traits are common to all LPWAN technologies. Each technology utilises some form of a star topology, with D7AP being the only exception as the inclusion of repeater nodes is explicitly defined in the protocol specification. The maximum data rate (DR) for the uplink is also generally low. There is also great variance in the maximum supported uplink data rate. However, it should be noted that unlike other technologies which would support Human-Type Communications (HTC) traffic (such as IEEE 802.11 and LTE), for mMTC applications there is generally a set amount of data to send per device in a given timeframe. Coupled with the low power requirements of LPWAN, this means that the maximum data rate is less important than the energy consumption per bit. For LPWANs, a faster data rate may correlate with a lower energy consumption as the device will not have to stay active for as long to send the same amount of data, but this is not guaranteed.

Table A.3: Direct Comparison of LPWAN Technologies, Part I

Protocol	**Topology**	**Max DR**	**Frequency Band**	**MAC Layer**	**Range (U/R)**
LoRaWAN	Star	50 kbps	sub-GHz ISM	Unslotted ALOHA	5 km/15 km
Sigfox	Star	100 bps	sub-GHz ISM	RFDMA	10 km/50 km
NB-IoT	Star	50 kbps	LTE & GSM	FDMA/OFDMA	15 km (R)
LTE-M CatM1	Star	1 Mbps	LTE bands	FDMA/OFDMA	11 km (R)
EC-GSM-IoT	Star	70 kbps	GSM bands	TDMA/FDMA	15 km (R)
Nwave	Star	100 bps	sub-GHz ISM	Unknown	10 km
Ingenu	Star	8 kbps	2.4 GHz ISM	RPMA	4 km
Weightless-P	Star	100 kbps	sub-GHz ISM	TDMA/FDMA	5 km
Telensa	Star	62.5 bps	sub-GHz ISM	Unknown	1 km (U)
WAVIoT	Star	100 bps	sub-GHz ISM	Unknown	16.6 km
D7AP	Star/tree	166 kbps	sub-GHz ISM	CSMA/CA	0-5 km

Table A.4: Direct Comparison of LPWAN Technologies, Part II

Protocol	Est.	Modulation	Prop. Aspects	Deployment Model	Encryption
LoRaWAN	2015	CSS	PHY	Priv. & Oper.	AES 128-bit
Sigfox	2009	DBPSK	PHY & MAC	Operator	Configurable
NB-IoT	2016	QPSK	Intern. Standard	Operator	3GPP
LTE-M CatM1	2016	16QAM	Intern. Standard	Operator	3GPP
EC-GSM-IoT	2016	GMSK	Intern. Standard	Operator	3GPP
Nwave	2010	UNB	Full Stack	Private	Unknown
Ingenu	2008	DSSS	Full Stack	Private	AES 128-bit
Weightless-P	2012	GMSK	Open Standard	Private	AES 128-bit
Telensa	2005	UNB 2-FSK	Full Stack	Private	Unknown
WAVIoT	2011	DBPSK	Full Stack	Priv & Oper.	XTEA
D7AP	2013	2-GFSK	Open Standard	Private	AES-CCM

One of the defining features of LPWAN technologies is the use of the sub-GHz ISM bands, in Europe, in particular the EU 868 MHz band. The band is also used by IEEE 802.11ah, a lot of IEEE 802.15.4-based solutions, and many short range devices. There is no overall control of access to the EU 868 MHz band, beyond the duty cycle regulations. The cellular IoT options are deployed in licensed spectrum, and so do not have any restrictions on access. Ingenu is unique amongst LPWAN protocols in that it is deployed solely at 2.4GHz. LoRaWAN has also recently added support for transmission at 2.4GHz. Figure A.8 shows a taxonomy of LPWAN technologies relative to choice of spectrum. The MAC layer of LoRaWAN and Sigfox are defined by the restrictions of the 868 MHz band, and use a simple ALOHA-based access method without any sort of LBT system. This does enable low power operation, reducing control plane packets and enabling asynchronous nodes. However, even factoring in the duty cycle regulations, the potential scalability of networks deployed using this system is limited as the probability of collision increase greatly as the network scales. Some of the other non-cellular solutions do not provide details on the MAC layer, but since synchronisation packets are generally not required it can be reasonably speculated that ALOHA-based MAC layers are used in these as well. On the other hand, C-IoT options have dedicated control channels, and a centralised control of the frequency space used, enabling greater management of scalability. However, this is typically at the cost of a higher access cost due to the dedicated spectrum.

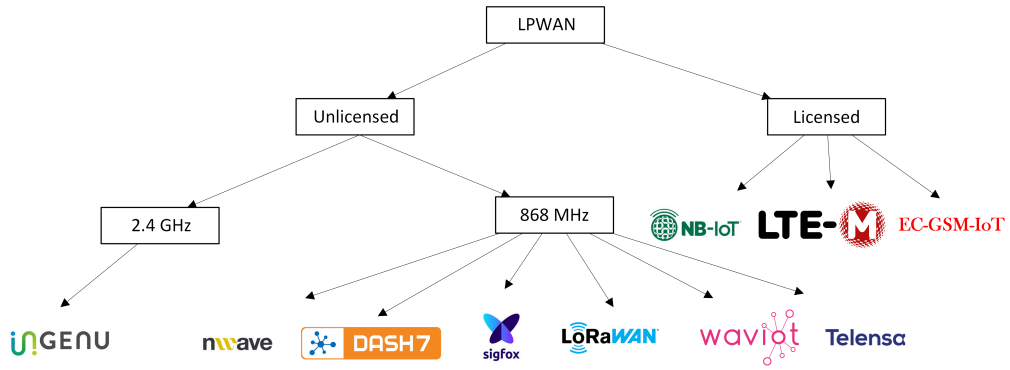


Figure A.8: A Taxonomy of LPWAN Technologies

The primary differentiator for unlicensed LPWAN on the physical layer is the use of wideband or narrowband modulation. Telensa, Sigfox, WAVIoT, Weightless-P and Nwave use ultra narrowband modulation on the uplink, where the used spectrum is split into many very narrow channels and the gateway nodes utilise robust signal processing algorithms in order to receive sent frames. On the other hand, LoRaWAN and D7AP utilise the robustness achieved through use of the spread of a signal through a wider channel in order to increase the possible coverage.

The majority of LPWAN solutions have proprietary aspects. Weightless-P and D7AP are the only fully open unlicensed protocols. The physical layer of LoRaWAN, LoRa, is proprietary, but has been significantly reversed-engineered in research. The Cellular IoT options constitute international standards. The proprietary LPWAN protocol developers also generally provide private deployment of networks as a service. The exceptions are D7AP and LoRaWAN, where networks are not deployed by the developers themselves (the Dash7 Alliance and Semtech, respectively). LoRaWAN has the additional factor of the development of community-driven LoRaWAN networks, where network access for prototypes and non-commercial systems are provided for free by volunteers.

A.10 Conclusions

This appendix has defined LPWAN as a concept, and has provided an overview and comparative analysis of LPWAN protocols. The analysis shows that some features are common to LPWAN technologies (star topology, long range, low data rate) and that there are others which define the differentiation between technologies (modulation format, deployment model, frequency band). The C-IoT options are the most differentiated from the others because of the availability of dedicated spectrum, and the use of shared infrastructure with current cellular standards.

Overall, there are factors which define the limits of performance for unlicensed LPWAN, which are common across almost all of the technologies: the achievable range, the shared channels, the duty cycle regulations, the lightweight MAC, the limited guarantee-able reliability, and the low cost requirement. Analysis and the development of new features for any of these unlicensed LPWAN protocols will require the consideration of the unique intricacies of the protocol, but the existence of so many common features indicates that developments for one protocol will potentially be applicable to others, with some minimal adaption. The next appendix will narrow the focus to LoRaWAN, highlighting the key features which define the achievable performance of the protocol.

APPENDIX B

A Review of LoRa & LoRaWAN

In this appendix, an in-depth explanation of the LoRa and LoRaWAN protocols is provided. A thorough understanding of the protocol and review of the research enables the identification and prioritisation of potential areas for enhancements and new work. The protocol will be discussed in terms of the key metrics: scalability, energy efficiency, throughput, and reliability. This appendix is divided into three main sections. As the design, overall behaviour and, ultimately, performance of a LoRaWAN system is dependent on the underlying physical layer, this appendix begins with a focus on LoRa. The second part focuses on LoRaWAN, which enables an upper layer stack for use with LoRa. The final third section discusses efforts related to the simulation of LoRa and LoRaWAN.

B.1 LoRa

LoRa is a proprietary physical layer technology developed and maintained by Semtech. However, reverse-engineering efforts have revealed much of the inner workings of the protocol [206, 207, 208]. Figure B.1 shows the typical LoRaWAN network stack. In LoRaWAN, the device is deployed at a particular frequency, and has to follow the regulations of the subband defined by an organisation such as ETSI. This influences the behaviour of the LoRa PHY layer, which in turn influences the behaviour of the LoRaWAN protocol. Finally, running on top of the LoRaWAN layer is the user-defined Application layer.

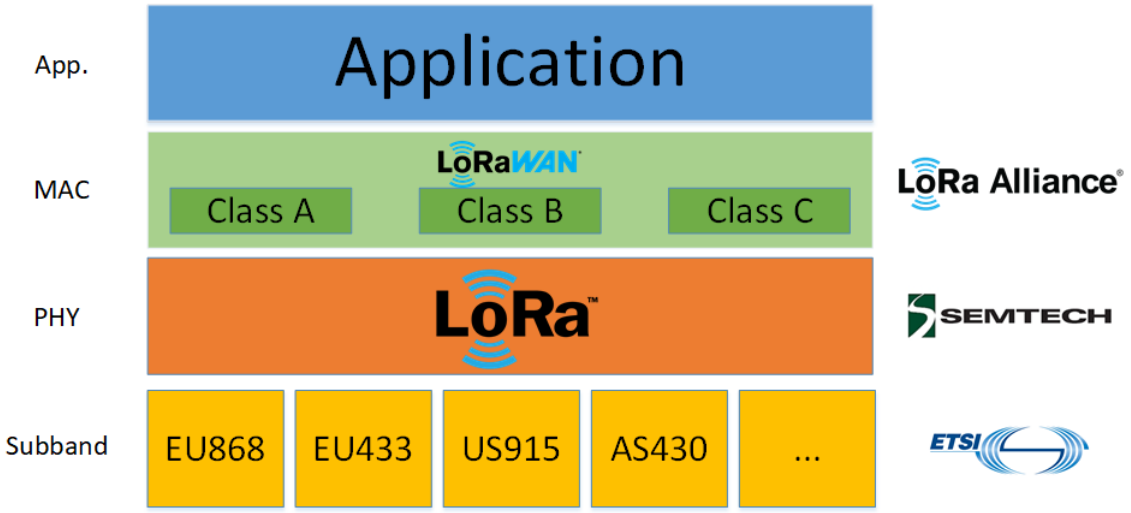


Figure B.1: The LoRaWAN Network Stack [209]

The LoRa physical layer defines the modulation format which enables long range communication at a low power. The major parameters of LoRa determine the reliability, time-on-air, and energy consumption of individual LoRaWAN transmissions. Therefore, an understanding of LoRa is key in understanding the realistic achievable performance of LoRaWAN networks. In this section, the key aspects of the LoRa protocol are explained, namely the modulation format, the spreading factors, the data encoding, and the frame detection method. As LoRa is based on Chirp Spread Spectrum, the section begins by explaining the fundamentals behind Spread Spectrum and the chirp waveform.

B.1.1 Spread Spectrum

Spread Spectrum, at its core, is a modulation method that results in the use of a transmission signal bandwidth that is much larger than needed to send the information. Though it would be normal practice to minimise the required transmission bandwidth when designing a modulation scheme, there are benefits to increasing the transmission bandwidth beyond what is strictly necessary. Spread spectrum signals are resistant to multipath fading [210]. The wide bandwidth of spread spectrum signals also enables location and timing acquisition [211]. Spread spectrum signals are also resistant to jamming, as the power of the jammer is forced to either be located in a subspace of the transmission, or also be distributed across the entire spectrum [212]. The most commonly used two forms of spread spectrum communications are Direct-Sequence Spread Spectrum (DSSS), for example in IEEE 802.11b, and Frequency Hopping Spread Spectrum (FHSS), for example in Bluetooth. CSS, which LoRa is based on, is a third, less commonly used form of spread spectrum communications.

In DSSS and FHSS, each device is assigned a unique spreading code, which takes the form of a binary sequence. These spreading codes are independent of the data to be sent by the device [212]. The codes are pseudorandom but are generated deterministically, and have several identifiable properties,

namely that the number of 1s and 0s is approximately the same, the number of consecutive 1s or 0s is small, the periodicity is large, the correlation in shifted versions of the sequence is low, and there is very low cross-correlation between any two generated sequences [212]. The sequence is then used to generate a square pulse wave, with pulses of the duration of the chip time. The generation of sequences with these properties leads to spreading codes which are approximately orthogonal to the other used codes [211]. Use of spread spectrum thus enables multiple users to share the same bandwidth with minimal interference as the spreading codes are designed to be approximately orthogonal [210]. The resulting transmissions are spread orthogonally in time (in DSSS) or frequency (in FHSS) using the spreading code by the transmitter, and despread by the intended receiver using the same code.

As an example, a generalized form of DSSS is as follows [211]:

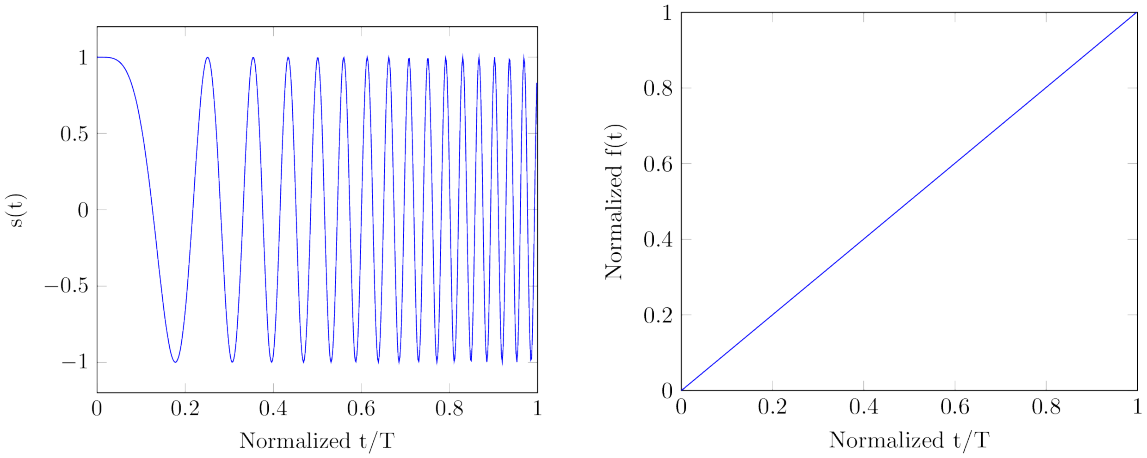
1. The data symbols are first linearly modulated to form the baseband modulated signal.
2. The modulated signal is multiplied (convolution, in the frequency domain) by the spreading code with chip time T_c, after which it is upconverted by the carrier. The chip rate is much faster than the rate of the modulated signal and thus the original signal is spread across a much wider bandwidth.
3. The signal passes through the channel, introducing narrowband interference, noise, and multipath components.
4. At the receiver, the signal is downconverted to baseband, with the carrier recovery loop locking to the carrier associated with the minimum delay multipath component.
5. Cross-correlation with the same spreading code used at the transmitter despreads the signal and recovers the modulated message.

B.1.2 Chirp Spread Spectrum

The main disadvantages of DSSS are that the system requires a highly accurate clock source, and the synchronisation and correlation of spreading codes with the received signal is expensive in terms of energy efficiency [213, 214]. A cheaper form of spread spectrum can be achieved through the use of the chirp waveform, which maintains resistance to multipath distortion [214], jamming [214], and the Doppler effect [19] while being cheaper to spread and despread the signal. A modulation format based on such a system is known as CSS. CSS systems have previously been used in military and radar applications [215], and as mentioned in Appendix A are also a PHY layer option in IEEE 802.15.4 at 2.4 GHz, since introduced in the IEEE 802.15.4a amendment.

A chirp signal takes the form of upchirps and downchirps. The transmission of a linear up-chirp waveform of duration T begins at the lowest frequency of the used channel (f_{min}), and changes at a constant rate ($\mu(t)$, which is positive for an up-chirp) over the transmission interval of the chirp until reaching the highest frequency of the channel (f_{max}). Similarly, a linear down-chirp begins at the highest frequency of the used channel, and changes at a constant rate (which is negative) over the

transmission interval of the chirp until reaching the lowest frequency of the channel. The number of times per second that the phase is adjusted is the chip rate. The signal and the frequency evolution of an upchirp can be seen in Figure B.2.



(a) s(t) (b) f(t)

Figure B.2: A CSS Upchirp

Thus, a chirp waveform can be written as [172, 214, 216]:

$$x(t) = a(t) * e^{j\phi(t)} = a(t) * e^{j2\pi(s\frac{BW}{2T}t^2 + f_c t)} \tag{B.1}$$

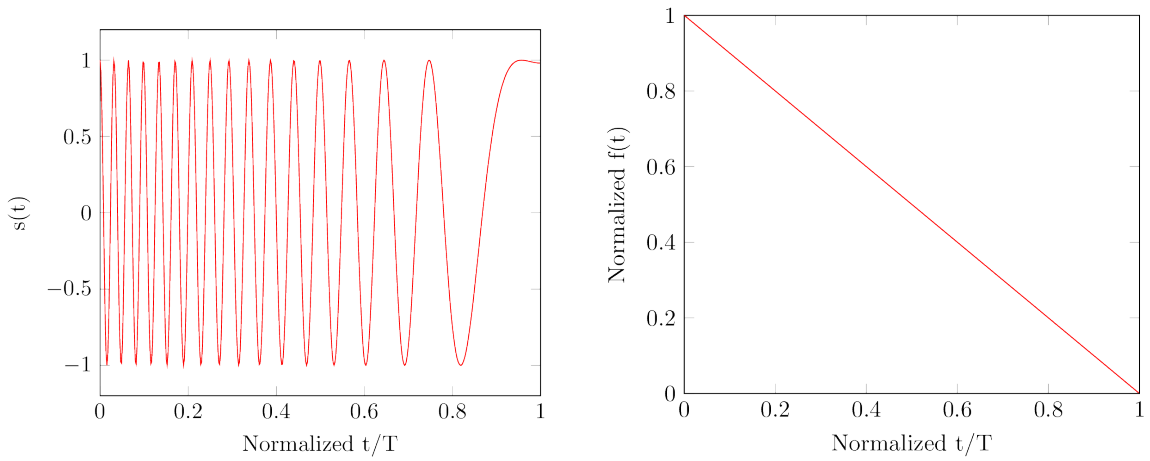
where $\phi(t)$ is the chirp phase, f_c is the central frequency of the channel, BW is the bandwidth of the channel and is $f_{max} - f_{min}$, T is the chirp length, and $a(t)$ is the envelope of the signal which is zero outside of the chirp duration T. s defines whether a linear chirp is an upchirp or a downchirp, and is 1 for an upchirp, and -1 for a downchirp. The instantaneous frequency is, assuming a system with linear chirps, thus:

$$f(t) = \frac{1}{2\pi}\frac{d\phi}{dt} = f_c + (s * \frac{BW}{T} * t) \tag{B.2}$$

And so the chirp rate is based on the bandwidth of the channel, the chirp length, and chirp sign, and is:

$$\mu(t) = \frac{df}{dt} = \frac{1}{2\pi}\frac{d^2\phi}{dt^2} = s * \frac{BW}{T} \tag{B.3}$$

and is the rate of change of the instantaneous frequency, and is positive for up-chirps and negative for downchirps. The equivalent downchirp to Figure B.2 is shown in Figure B.3. Note that the above is simply a description of the chirp waveform, not CSS. Modulation schemes which are created based on this waveform are called CSS.



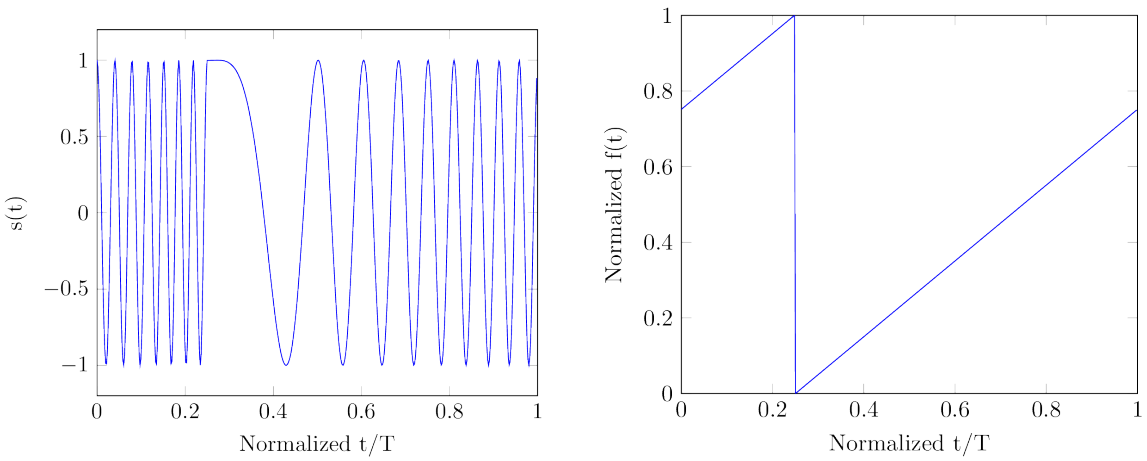
(a) s(t) (b) f(t)

Figure B.3: A CSS Downchirp

The required complexity of the receiver design is greatly reduced in comparison to DSSS, as for each chirp the impulse response for the matched filter is simply the same chirp but with opposite chirp rate sign, and the timing and frequency offsets between transmitter and receiver are equivalent [217]. In addition, as with DSSS, the transmission of the signal across a wide bandwidth averages out the noise and thus reduces the impact of impulsive noise [218]. However, this reduction in complexity comes at the expense of assigned orthogonal spreading codes for individual devices. As we will see, in LoRa, orthogonality is still achieved, but only between devices using different data rates. This is because orthogonality is achieved through varying the length of chirps, rather than through the use of spreading codes.

B.1.3 The LoRa Frequency Shift

The LoRa implementation of CSS extends the waveform described above to include a shift of the starting frequency of the chirp [74]. Use of a spreading factor (SF) results in chirps that are 2^{SF} chips long, and so the number of possible shifts is 2^{SF}; it is in this shift that data is encoded. Thus, for LoRa, the T variable defined in the previous section is equal to 2^{SF}. In LoRa, aside from in the LoRa preamble, transmissions take the form of upchirps only. For a single chirp, transmission starts at a frequency f, then increases at a linear rate $\mu(t)$ until reaching f_{max}, then wraps around starting at f_{min} and increases until reaching frequency f. The signal and the frequency evolution of an example LoRa chirp can be seen in Figure B.4. Correlation with a base CSS down-chirp at the receiver enables identification of the original shifted chirp [74]. Each different possible frequency-shifted chirp is equivalent to a LoRa symbol, and has a direct unique mapping to a number of bits.



(a) s(t) (b) f(t)

Figure B.4: A Frequency Shifted CSS Upchirp (the LoRa Modulation), with a Value of 8

When the shifted up-chirp is multiplied with a base down-chirp of the same chirp rate (which is the complex conjugate of the base up-chirp), in the absence of noise, the destructive superposition of the two waveforms results in a pure sinusoid with an unvarying frequency, as only the constant and linear phase terms remain. Then, the discrete Fourier transform (DFT) (which is 2^{SF} bins wide) will provide a peak at just one bin. The index of this bin corresponds to the original modulated data; if a base up-chirp is multiplied by a base down-chirp, the peak will be at 0, if the up-chirp has been frequency-shifted by 1 the peak will be at 1, etc. The result of the DFT of the signal shown in Figure B.4 after multiplication with a base downchirp is shown in Figure B.5. Thus in the absence of excessive noise and interference, the original bits can be extracted by taking the *argmax* of the DFT, resulting in an integer in the range of 0 to $2^{SF} - 1$ [19, 217]. The reason for this is that in LoRa the shifted symbols are a set of cyclically shifted versions of a base Zadoff-Chu sequence [216]. In a Zadoff-Chu sequence, the periodic autocorrelation is orthogonal for all shifted replicas of the base sequence, resulting in a non-zero value only at the position which corresponds to the shift. The complexity of this DFT-based approach to identification of the chirp is $O(NlogN)$ [219]. Alternatively, the correlation of the received symbol with every possible symbol can be computed, and the *argmax* taken to find the matching symbol and the original bits. The complexity of this approach is $O(N^2)$, and requires storage of a look-up table of all of the possible samples of the up-chirp [219].

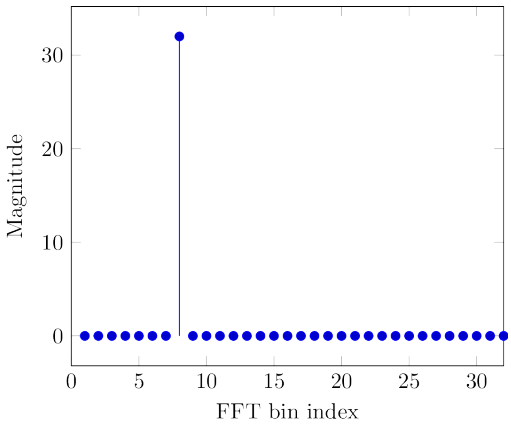
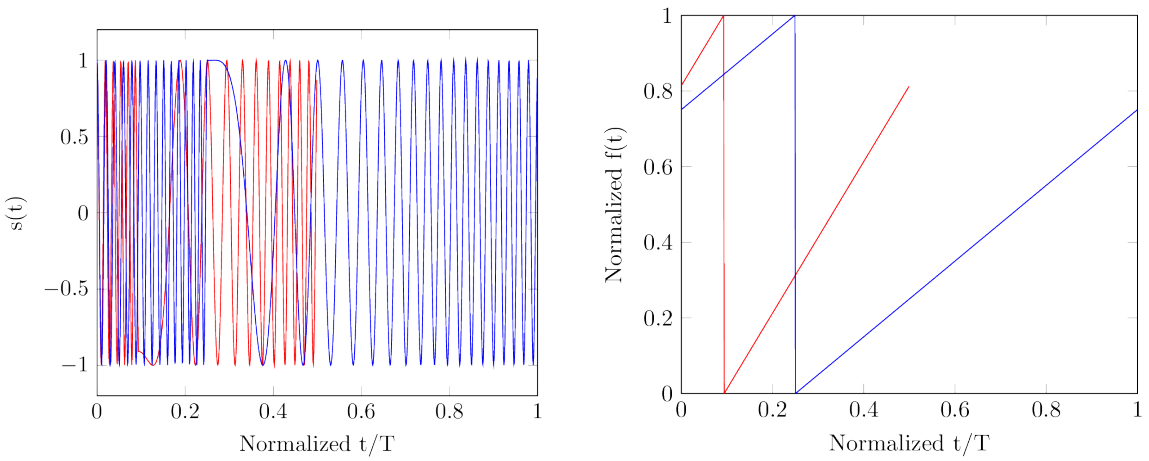


Figure B.5: DFFT of the Multiplied Result of a Frequency Shifted Upchirp with a Shift of 8, using SF5, with an Unshifted Downchirp of the Same Length

B.1.4 LoRa Spreading Factors

As mentioned, LoRa features a set of spreading factors, which define the number of chips in a chirp and thus effectively determine the angle of the chirp (i.e. $\mu(t)$) [19]. Use of a spreading factor of SF results in a chirp that is 2^{SF} chips long. Thus for two transmissions using $SF(x)$ and $SF(x+1)$, the chirps of the latter transmission would contain twice the number of chips. Figure B.6 shows the signal and frequency evolution of two chirps using two adjacent spreading factors; note that the chirp length for one is twice as long as the other. Spreading factors ranging from 6 to 12 are available for use. Increasing the duration of the chirps increases the signal-to-interference-plus-noise ratio (SINR) of the signal but also the length of time of the transmission; thus choice of spreading factor is a trade-off between coverage and data rate. The differing gradients of spreading factors enables the receiver demodulator to distinguish between transmissions using different spreading factors in the same channel i.e. spreading factors are mostly orthogonal to one another [19, 220], and multiple users using different data rates can share the same bandwidth with minimal interference.



(a) s(t) (b) f(t)

Figure B.6: Two Frequency Shifted Upchirps of Different Spreading Factors (Encoding 6 and 16 Respectively)

The LoRa spreading factor also determines the number of bits represented by a single chirp. As there are 2^{SF} chips in a chirp, there are 2^{SF} different possible frequency shifts of the chirp for each spreading factor. This is the key to the modulation of LoRa: each different possible frequency shift of an unmodulated (unshifted) chirp represents a different value between 0 and $2^{SF} - 1$. Thus, a single LoRa chirp represents SF data bits. The LoRa modulation encodes data onto a chirp by frequency shifting the chirp exactly once. Table B.1 below shows the bits per chirp, chips per chirp, and hence chips per bit for each spreading factor.

Table B.1: Relationship Between Spreading Factor, Chirp and Chip

Spreading Factor	**Bits per chirp** ($=SF$)	**Chips per chirp** ($=2^{SF}$)	**Chips per bit** ($=2^{SF}/SF$)
6	6	64	10.66
7	7	128	18.29
8	8	256	32.00
9	9	512	56.89
10	10	1024	102.40
11	11	2048	186.18
12	12	4096	341.33

The length of an individual chip is inversely proportional to the bandwidth of the channel i.e. a LoRa transmission in a channel of 125 kHz of bandwidth has a chip rate of 125,000 chips/sec, and so the length of a chip is $\frac{1}{BW}$ [213]. Thus overall the duration of a single LoRa chirp is dependent on both the spreading factor and the bandwidth and is calculated as:

$$T_{chirp} = \frac{2^{SF}}{BW} \quad [\text{s}] \quad \text{(B.4)}$$

and thus the chirp rate is:

$$R_{chirp} = \frac{1}{\frac{2^{SF}}{BW}} = \frac{BW}{2^{SF}} \quad \text{[chirp/s]} \quad \text{(B.5)}$$

which is equivalent to Equation B.3, and finally the modulation bit rate is:

$$R_{bit} = SF * \frac{1}{\frac{2^{SF}}{BW}} \quad \text{[b/s]} \quad \text{(B.6)}$$

The extended chirp length affects the sensitivity of the receiver and enables a greater range for higher spreading factors, according to the formula [213]:

$$S = -174 + 10log_{10}(BW) + NF + SNR \quad \text{[dBm]} \quad \text{(B.7)}$$

where S is the sensitivity of the LoRa receiver, (-174) is due to the thermal noise at the receiver, NF is the noise figure for the receiver (and for common LoRa transceivers such as the SX1276, is equal to 6 dB), and SNR is the signal-to-noise-ratio required for the modulation, and is SF-dependent and displayed in Table B.2 [213, 221, 76]. As can be seen in the table, the expected increased sensitivity at the receiver from use of the next-slowest data rate is 2.5 dB.

Table B.2: LoRa Demodulation SNR vs Spreading Factor

Spreading Factor	**Chips per symbol**	**LoRa Demodulator SNR**
6	64	-5 dB
7	128	-7.5 dB
8	256	-10 dB
9	512	-12.5 dB
10	1024	-15 dB
11	2048	-17.5 dB
12	4096	-20 dB

This formula and table can be used to generate Table B.3, which shows example bit rates for each LoRa spreading factor, for set bandwidths. The bandwidth of channels can be configured on LoRa devices, but 125 kHz width channels are predominantly used, with 250 kHz and 500 kHz less frequently used. As can be seen in the table, there is a tradeoff between bit rate (and thus energy efficiency), and range. Notice that while the increase in symbol duration is exactly linear, the decrease in bit rate is not, because of the increased number of bits that can be represented in a single symbol for slower data rates. The range of LoRa has been shown to be up to 10 km in semi-urban areas [123]. LoRa coverage is significantly reduced in dense urban areas, where Rayleigh fading characteristics predominate [222].

Table B.3: LoRa Bit Rates

Spreading Factor	**Bandwidth (kHz)**	**Symbol Duration (s)**	**Bit rate (b/s)**	**Sensitivity (dBm)**
7	500	0.000256	27343.750	-118
7	250	0.000512	13671.875	-122
6	125	0.000512	11718.750	-121
7	125	0.001024	6835.938	-125
8	125	0.002048	3906.25	-128
9	125	0.004096	2197.266	-131
10	125	0.008192	1220.703	-134
11	125	0.016384	671.387	-136
12	125	0.032768	366.211	-137

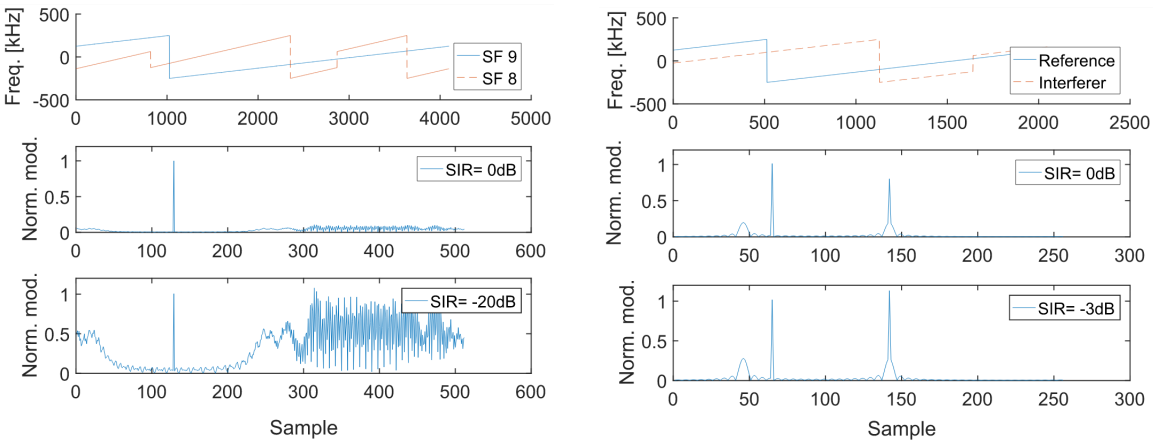
The differing chirp angles of the different spreading factors enables the design of LoRa transceivers that can receive packets from different spreading factors simultaneously on the same channel. A typical LoRa device may only detect packets on a single pre-set channel and SF. However, a LoRa gateway may detect packets across multiple channels and SFs simultaneously, including multiple preambles on different SFs on the same physical channel. A typical LoRa gateway can receive up to eight packets simultaneously. As the vast majority of traffic in LoRa-based applications is in the uplink, this results in up to six virtual channels being available for every physical channel in a network (as use of $SF6$ is typically not permitted by the network operator). Collisions on a channel from two transmissions sent using different spreading factors is only a factor when the difference in receive power is sufficiently high. Additionally, uplink and downlink transmissions do not collide because of the use of I/Q inversion by the gateway when transmitting [159]. Table B.4 provides results from [220], which shows the difference in signal power required to cause a collision between spreading factors.

Table B.4: LoRaWAN Co-channel Rejection Table [220]

SF_{ref} / SF_{int}	**6**	**7**	**8**	**9**	**10**	**11**	**12**
6	0	-8	-10	-11	-11	-11	-11
7	-11	0	-11	-13	-14	-14	-14
8	-14	-13	0	-14	-16	-17	-17
9	-17	-17	-16	0	-17	-19	-20
10	-19	-19	-19	-19	0	-20	-22
11	-22	-22	-22	-22	-22	0	-23
12	-24	-24	-24	-25	-25	-25	0

In addition, it has also been found that there is a strong capture effect in LoRa transmissions, meaning that in the case of a collision in both spreading factor and channel, there is a high chance that the stronger signal received can be correctly demodulated [220]. The capture effect of LoRa enables a lower Packet Error Rate (PER) and greater maximal load than an equivalent traditional ALOHA network [223]. In [224], it is shown that six times more traffic can be sent within a single cell LoRa network than in an equivalent ALOHA network, due to the capture effect. The average co-channel rejection for LoRa spreading factors is -16 dB [220], as in the DFT of the intended received

symbol, if the interfering signal is transmitted using a different SF (which fits a different sized bin), the interfering signal has a lower spectral density, and unless the interfering signal has been received with a much higher power, the original intended signal still provides the highest peak in the DFT and thus the symbol can still be successfully decoded [220]. This is demonstrated in Figure B.7 (a), from [220].



(a) Impact of the Capture Effect when the Interfering Signal is using a Different Spreading Factor

(b) Impact of the Capture Effect when the Interfering Signal is using the Same Spreading Factor

Figure B.7: Impact of the Capture Effect on LoRa Reception [220]

For interfering transmissions using the same SF, if the preamble of the intended reception has been successfully received, then the DFT of the intended symbol shows a peak in one bin, while the reception of the out of sync receiver is split across two partial symbols, resulting in two smaller peaks at two DFT bins. So, if the two signals have been received at the same power (an $SINR$ of $0dB$), the intended packet can still be successfully received [220]. This is demonstrated in Figure B.7 (b), from [220]. This effect holds and remains consistent for all SFs [225]. If the interferer is neither chip- nor phase-aligned with the intended signal, the probability of receiving the intended signal is even higher [219]. Additionally, LoRa has also been shown to have a high resistance to the Doppler effect [226].

B.1.5 LoRa Encoding: Cyclic Redundancy Check, Error Detection, Whitening, Interleaving, and Gray Encoding

The format of the LoRa modulation has been explained. Now, the process in which a series of data bits is encoded into a series of LoRa symbols is discussed. First, an optional 16-bit cyclic redundancy check (CRC) is appended to the payload for error detection. Then, Hamming Code error detection and correction is used to improve the robustness of the signal. Coding rates of 4/5 to 4/8 are available, where 4/5 and 4/6 provide error detection (like a parity bit), and 4/7 and 4/8 provide single bit error correction. 4/8 additionally provides dual error detection [208]. The bit rate

equation B.6 can thus be redefined as:

$$R_b = SF * \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} \quad \text{[b/s]} \quad \text{(B.8)}$$

to include the redundancy introduced by the coding [213]. Table B.5 shows the actual bit rate with different encoding options, for $SF7$ and $SF12$. These results can be compared to the bit rates shown in Table B.3.

Table B.5: LoRa Bit Rates Including Coding, With a 125kHz Channel

Spreading Factor	**Coding Rate**	**Actual Bit rate (b/s)**
7	4/5	5468.750
7	4/6	4557.292
7	4/7	3906.250
7	4/8	3417.969
12	4/5	292.969
12	4/6	244.141
12	4/7	209.263
12	4/8	183.105

Next, the sequence is XORed with a whitening sequence, to reduce the correlation between bits introduced by the channel encoding [208] and provide more features for clock recovery [221]. Finally, interleaving is performed to provide a deterministic shuffle of the bits of the sequence, to reduce the impact of impulsive noise [206]. The sequence is then separated out into chunks of SF bits. Each chunk is converted to a LoRa symbol, which is an individual frequency-shifted chirp. The mapping of SF-sized chunks to symbols follows Gray coding.

B.1.6 The LoRa Preamble and Header

As mentioned previously, synchronisation is required by the receiver in order to delimit the starting point of the chirps of a transmission. This is provided by the LoRa preamble. Without synchronisation, the imprecision in timing will result in the energy of a chirp being split between adjacent DFTs (in incorrect bins) and reception will fail [208]. A LoRa preamble consists of at least eight base up-chirps, followed by two frequency-shifted up-chirps, and finally by two and a quarter base down-chirps [220]. The number of base up-chirps is programmable and is used for frame detection; a frame is detected if enough consecutive DFTs have the same $argmax$ (synchronisation is not needed as the timing offset for all of the base up-chirps will be the same) [208]. The two frequency-shifted upchirps are the sync word, and used for frame synchronisation, and to identify the network and prevent gateways from unnecessarily receiving packets from another LoRa network [224]. The preamble ends with two base down-chirps, used for frequency synchronisation. Following this, a timing of a quarter of a chirp is left in silence, to let the receiver align the time [220].

Since the preamble is used to provide synchronisation for the receiver, interference affects the frame

reception differently depending on when the interference starts. In [220], it is shown that if the last six symbols of the preamble are received correctly, the receiver can synchronise. Furthermore, if an interfering frame starts after the preamble has been received, and the interfering frame has the same or lower Received Signal Strength Indicator (RSSI) as the intended frame, then the intended frame can be successfully received. If the RSSI is higher, then the intended frame will be received with an incorrect CRC. Overall, interference has a much lower effect on successful packet reception if it occurs after the correct reception of the preamble [170].

LoRa transmissions can optionally include a LoRa PHY header, which contains the length of the payload in bytes, the FEC code rate of the payload, and indicates the presence of a CRC for the payload. This header is transmitted using the highest code rate, 4/8, and contains a CRC computed only on the header. If the parameters contained in the header are known by the receiver in advance, the header may be omitted [224]. However, the LoRaWAN protocol mandates the use of the LoRa PHY header in every transmission [227].

B.1.7 Transmission in the ISM Bands

As discussed in Section A.2, LoRaWAN is frequently deployed in the sub-GHz ISM bands, and so regional regulations must be adhered to by devices. LoRaWAN does not utilise any channel sense mechanism before transmit, so the "polite spectrum access" policy does not apply to the vast majority of LoRa deployments. LoRa devices in Europe may also be deployed in the 433 MHz ISM band, and more recently in the 2.4 GHz ISM band [228].

B.1.8 A LoRa Transmission, Overall

To summarise, the entire LoRa transmission process is as follows. This can be compared to the general DSSS format provided at the start of this appendix. For a given payload of PL bits, a spreading factor SF, a bandwidth BW, and a coding rate CR:

1. Regular transmissions may only occur on a channel at a rate which maintains adherence to the duty cycle regulations of the subband.
2. The CRC check is computed, and appended to the payload.
3. FEC in the form of Hamming Codes is applied to the payload, increasing the robustness of the final signal but also increasing the length of the payload to $PL * \frac{4+CR}{4}$ bits.
4. Whitening is applied to the payload to reduce the correlation between bits introduced by the encoding.
5. Interleaving is applied to the payload to reduce the impact of impulsive noise.
6. If used, the LoRa header is prepended to the payload.

7. The resulting data is divided into bit chunks of size SF. Each chunk represents an integer between 0 and $2^{SF} - 1$.

8. Each chunk is converted to a modulated LoRa chirp by introducing a time shift of $t = Gray^{-1}(i)\frac{T}{2^{SF}}$ to an unmodulated base up-chirp.

9. The signal, including the prepended LoRa preamble, is upconverted by the carrier and passes through the channel of bandwidth BW, introducing narrowband interference, noise, and multipath components. The bit rate of the part of the signal containing the payload is $SF * \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}}$.

10. At the receiver, the LoRa preamble (starting with unmodulated chirps) is first detected, minimising the phase offset of later received chirps.

11. Then, each received chirp is downconverted to baseband and multiplied by a conjugate base chirp of the same spreading factor and bandwidth (i.e. an unmodulated downchirp), which enables identification of the frequency-shift of each chirp and, by the opposite procedure of what is described above, the retrieval of the original payload. The integrity of the payload is ensured through the use of FEC and the CRC.

The LoRa modulation enables transmission of signals over a long distance while maintaining a cheap design for the receiver, by spreading the signal over a wide bandwidth and encoding information on frequency shifted chirps. Individual LoRaWAN links can trade off range and energy efficiency through the choice of spreading factor, which results in the transmission of longer chirps. The decrease in the data rate for use of each subsequent spreading factor is not quite linear, as while the length of a chirp is $\propto 2^{SF}$, a chirp also encodes SF bits. Overall, the configurable features of LoRa which impact the performance of a LoRaWAN link are the selected physical channel (and channel bandwidth), the spreading factor, the coding rate, and the preamble length.

B.2 LoRaWAN

This section provides an explanation of LoRaWAN, which defines an upper layer for use with LoRa. As previously mentioned, LoRa is a proprietary technology developed and owned by Semtech. In comparison, LoRaWAN is an open protocol which is developed and maintained by the LoRa Alliance, a non-profit organisation of over 500 telecommunications companies, equipment manufacturers, startups and universities. The LoRa Alliance also manages the certification of LoRaWAN devices. The LoRaWAN board of directors features representatives from Semtech, Cisco, Kerlink, and Orange, amongst other companies. This section will provide an overview of the LoRaWAN protocol, including the network structure defined by the protocol, as well as the three different device classes, the defined LoRaWAN data rates, and LoRaWAN MAC layer header.

B.2.1 Network Architecture

LoRaWAN networks are organised in a star-of-stars topology, where devices communicate directly with a central LoRaWAN Network Server, with LoRaWAN gateways acting as relay nodes. The Network Server then forwards arriving packets from devices to appropriate Application Servers. A separate Join Server manages the storage and maintenance of security keys for connected devices. Thus a single LoRaWAN network can feature many different concurrently running applications from different organisations, without security issues. The LoRaWAN network architecture is shown in Figure B.8. The LoRaWAN protocol on top of a LoRa wireless transmission provides the link between the LoRaWAN device and gateway. The received LoRaWAN packet is then forwarded by the gateway to the central Network Server, through some Internet Protocol (IP) link, either a wired or cellular connection. Since LoRaWAN gateways act purely as relays, a single transmission can be received by multiple gateways in the network, and duplicates are filtered out at the Network Server.

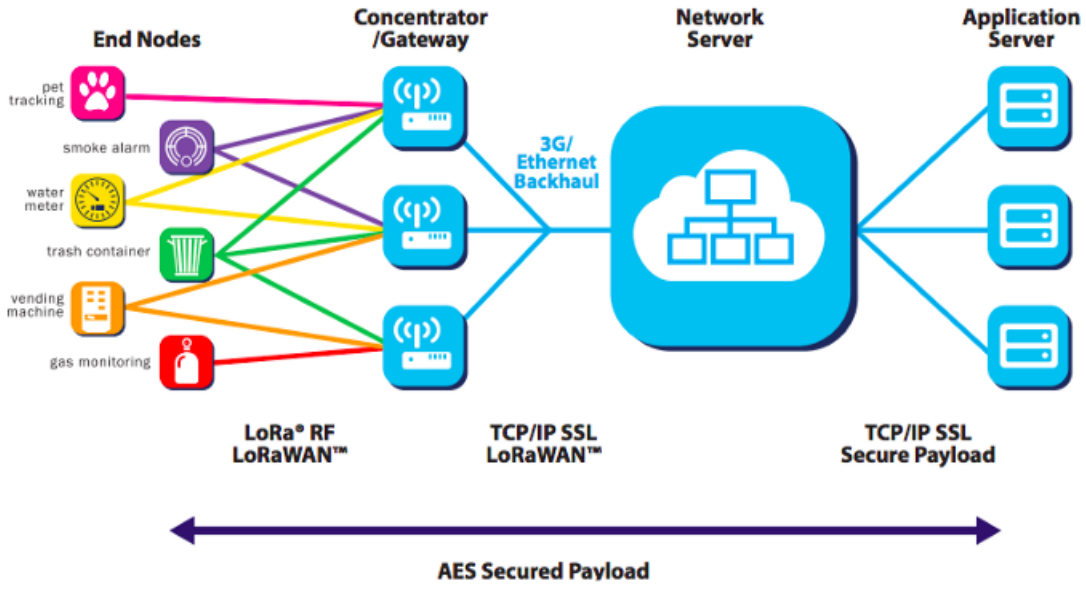


Figure B.8: The LoRaWAN Network Architecture [209]

Whereas other LPWAN solutions such as Sigfox and WavIoT are structured such that the core developing company also features as the major LPWAN network provider, LoRaWAN takes a different approach where LoRaWAN networks can be deployed by anyone. Smaller scale networks can be deployed by purchasing a LoRaWAN gateway and setting up an instance of a LoRaWAN Network Server, the software for which has been released open-source by Semtech. Alternatively, access can be gained through use of a public or private LoRaWAN network operator, such as Orange, the Things Network, or Pervasive Nation. Alternative approaches to LoRaWAN Network structure, in particular the use of multi-hop, have been suggested by many researchers. Proposed approaches include tree-based, cluster-based, and flooding-based network structures. An overview of proposed systems can be found in [229] and [230]. A typical LoRaWAN gateway is the Multitech Conduit, which has models providing support for subbands in the EU, US, Japan, and Australia. A photograph of this

gateway can be seen in Figure B.9.

Figure B.9: Multitech Conduit LoRaWAN Gateway

A typical development kit for LoRaWAN is the B-L072Z-LRWAN1 from ST, which enables transmission of LoRa, Sigfox, and FSK signals. A photo of this device can be seen in Figure B.10.

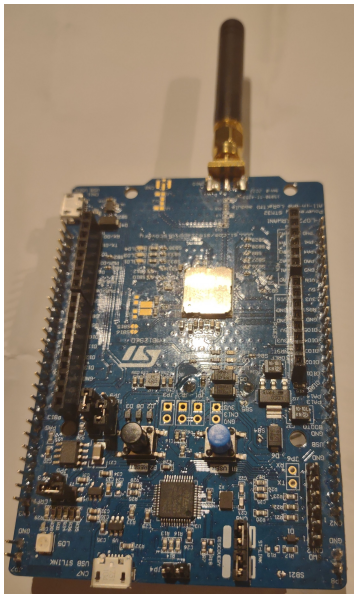


Figure B.10: B-L072Z-LRWAN1, from ST

B.2.2 LoRaWAN Data Rates

LoRaWAN delineates the spreading factor and bandwidth options of the LoRa protocol into defined data rates. Each data rate also specifies a maximum permissible payload size. For the EU 868 MHz band, the data rates are outlined in Table B.6. As described previously, the spreading factor parameter of these data rates provides a trade-off between coverage and bit rate, and thus has a significant influence on the device lifetime for a particular required device throughput ($> 50\%$, comparing the fastest and slowest data rates [231]).

Table B.6: LoRaWAN Data Rates

Data Rate	Spreading Factor	Bandwidth	Max Payload Size
0	12	125 kHz	59 bytes
1	11	125 kHz	59 bytes
2	10	125 kHz	59 bytes
3	9	125 kHz	123 bytes
4	8	125 kHz	250 bytes
5	7	125 kHz	250 bytes
6	7	250 kHz	250 bytes

The optimal data rate search function for LoRaWAN is the Adaptive Data Rate (ADR) algorithm, which is described in detail later in this section.

B.2.3 The LoRaWAN Overhead

The LoRaWAN overhead per packet is 13 bytes, which consists of a MAC header, a Frame header, an FPort, and a Message Integrity Check (MIC) which is calculated across the contents of the payload using a secure key, to ensure message integrity. The format of the LoRaWAN headers is shown in Figure B.11.

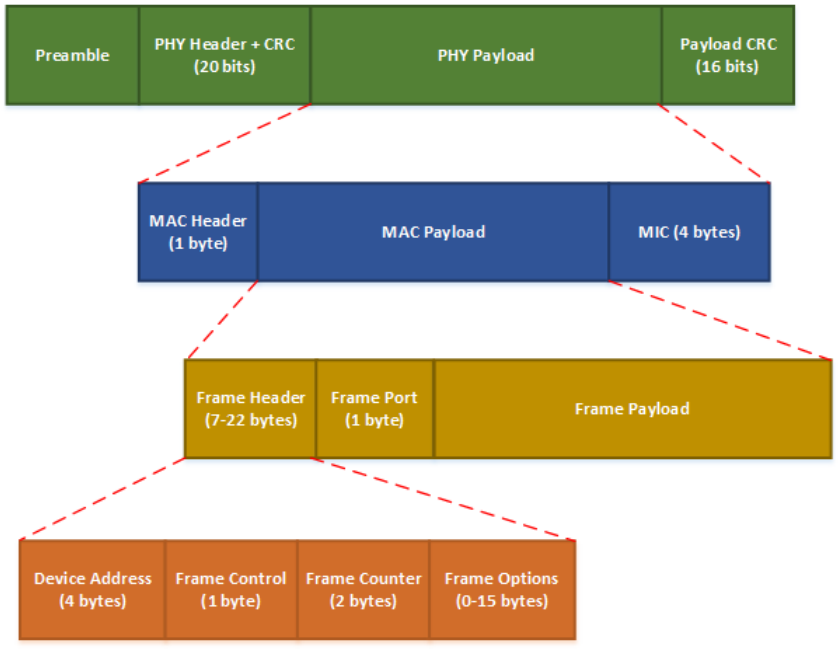


Figure B.11: Format of the LoRaWAN Headers

Contained within the LoRaWAN headers are:

- *Message Type*: Defines whether this frame is being sent in confirmed mode, where an acknowledgement for the frame is expected, or unconfirmed mode, where one is not. Also declares

whether this is an uplink or downlink frame.

- *DevAddr*: the device address, which is the address of the sender for uplink traffic, and the address of the intended recipient for downlink traffic.
- *ACK*: whether this frame is an acknowledgement or not.
- *ADR*: whether the ADR algorithm is turned on for this device.
- *Class B*: whether this device is operating in Class B mode or not (more details on this are provided later in this appendix).
- *FPending*: whether there is more data to be sent from the Network Server to this device (downlink only).
- *FCnt*: a frame counter, used to filter out multiple receives of the same frame.
- *FOpts*: contains MAC layer commands, which can be piggybacked inside regular LoRaWAN frames, and used to modify link parameters.
- *FOptsLen*: defines the number of bytes of MAC commands contained within FOpts for a frame.
- *FPort*: defines the particular application this frame is for, to enable routing to the correct Application Server by the Network Server.

This set of headers is common to every LoRaWAN transmission sent after joining the network. With regard to performance and scalability, the key parts of the header are the *ADR*, *ACK*, and *Class B* bits. When the *ADR* bit is set to 1, the data rate of a device will be set to a suitable choice based on the recent traffic. When the *ACK* bit is set to 1 on the uplink, a device is requesting a downlink response from the Network Server. Since the duty cycle limits of the sub-GHz ISM bands also apply to the gateways, misuse of this parameter can severely impact the performance of the network. When the Class B bit is set, the device will operate in a mode which has a higher energy consumption but also enables greater flexibility for downlink communications. Each of these features will be discussed in further detail later in this appendix.

The LoRaWAN protocol separates devices into three distinct classes. Class A supports device-initiated uplink and downlink traffic. Class B extends this by providing support for applications which require server-initiated communication and a deterministic latency on downlink communications, as regular receive windows are scheduled for potential downlink traffic. Class C is for devices without strict energy efficiency-constraints, as devices remain in continuous receive mode. An overview of each class will now be provided.

B.2.4 Class A

LoRaWAN Class A is the only class which *must* be implemented on all devices [227]. LoRaWAN Class A enables sporadic uplink data transmission for energy-constrained devices. Uplink transmission is via ALOHA without any LBT mechanism. Downlink transmissions are restricted as only in

response to uplink frames. Class A devices consume the least amount of energy, as when the device is not currently transmitting or waiting for a downlink response, the device will typically remain in sleep mode.

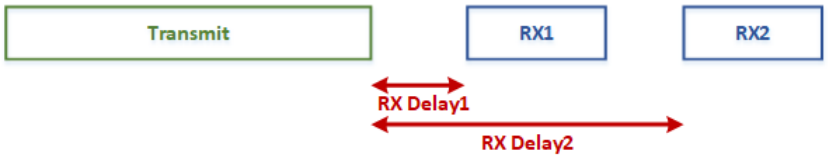


Figure B.12: LoRaWAN Class A Transmission

The format of a Class A transmissions is shown in Figure B.12, with time in transit mode shown in green and the time in receive mode shown in blue. Following every Class A LoRaWAN transmission the device opens two mutually exclusive receive windows. In the first receive window (RX1), the device waits for a downlink frame on the same channel as the preceding uplink frame, and using a data rate calculated as a function of the data rate of the preceding uplink frame and a configurable parameter, named $RX1DROffset$. In the second receive window (RX2), the device waits for a downlink frame using a predefined channel and data rate (by default using SF12 and in the channel providing 10% duty cycle access). A receive window is opened for long enough to detect a preamble. If a preamble is detected, the device continues to receive the entire frame. If not, the receive window is immediately closed. If a downlink frame is received in RX1, the device does not open RX2.

B.2.4.1 Confirmed Frames and LoRaWAN Downlink Capabilities

As mentioned previously, frames can be transmitted in confirmed or unconfirmed mode. Confirmed frames and unconfirmed frames are transmitted $NbTrans$ times, with frequency hopping, unless a downlink frame is received in response to one of the transmissions. $NbTrans$ is by default 1. For regular uplink frames, an exact delay between retransmissions is not mandated, but must be long enough to allow receive windows from the previous uplink transmission to expire, and to ensure adherence to the duty cycle regulations of the 868 MHz band. The delay of retransmissions of confirmed frames that were initially triggered by some external event must be randomised if that event is expected to be synchronised across multiple devices (a correlated, as opposed to independent, event). The constraint on the delay between transmissions is dependent on the length of time the device has been powered up, and detailed in Table B.7. The randomised delay is in particular designed to mitigate the effect of many devices attempting to join the network simultaneously.

Table B.7: LoRaWAN Correlated-Event-based Confirmed Frames Retransmissions Delay Policy

Time Since Power-up	**Randomised Delay**
$<$ 1 hour	$< 36s$
$>$ 1 hour and $<$ 11 hours	$< 36s$
otherwise: where N = (hours since power-up - 11) / 24	$< N * 8.7s$

A naive approach requiring all frames to be retransmitted with a high number of retransmissions allowed for each frame results in an avalanche effect of collisions, limiting the capacity of the network [232]. In general, LoRaWAN downlink traffic is limited as the gateways also have to adhere to the regulatory time-on-air limits. Improper use of confirmed frames and retransmissions severely impacts network performance as, when the gateway cannot transmit acknowledgements due to the duty cycle limits, frames that have been successfully received are not responded to. This causes further retransmissions of already-received frames by oblivious devices, reducing the goodput of the network and further increasing the probability of collisions [159]. The recommended number of transmissions of a single frame in the LoRaWAN specification has already been revised from 8 down to 1.

B.2.4.2 Channel Selection

Available channels for transmission are located in the EU 433 MHz and EU 868 MHz bands, and have to adhere to the regulations outlined in Section A.2. The allocation of channels in the EU 868 MHz band is at the discretion of the network operator, with the exception that the following three channels must be implemented:

Table B.8: LoRaWAN Mandated Channels in the EU 868 band

Channel Frequency	Bandwidth	Band number	Duty Cycle
868.10MHz	125kHz	48	1%
868.30MHz	125kHz	48	1%
868.50MHz	125kHz	48	1%

The choice of channel of transmission is taken on a pseudo-random basis, with the constraint that channel usage must not cause the regular breakage of the duty cycle limitations for the subband. Adaptive channel hopping approaches and dedicated channels for high priority traffic and downlink feedback have been identified as potential areas of study for LoRaWAN [126].

B.2.5 Class B

Class B of LoRaWAN is designed to enable server-controlled bi-directional communication while still maintaining a device lifetime suitable for IoT devices. Effectively, choice between Class A and Class B is a trade-off between energy efficiency and downlink delay; Class A is energy efficient but the delay is dependent on the rate of uplink frames. Class B requires the device to wake periodically and so is less energy efficient, but lowers the delay for downlink transmissions. Class B devices schedule receive windows, referred to as “ping slots”, in which to potentially receive downlink frames. This requires precise time synchronisation across the network, and so every gateway periodically broadcasts a timing beacon, and Class B devices synchronise their clocks to the included timestamp. Devices then schedule ping slots in the time between beacons. At each occurrence of a ping slot the device reads for an incoming frame, and the Network Server can potentially send a downlink frame. For uplink, the Class B device transmits using Class A-style channel access, with just one bit changed in

the frame header to indicate the device class. Class B also supports multicast downlink transmissions, though the setup of this scheme is not defined in the LoRaWAN protocol. The general format of Class B communications is shown in Figure B.13, with the introduced Class B features shown in yellow.

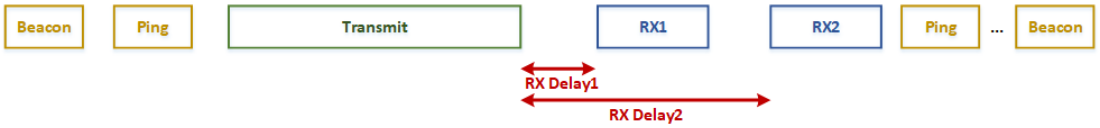


Figure B.13: LoRaWAN Class B Transmission

Beacons are transmitted every 128 s simultaneously from every LoRaWAN gateway. Class B devices use the timestamp contained within, along with the device address, to calculate the assigned ping slot for use in this round of transmission. The exact periodicity of beacons is defined as being every 128 s since the beginning of GPS time (i.e. 00:00:00 6th Jan 1980) plus a delay of $1.5ms \pm 1\mu s$. Gateway beacons are thus synchronised across all LoRaWAN networks. Every 128 s, a beacon is received by the device from its nearest gateway. The next 2.120 s are reserved for the frame Time-On-Arrival, and for potential network management frames. The following 122.880 s is the beacon window period, which is split into $2^{12} = 4096$ $30ms$ ping slots. The final $3s$ constitutes a guard period. The format of the beacon period is shown in Figure B.14.

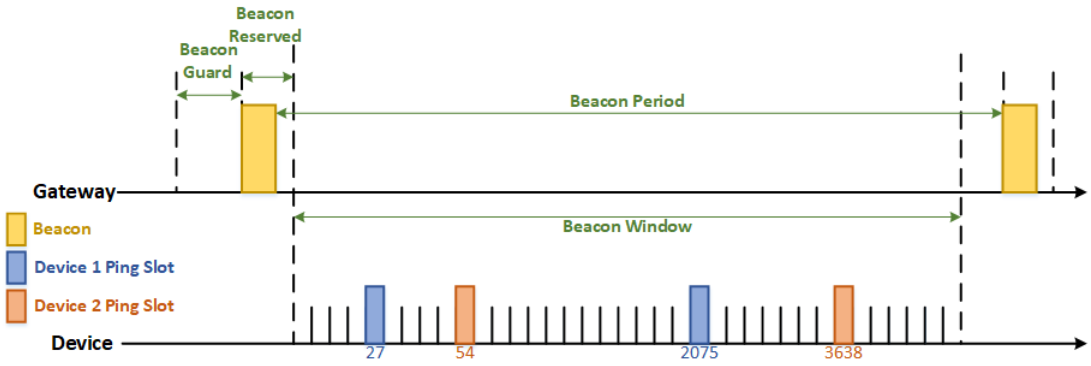


Figure B.14: Class B Beacon Period Format

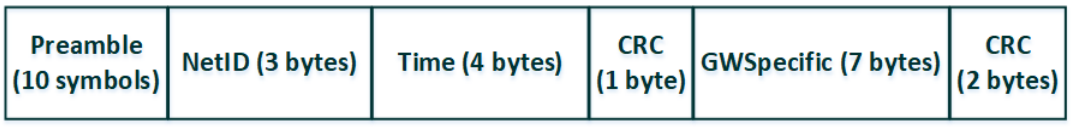


Figure B.15: Class B Beacon Frame Format

The beacon format in Europe is displayed in Figure B.15, and consists of a preamble of 10 unmodulated symbols, followed by a 17-byte payload without a LoRa physical header or CRC. The $Time$ field is the least significant 32 bits of GMT time. $GwSpecific$ is gateway-dependent, and may contain location data for the closest gateway. For stationary devices, the timestamp is enough to

continue to operate in Class B mode. In Europe, beacons are transmitted using DR3 and a coding rate of 4/5. In EU 868 MHz, the default broadcast frequency is 869.525 MHz, which is inside the subband with the highest spectrum access allocation (10%) and highest maximum effective radiated power (500 mW).

Next, the Network Server calculates the ping slots to be used by each device in the beacon period. The number of slots for each device is prearranged by the Network Server, and must be a power of two, up to a maximum of 128 (i.e. a ping slot every second):

$$period = 2^{12}/slots \tag{B.9}$$

The initial offset value O for a device in a beacon period is calculated using the first two bytes of the result of applying the AES-128 encryption algorithm to a padded block consisting of the beacon timestamp *(Time)* and the device address *(DevAddr)*, using a key of all zeroes:

$$R = AES128\_enc(16 * (0x00), Time|DevAddr|pad16) \tag{B.10}$$

$$O = (R[0] + R[1] * 256)\%period \tag{B.11}$$

As a device has no knowledge of the scheduled ping slots of other devices, the inclusion of the timestamp in the Equation B.10 prevents continuous collisions between devices across beacon periods. The timings of ping slots are then equally spaced in the broadcast period, starting from the offset:

$$timings = \{O + x * period \mid x < slots, x \in \mathbb{N}\} \tag{B.12}$$

If the preamble is detected during a ping slot, the packet is received and handled based on the MAC header. If no preamble is detected the device transitions directly into idle or sleep mode.

If a beacon is received at the start of the beacon period by a device, the timestamp is used to generate the exact timings of the ping slots to be used in this beacon period, in an identical fashion to the Network Server. If no beacon is received, the device uses an internal clock to generate the correct timestamp in order to retrieve the ping slot allocations, and gradually increases the length of time the receive windows are open for the expected Class B downlink frames. A device must be able to function in this "beacon-less" mode for up to two hours. After two hours in beacon-less mode, if a beacon frame has still not been received the device reverts back to Class A mode. The device may transition to Class B mode by periodically searching for network beacon frames. Such a search is referred to as "beacon tracking" in the specification [227].

B.2.6 Class C

LoRaWAN Class C is designed for devices without strict energy constraints, and functions similarly to Class A except that the RX2 window is extended to occur whenever possible. LoRaWAN Class C also supports multicast transmissions, though similarly to Class B, the setup procedure is not defined in the protocol specification. As most LoRaWAN applications have constraints on energy consumption, research into Class C has been limited. The format of Class C operation is shown in Figure B.16.

Figure B.16: LoRaWAN Class C Transmission

B.2.7 LoRaWAN Adaptive Data Rate

The performance of a LoRaWAN network is controlled by the Adaptive Data Rate (ADR) algorithm, which modifies the data rate of devices based on the current network conditions. Optimal choice of the data rate for transmissions ensures that devices operate in an energy efficient manner and increases the maximum number of devices that can be managed by a single gateway. The use of ADR is optional but highly recommended by the LoRaWAN protocol specification [227], and LoRaWAN networks do not realistically scale without use of ADR [94]. ADR is split into two independent algorithms, one of which runs on the device (hereby referred to as the End Device-side algorithm, or ED-side algorithm), and the other which runs on the Network Server (hereby referred to as the Network Server-side algorithm, or NS-side algorithm).

B.2.7.1 ADR - End Device-side

The End Device-side algorithm is explicitly defined in the LoRaWAN protocol specification [227]. The algorithm reacts to a lack of requested downlink feedback by decreasing the data rate (and thus increasing the range). Thus, on the End Device-side the data rate is only decreased (and therefore slowed). Pseudocode of this algorithm is provided in Algorithm B.1. If a device is not currently using the slowest available data rate, a device requests a downlink response after ADR_ACK_LIMIT uplink frames by setting the *ADRAckReq* bit in the LoRaWAN header. For each ADR_ACK_DELAY frames sent without a downlink response, the data rate of the device is decremented. Upon receipt of any downlink frame, the *ADRAckReq* bit is unset and all counters are cleared. By default, ADR_ACK_LIMIT and ADR_ACK_DELAY are both equal to 32.

Algorithm B.1 End Device-Side Adaptive Data Rate [227]

```
1: on downlink frame receive:
2: counter ← 0.
3: adr_ack_req_bit ← 0.
4: on uplink frame send:
5: if data_rate ≠ DATA_RATE_MIN then
6:     counter ← counter + 1.
7:     if adr_ack_req_bit = 0 and counter = ADR_ACK_LIMIT then
8:         adr_ack_req_bit ← 1.
9:     else if counter = ADR_ACK_LIMIT + ADR_ACK_DELAY then
10:        counter ← ADR_ACK_LIMIT
11:        if tx_power ≠ TX_POWER_MAX then
12:            tx_power ← TX_POWER_MAX
13:        else if data_rate ≠ DATA_RATE_MIN then
14:            data_rate ← data_rate − 1
15:        else
16:            reenableDefaultUplinkChannels()
```

B.2.7.2 ADR - Network Server-side

The second algorithm runs simultaneously on the Network Server. There is no official version of the algorithm defined in the LoRaWAN protocol, but Semtech do provide a recommended algorithm [233], which has been adopted by open-source projects such as the Things Network [234] and ChirpStack [235], and has been the assumed algorithm for previous research analysing ADR [95, 97, 99]. In this algorithm, the Network Server records the highest SINR value for each incoming packet, and then, if the device is not already using the fastest data rate and lowest transmission power, calculates the expected most suitable data rate for the device. The new data rate is calculated based on the previously received SINR values, relative to the current data rate (see Table B.9) and transmission power level (see Table B.10). An added $margin\_db$ parameter (usually set to $5dB$) prevents oscillation between data rates. The new data rate and transmission power for the device are communicated through the *LinkADRReq* MAC command in a downlink frame, and this is acknowledged with the piggybacking of a *LinkADRAns* MAC command in the device's next uplink frame. This algorithm is described in pseudocode in Algorithm B.2. Note that this algorithm only increases the data rate.

Table B.9: SNR_TABLE(DR)

Data Rate	Required SNR (dBm)
DR0	-20
DR1	-17.5
DR2	-15
DR3	-12.5
DR4	-10
DR5	-7.5

Table B.10: TX Power Table

tx_power	**Configuation (EIRP)**
0	Max EIRP
1	Max EIRP - 2dB
2	Max EIRP - 4dB
3	Max EIRP - 6dB
4	Max EIRP - 8dB
5	Max EIRP - 10dB
6	Max EIRP - 12dB
7	Max EIRP - 14dB
8..14	Reserved For Use
15	Defined in LoRaWAN

The frequency the NS-side algorithm calculates the most suitable data rate for each device is not specifically defined in the Semtech document, but two recommendations are provided: to run the algorithm either every time the *ADRAckReq* bit is set by a device, or to run the algorithm once every Nth uplink frame from a device. The second approach (with $N = 20$) was taken in the implementation of the Things Network and ChirpStack, and in [95, 97].

Algorithm B.2 Network Server-Side Adaptive Data Rate [233]

```
1:  on uplink frame receive (device, packet):
2:  if length(device.packets) = 20 then
3:      packets.pop()
4:  device.packets.push(packet.snrMax).
5:
6:  on downlink frame send (device, packet):
7:  if packet.frameCounter%20 = 0 then
8:      snr_max ← max(device.packets.snrMax)
9:      snr_margin ← int(snr_max − SNR_TABLE(data_rate) − margin_db)
10:     n_step ← snr_margin/3
11:     while n_step ≠ 0 do
12:         if n_step > 0 then
13:             if data_rate < 5 then
14:                 data_rate ← data_rate + 1.
15:             else
16:                 if tx_power ≠ TX_POWER_MIN then
17:                     tx_power ← tx_power + 1.
18:             n_step ← n_step − 1.
19:         else
20:             if tx_power ≠ TX_POWER_MAX then
21:                 tx_power ← tx_power − 1.
22:                 n_step ← n_step + 1.
```

B.2.8 Traffic Patterns

As mentioned in Chapter 1, LPWAN technologies such as LoRaWAN are particularly suited to contribute to coverage of mMTC use cases [157]. mMTC differs significantly from Human-Type

Communications (HTC). Overall there are a few key features which characterise mMTC:

- mMTC features a large number of heterogeneous devices simultaneously connected to the same gateway (with a targeted connection density of 1 million devices per km^2).
- mMTC devices are typically low-cost, energy-constrained, and ultra-low power [236]. mMTC devices are also generally less mobile than HTC devices [44].
- mMTC traffic is uplink-dominant, packets are generally shorter, and there are longer periods in-between transmissions [44, 237].
- mMTC traffic has different diurnal patterns to HTC, where the difference between the level of activity during peak and non-peak hours is not as extreme [238, 239].
- In addition, whereas HTC traffic is uncoordinated on small timescales, mMTC may be co-ordinated (many machines react on global events in a synchronised fashion).

Analysis of mMTC traffic has led to the development of mMTC traffic models. Note that the designation mMTC was first formally defined by 3GPP and ITU in 2018 as a subset of Machine-Type Communications (MTC) (i.e. massive MTC, as opposed to critical MTC which has stricter latency requirements) [157]. The majority of the research described below pre-dates the new designation and simply refers to MTC. However, the researchers generally do model MTC traffic that is much more similar to mMTC rather than critical MTC (no major focus on low latency) and as such can be considered related to mMTC. 3GPP have developed an MTC traffic model consisting of the two scenarios [158], which categorises traffic into two types: 1) periodic: deterministic traffic that is transmitted at regular intervals, and 2) event-driven: asynchronous traffic that is triggered by some external event, as can be seen in Figure B.17 [142]. Note that a single device may transmit both periodic and event-driven traffic (for example, a rainfall monitor that reports an alert when the reading exceeds a threshold value) [44]. In addition, event-driven traffic may be independent to a single device, or may be geographically correlated (for example, flood alerts) [146]. Spacial and temporal correlated events can result in bursty traffic which impairs network performance [143].

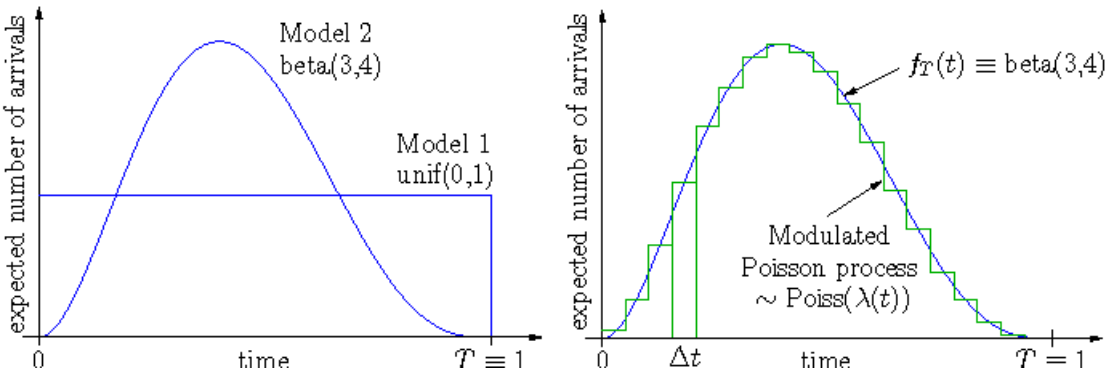


Figure B.17: 3GPP MTC Traffic Model [142]

B.3 Evaluating LoRaWAN Using Simulation

Though the price of an individual LoRaWAN device is low, these networks are expected to scale where an individual gateway will be handling the traffic of thousands of devices. Simulation tools enable the study and analysis of different potential LoRaWAN deployment options, where the otherwise real-world deployment of the network would not be realistically feasible because of the cost involved. LoRaWAN has been implemented in different simulators to enable the analysis of the protocol in detail at scale. Modules implementing the physical layer of LoRa and Class A of LoRaWAN have been developed in ns-3 [75, 73, 240, 241], SimPy [94] and OMNET++ [97]. In analysis, [97] models the ADR algorithm, [73] simulates dense environments, [75] focuses on scalability when downlink feedback is required, and [241] evaluates the impact of Carrier Sense in LoRaWAN.

Network Simulator 3 (ns-3) is a widely used simulator in wireless research. ns-3 is a discrete-event network simulator which enables the simulation of heterogeneous networks consisting of thousands of nodes. Support is currently available for a number of wireless and wired protocols, including LTE, Wi-Fi, and IEEE 802.15.4. While the LoRaWAN MAC layer can be closely implemented in simulation, there are different methods to modelling the PHY layer which each approach an approximation to real-world modulation, propagation, and reception. In [75], the LoRa PHY layer error model is based on measuring the bit error rate for different configurations over an additive white Gaussian noise (AWGN) channel, based on a LoRa PHY baseband implementation in Matlab. Packets received below a cutoff SINR value (which includes thermal noise), based on the data and coding rate, are immediately discarded. A chunk-based approach to packet reception is taken, similar to the modelling of IEEE 802.11 and IEEE 802.15.4 in ns-3. In this approach, every time the SINR changes during the reception of a packet, the reception of the bits since the last SINR change is evaluated, based on the error model and SINR. The ns-3 default propagation loss model, *LogDistancePropagationLoss* is used, though this can be configured in the simulator. If the reception of any chunk fails, the packet is failed to be received. This approach models propagation loss as well as enabling modelling of interference from LoRa transmissions and transmissions from other devices operating in the same band. The simulator also ensures that devices and gateways adhere to the duty cycle regulations of the EU 868 MHz band.

A similar error-model and chunk-based approach is taken in [240], except the instantaneous SINR of colliding packets is taken as a parameter to the bit error calculations instead of the average SINR, enabling the frames to be destroyed by short high-powered interfering messages. In addition, the *Hata* propagation loss model, which is suitable for urban areas, is used [74]. The module presented in [241] is less well documented but appears to take the same approach, with the error model directly based on an open-source implementation of the LoRa protocol from [208]. The propagation loss model used in [73] is a 3GPP-defined model for macro cell propagation model in urban areas, with an additional model for loss due to the penetration of buildings. Additionally, the impact of co-spreading factor interference is specifically handled in this module. The module in [73] was extended in [84] to implement Class B mode. Table B.11 shows a direct comparison of each of the LoRaWAN simulators.

Table B.11: Comparison of LoRaWAN Simulators

Simulator	**LoRaSim**	**ns-3**				**FLoRa**
Module	**[94]**	**[75]**	**[240]**	**[73]**	**[241]**	**[97]**
Capture Effect	Yes	Yes*	Yes*	No	Yes*	Yes
SF Imperfect Orthogonality	No	Yes*	Yes*	Yes	Yes*	No
Downlink Frames	Yes	Yes	Yes	Yes	Yes	Yes
Duty Cycle Limits	Yes	Yes	Yes	Yes	Yes	Yes
Class A	Yes	Yes	Yes	Yes	Yes	Yes
Class B	No	No	No	Yes	No	No
Class C	No	No	No	No	No	No
ADR	Yes	No	No	No	No	Yes
MAC Commands	No	No	Yes	No	No	No
Join Procedure	No	No	No	No	No	No
Energy Consumption	Yes	No	Yes	No	Yes	Yes

*through use of the ns-3 spectrum module, not directly modelled

An alternative LoRaWAN simulator is LoRaSim, which is based on SimPy and was introduced in [94]. LoRaSim is a discrete-event simulator with support for LoRa communications using each of the defined data rates. The log-distance path loss model is used with individual parameters calculated through empirical measurements of a real LoRa system. The capture effect is supported, though the spreading factors are presumed to be fully orthogonal. Duty cycle restrictions and downlink frames were added to the simulator in [159], and ADR was added in [95]. The energy consumption framework of the simulator was extended in [242].

A final noteworthy LoRaWAN simulator is FLoRa, which was introduced in [97]. FLoRa is based on OMNET++, and has full support for LoRaWAN Class A communications and the modelling of the energy consumption of LoRaWAN transmissions. The module was developed to enable the analysis of the LoRaWAN ADR.

Overall, simulation tools for LoRaWAN are vital in the evaluation of LoRaWAN networks because of the expected scale of these networks. LoRaWAN simulators are in active development, and the work in this thesis contributes to the open development of LoRaWAN features. In particular, this is achieved by extending the ns-3 module introduced in [75], though the work has been designed in a modular fashion to enable the integration of developed features into any LoRaWAN ns-3 module.

B.3.1 Development Process of ns-3

Wireless stacks in ns-3 are structured to follow the OSI model; in the simulation setup a virtual node is defined, and then individual layers are applied to build up the network stack. As such, the development process for adding additional components to ns-3 follows a modular approach. Additionally, even for closely defined stacks (e.g. the PHY+MAC of IEEE 802.11n) the separate components are clearly defined, with all interactions between the layers occurring through callback functions. A NET layer is defined on top of the MAC layer, which is used to configure the MAC and PHY layer options (as an example, for a LoRaWAN gateway in [75], since the gateway can receive

on multiple channels and multiple spreading factors, the handling of each channel is modelled as an individual MAC, and the each spreading factor for each channel is modelled as an individual PHY). All development of core components is performed using C++. ns-3 uses the *waf* build automation tool for the compilation of the software and running of individual simulations. ns-3 also provides the generation of random seeds, with each random seed also providing a sequence of run numbers. Using the same random seed, each run is statistically independent. Thus, as long as the seed and run numbers are maintained, ns-3 enables reproducible yet statistically independent runs of an individual simulation.

B.4 Conclusions

The LoRaWAN protocol has been developed to provide connectivity for energy-constrained devices, enabling pervasive, low throughput applications at a low cost. This appendix has provided an overview of LoRaWAN. This review has also highlighted the limitations of the evaluation tools for the protocol, as well as the protocol itself. The configurable parameters of a LoRa transceiver which affect the performance of a link are the spreading factor, the coding rate, the selected physical channel, the preamble length, the packet size and the transmission periodicity. The performance of ALOHA-based access of LoRaWAN Class A does not scale gracefully, but does outperform classic models of ALOHA because of the strong capture effect property of LoRa.

Less research has focused on LoRaWAN Class B, which enables the development of reactive and evolving LoRaWAN applications, as well as having use in firmware updates and for multicast communications. A full model of the LoRaWAN protocol in simulation is not yet available. For simulations in ns-3, this includes key features of the protocol which can markedly affect overall network performance at scale, including the ADR and Class B. A more complete model of the LoRaWAN protocol in simulation would enable the evaluation of realistic large-scale LoRaWAN networks. Each of these factors has motivated the work in this thesis, which enables a more realistic evaluation of LoRaWAN networks (through implementations of energy consumption models, Class B and ADR in simulation) and develops approaches beyond regular ALOHA that enable more scalable and reliable LoRaWAN networks.

APPENDIX C

The Relationship between the Energy-related Units

A LoRaWAN device which is configured to transmit at a particular data rate and a particular transmission power will expend an amount of energy to perform the task. The total energy consumed is dependent on the length of time for the transmission (the "time-on-air", which is dependent on the data rate and length of the payload, see Subsection 2.2.1.1 for details), the current consumption for the transmit state, and the voltage of the device:

$$E_{tx} = t_{tx} * I_{tx} * Vcc \tag{C.1}$$

The energy consumption of the other states of a LoRaWAN transaction can be calculated in a similar manner e.g.

$$E_{rx} = t_{rx} * I_{rx} * Vcc \tag{C.2}$$

where the time taken for the reception is the length of time to receive the entire frame if a frame is detected, and the length of time to receive just the preamble if no frame is detected.

To calculate the energy consumption of an entire LoRaWAN transaction, the energy consumption in each of the states can simply be summed e.g.

$$E_{transaction} = E_{tx} + E_{ww1} + E_{rx1} + E_{ww2} + E_{ww2} \tag{C.3}$$

for the case where no downlink frame is received by the device.

Finally, the sleep state can be modelled in a similar method to the other states:

$$E_{sleep} = t_{sleep} * I_{sleep} * Vcc \tag{C.4}$$

where the value for the time in sleep mode can be calculated as the time *not* in the other states; for a LoRaWAN device sending periodic traffic the calculation of this can be perform based on the length of time in a single period.

The current consumption for each of the states of the transceiver remain within a small margin of error of a typical value, and can typically be modelled as constants. The values for these constants can be derived from the device datasheet, or through direct measurements. The value for the voltage can also be derived from the datasheet, and will vary between a maximum and minimum operating voltage, and decreases as the battery is drained. Based on the periodicity of traffic, the data rate, and the transmission power used, the average overall current consumption across time for the device can be calculated, and used to calculate the length of time before the voltage of the device drops below the minimum operating voltage, which is dependent on the battery capacity.

References

[1] A. Whitmore, A. Agarwal, and L. Da Xu. The Internet of Things: A survey of topics and trends. *Information Systems Frontiers, Springer*, 17(2):261–274, Apr 2015.

[2] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, Oct 2010.

[3] F. Wortmann and K. Flüchter. Internet of Things – Technology and Value Added. *Business & Information Systems Engineering*, 57(3):221–224, Jun 2015.

[4] B. S. Chaudhari, M. Zennaro, and S. Borkar. LPWAN Technologies: Emerging Application Characteristics, Requirements, and Design Considerations. *Future Internet*, 12(3), Mar 2020.

[5] A. Gupta, T. Tsai, D. Rueb, M. Yamaji, and P. Middleton. Forecast: Internet of Things - Endpoints and Associated Services Worldwide, Dec 2017. Gartner. [Online]. Available: https://www.gartner.com/en/documents/3840665/forecast-internet-of-things-endpoints-and-associated-ser.

[6] Statista Research Department. Internet of Things - number of connected devices worldwide 2015-2025, Nov 2016. Statista. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[7] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon. The Internet of Things: Mapping the Value Beyond The Hype, Jun 2015. McKinsey.

[8] S. Li, L. Da Xu, and S. Zhao. 5G Internet of Things: A survey. *Journal of Industrial Information Integration*, 10:1–9, Jun 2018.

[9] ITU. Rec. ITU-R M.2083-0: IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond, Sep 2015.

[10] H. G. S. Filho, J. P. Filho, and V. L. Moreli. The adequacy of LoRaWAN on smart grids: A comparison with RF mesh technology. In *IEEE International Smart Cities Conference (ISC2)*, pages 1–6, Sep 2016.

[11] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas. Low Throughput Networks for the IoT: Lessons Learned from Industrial Implementations. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 181–186, Dec 2015.

[12] ETSI. European Standard 300 220-1: Short Range Devices (SRD) operating in the frequency range 25 MHz to 1000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment , Jun 2018.

[13] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni. Computer Network Simulation with ns-3: A Systematic Literature Review. *Electronics*, 9(2), Feb 2020.

[14] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades. The Power of Models: Modeling Power Consumption for IoT Devices. *IEEE Sensors Journal*, 15(10):5777–5789, Oct 2015.

[15] B. Martinez, M. Montón, I. Vilajosana, and X. Vilajosana. Early Scavenger Dimensioning in Wireless Industrial Monitoring Applications. *IEEE Internet of Things Journal*, 3(2):170–178, Apr 2016.

[16] L. Casals Ibanez, B. Mir Masnou, R. Vidal Ferre, and C. Gomez. Modeling the energy performance of LoRaWAN. *Sensors*, 17:2364, Oct 2017.

[17] T. Bouguera, J. Diouris, J. Chaillout, and G. Andrieux. Energy consumption modeling for communicating sensors using LoRa technology. In *IEEE Conference on Antenna Measurements & Applications (CAMA)*, pages 1–4, Sep 2018.

[18] T. Bouguera, J. Diouris, J. Chaillout, R. Jaouadi, and G. Andrieux. Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. *Sensors*, 18(7), Jun 2018.

[19] J. Liando, A. Gamage, A. Tengourtius, and M. Li. Known and Unknown Facts of LoRa: Experiences from a Large-Scale Measurement Study. *ACM Transactions on Sensor Networks*, 15(2), Feb 2019.

[20] M. Nurgaliyev, A. Saymbetov, Y. Yashchyshyn, N. Kuttybay, and D. Tukymbekov. Prediction of energy consumption for LoRa based wireless sensors network. *Wireless Networks*, 26:3507–3520, Feb 2020.

[21] P. S. Cheong, J. Bergs, C. Hawinkel, and J. Famaey. Comparison of LoRaWAN classes and their power consumption. In *IEEE Symposium on Communications and Vehicular Technology (SCVT)*, pages 1–6, Nov 2017.

[22] 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*, PP:1–11, Apr 2020.

[23] A. Ikpehai, B. Adebisi, and K. Anoh. Effects of Traffic Characteristics on Energy Consumption of IoT End Devices in Smart City. In *Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6, Oct 2018.

[24] R. Singh, P. Puluckul, R. Berkvens, and M. Weyn. Energy Consumption Analysis of LPWAN Technologies and Lifetime Estimation for IoT Application. *Sensors*, 20(17):4794, Aug 2020.

[25] G. Hassan, M. ElMaradny, M. A. Ibrahim, A. M. Rashwan, and H. S. Hassanein. Energy Efficiency Analysis of Centralized-Synchronous LoRa-based MAC Protocols. In *International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 999–1004, Jun 2018.

[26] A. Kurtoglu, J. Carletta, and K. Lee. Energy consumption in long-range linear wireless sensor networks using LoRaWan and ZigBee. In *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1163–1167, Aug 2017.

[27] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal. Towards Energy-Fairness in LoRa Networks. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 788–798, Jul 2019.

[28] 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. In *Hindawi Wireless Communications and Mobile Computing*, volume 2019, Jul 2019.

[29] D. Purkovic, M. Honsch, and T. R. M. K. Meyer. An Energy Efficient Communication Protocol for Low Power, Energy Harvesting Sensor Modules. *IEEE Sensors Journal*, 19(2):701–714, Jan 2019.

[30] F. Orfei, C. Benedetta Mezzetti, and F. Cottone. Vibrations powered LoRa sensor: An electromechanical energy harvester working on a real bridge. In *IEEE SENSORS*, pages 1–3, Oct 2016.

[31] W. Lee, M. J. W. Schubert, B. Ooi, and S. J. Ho. Multi-Source Energy Harvesting and Storage for Floating Wireless Sensor Network Nodes With Long Range Communication Capability. *IEEE Transactions on Industry Applications*, 54(3):2606–2615, May 2018.

[32] G. Loubet, A. Takacs, and D. Dragomirescu. Towards the Design of Wireless Communicating Reinforced Concrete. *IEEE Access*, 6:75002–75014, Nov 2018.

[33] S. Tjukovs, J. Eidaks, and D. Pikulins. Experimental Verification of Wireless Power Transfer Ability to Sustain the Operation of LoRaWAN Based Wireless Sensor Node. In *Advances in Wireless and Optical Communications (RTUWO)*, pages 83–88, Nov 2018.

[34] C. Delgado, J. M. Sanz, and J. Famaey. On the Feasibility of Battery-Less LoRaWAN Communications Using Energy Harvesting. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2019.

[35] Semtech. SX1272/3/6/7/8: LoRa Modem Designer's Guide, Jul 2013.

[36] C. Gomez, J. Veras, R. Vidal, L. Casals, and J. Paradells. A Sigfox Energy Consumption Model. *Sensors*, 19(3):681, Feb 2019.

[37] 3GPP. TR 45.820: Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT), 2015.

[38] 3GPP. R1-157647: Summary of evaluation results, 2015.

[39] 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*, 6(3):5691–5701, Jun 2019.

[40] B. Martinez, F. Adelantado, A. Bartoli, and X. Vilajosana. Exploring the Performance Boundaries of NB-IoT. *IEEE Internet of Things Journal*, 6(3):5702–5712, Jun 2019.

[41] A. K. Sultania, P. Zand, C. Blondia, and J. Famaey. Energy Modeling and Evaluation of NB-IoT with PSM and eDRX. In *IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, Dec 2018.

[42] M. ElSoussi, P. Zand, F. Pasveer, and G. Dolmans. Evaluating the Performance of eMTC and NB-IoT for Smart City Applications. In *IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.

[43] T. Elshabrawy and J. Robert. Enhancing LoRa Capacity using Non-Binary Single Parity Check Codes. In *International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–7, Oct 2018.

[44] T. Hoßfeld, F. Metzger, and P. E. Heegaard. Traffic modeling for aggregated periodic IoT data. In *Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 1–8, Feb 2018.

[45] M. O. Farooq and D. Pesch. Analyzing LoRa: A use case perspective. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 355–360, 2018.

[46] M. Magno, F. A. Aoudia, M. Gautier, O. Berder, and L. Benini. WULoRa: An energy efficient IoT end-node for energy harvesting and heterogeneous communication. In *Design, Automation Test in Europe Conference (DATE)*, pages 1528–1533, Mar 2017.

[47] S. Kim, M. Lee, and C. Shin. IoT-Based Strawberry Disease Prediction System for Smart Farming. *Sensors*, 18(11):4051, Nov 2018.

[48] G. Peruzzi and A. Pozzebon. A Review of Energy Harvesting Techniques for Low Power Wide Area Networks (LPWANs). *Energies*, 13(13):3433, Jun 2020.

[49] Sensirion. Datasheet SHT3x-DIS Humidity and Temperature Sensor, May 2018.

[50] A. S. Andrenko, X. Lin, and M. Zeng. Outdoor RF spectral survey: A roadmap for ambient RF energy harvesting. In *IEEE Region 10 Conference (TENCON)*, pages 1–4, Nov 2015.

[51] T. Beng Lim, N. M. Lee, and B. K. Poh. Feasibility study on ambient RF energy harvesting for wireless sensor network. In *IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO)*, pages 1–3, Dec 2013.

[52] L. Guenda, E. Santana, A. Collado, K. Niotaki, N. Carvalho, and A. Georgiadis. Electromagnetic energy harvesting – global information database. *Transactions on Emerging Telecommunications Technologies*, 25:56–63, Jan 2014.

[53] A. Alex-Amor, A. Palomares-Caballero, J.M. Fernandez, P Padilla, D. Marcos, M. Sierra-Castaner, and J. Esteban. RF Energy Harvesting System Based on an Archimedean Spiral Antenna for Low-Power Sensor Applications. *Sensors*, 19(6):1318, Mar 2019.

[54] M. Mrnka, P. Vasina, M. Kufa, V. Hebelka, and Z. Raida. The RF Energy Harvesting Antennas Operating in Commercially Deployed Frequency Bands: A Comparative Study. *International Journal of Antennas and Propagation*, 2016:1–11, Apr 2016.

[55] R. J. Vyas, B. B. Cook, Y. Kawahara, and M. M. Tentzeris. E-WEHP: A Batteryless Embedded Sensor-Platform Wirelessly Powered From Ambient Digital-TV Signals. *IEEE Transactions on Microwave Theory and Techniques*, 61(6):2491–2505, Jun 2013.

[56] R. Shigeta, T. Sasaki, D. M. Quan, Y. Kawahara, R. J. Vyas, M. M. Tentzeris, and T. Asami. Ambient RF Energy Harvesting Sensor Device With Capacitor-Leakage-Aware Duty Cycle Control. *IEEE Sensors Journal*, 13(8):2973–2983, Aug 2013.

[57] Y. Luo, L. Pu, G. Wang, and Y. Zhao. RF Energy Harvesting Wireless Communications: RF Environment, Device Hardware and Practical Issues. *Sensors*, 19(13):3010, Jul 2019.

[58] Y. Chapre, P. Mohapatra, S. Jha, and A. Seneviratne. Received signal strength indicator and its analysis in a typical WLAN system. In *IEEE Conference on Local Computer Networks (LCN)*, pages 304–307, Oct 2013.

[59] M. Prauzek, J. Konecny, M. Borova, K. Janosova, J. Hlavica, and P. Musilek. Energy Harvesting Sources, Storage Devices and System Topologies for Environmental Wireless Sensor Networks: A Review. *Sensors*, 18:2446, Jul 2018.

[60] M. A. Andersson, A. Özçelikkale, M. Johansson, U. Engström, A. Vorobiev, and J. Stake. Feasibility of Ambient RF Energy Harvesting for Self-Sustainable M2M Communications Using Transparent and Flexible Graphene Antennas. *IEEE Access*, 4:5850–5857, Aug 2016.

[61] M. Piñuela, P. D. Mitcheson, and S. Lucyszyn. Ambient RF Energy Harvesting in Urban and Semi-Urban Environments. *IEEE Transactions on Microwave Theory and Techniques*, 61(7):2715–2726, Jul 2013.

[62] A. Collado, S. Daskalakis, K. Niotaki, R. Martinez, F. Bolos, and A. Georgiadis. Rectifier Design Challenges for RF Wireless Power Transfer and Energy Harvesting Systems. *Radioengineering*, 26(2):411–417, Jun 2017.

[63] A. Boaventura, A. Collado, N. B. Carvalho, and A. Georgiadis. Optimum behavior: Wireless power transmission system design through behavioral models and efficient synthesis techniques. *IEEE Microwave Magazine*, 14(2):26–35, Mar 2013.

[64] K. Niotaki, S. Kim, S. Jeong, A. Collado, A. Georgiadis, and M. M. Tentzeris. A Compact Dual-Band Rectenna Using Slot-Loaded Dual Band Folded Dipole Antenna. *IEEE Antennas and Wireless Propagation Letters*, 12:1634–1637, Dec 2013.

[65] K. Niotaki, A. Georgiadis, A. Collado, and J. S. Vardakas. Dual-Band Resistance Compression Networks for Improved Rectifier Performance. *IEEE Transactions on Microwave Theory and Techniques*, 62(12):3512–3521, Dec 2014.

[66] G. Saini, S. Sarkar, M. Arrawatia, and M. S. Baghini. Efficient power management circuit for RF energy harvesting with 74.27% efficiency at 623nW available power. In *IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 1–4, Jun 2016.

[67] X. Chen, L. Huang, J. Xing, Z. Shi, and Z. Xie. Energy harvesting system and circuits for ambient WiFi energy harvesting. In *International Conference on Computer Science and Education (ICCSE)*, pages 769–772, Aug 2017.

[68] Z. Zeng, J. J. Estrada-López, M. A. Abouzied, and E. Sánchez-Sinencio. A Reconfigurable Rectifier With Optimal Loading Point Determination for RF Energy Harvesting From -22 dBm to -2 dBm. *IEEE Transactions on Circuits and Systems*, 67(1):87–91, Jan 2020.

[69] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan. Reincarnation in the Ambiance: Devices and Networks with Energy Harvesting. *IEEE Communications Surveys & Tutorials*, 16(1):195–213, Feb 2014.

[70] G. V. Merrett and A. S. Weddell. Supercapacitor leakage in energy-harvesting sensor nodes: Fact or fiction? In *International Conference on Networked Sensing (INSS)*, pages 1–5, Jun 2012.

[71] A. Pegatoquet, T. N. Le, and M. Magno. A Wake-Up Radio-Based MAC Protocol for Autonomous Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 27(1):56–70, Feb 2019.

[72] B. Munir and V. Dyo. On the Impact of Mobility on Battery-Less RF Energy Harvesting System Performance. *Sensors*, 18:3597, Oct 2018.

[73] D. Magrin, M. Centenaro, and L. Vangelista. Performance evaluation of LoRa networks in a smart city scenario. In *IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017.

[74] B. Reynders and S. Pollin. Chirp spread spectrum as a modulation technique for long range communication. In *Symposium on Communications and Vehicular Technologies (SCVT)*, pages 1–5, Nov 2016.

[75] 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*, 4(6):2186–2198, Dec 2017.

[76] Semtech. SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver Rev. 3.1, Mar 2017.

[77] H. Wu, S. Nabar, and R. Poovendran. An Energy Framework for the Network Simulator 3 (ns-3). In *International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, pages 222–230, Mar 2011.

[78] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi. Energy Consumption Anatomy of 802.11 Devices and Its Implication on Modeling and Design. In *International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 169–180, Jan 2012.

[79] S. T. V. Pasca, B. Akilesh, A. V. Anand, and B. R. Tamma. A NS-3 module for LTE UE energy consumption. In *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, Nov 2016.

[80] V. Rege and T. Pecorella. A Realistic MAC and Energy Model for 802.15.4. In *Workshop on NS-3 (WNS3)*, pages 79–84, Jun 2016.

[81] F. Delobel, N. El Rachkidy, and A. Guitton. Analysis of the delay of confirmed downlink frames in Class B of LoRaWAN. In *IEEE Vehicular Technology Conference (VTC)*, pages 1–6, Jun 2017.

[82] D. Ron, C. Lee, K. Lee, H. Choi, and J. Lee. Performance Analysis and Optimization of Downlink Transmission in LoRaWAN Class B Mode. *IEEE Internet of Things Journal*, 7(8):7836–7847, Aug 2020.

[83] F. Hessel, L. Almon, and F. Alvarez. ChirpOTLE: a framework for practical LoRaWAN security evaluation. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 306–316, Jul 2020.

[84] Y. Shiferaw, A. Arora, and F. Kuipers. LoRaWAN Class B Multicast Scalability. In *IFIP Networking Conference*, pages 609–613, Jun 2020.

[85] M. Pasetti, S. Rinaldi, E. Sisinni, P. Ferrari, and A. Flammini. On the Use of Class B LoRaWAN for the Coordination of Smart Interface Protection Systems. In *IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, pages 1–6, Sep 2019.

[86] M. Pasetti, E. Sisinni, P. Ferrari, S. Rinaldi, A. Depari, P. Bellagente, D. Della Giustina, and A. Flammini. Evaluation of the Use of Class B LoRaWAN for the Coordination of Distributed Interface Protection Systems in Smart Grids. *Journal of Sensor and Actuator Networks*, 9(1):13, Feb 2020.

[87] C. E. Fehri, M. Kassab, N. Baccour, S. Abdellatif, P. Berthou, and I. Kammoun. An Uplink Synchronization scheme for LoRaWAN Class B. In *International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 47–52, Oct 2019.

[88] L. Chasserat, N. Accettura, and P. Berthou. Short: Achieving energy efficiency in dense LoRaWANs through TDMA. In *IEEE International Symposium On a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–5, Sep 2020.

[89] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, Jun 2018.

[90] J. Lee, W. Jeong, and B. Choi. A Scheduling Algorithm for Improving Scalability of LoRaWAN. *International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1383–1388, Oct 2018.

[91] F. A. Aoudia, M. Magno, M. Gautier, O. Berder, and L. Benini. A Low Latency and Energy Efficient Communication Architecture for Heterogeneous Long-Short Range Communication. In *Euromicro Conference on Digital System Design (DSD)*, pages 200–206, Aug 2016.

[92] K. Abdelfadeel, T. Farrell, D. McDonald, and D. Pesch. How to make Firmware Updates over LoRaWAN Possible. In *IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–10, Sep 2020.

[93] Y. Oh, J. Lee, and C. K. Kim. TRILO: A traffic indication-based downlink communication protocol for LoRaWAN. *Wireless Communications and Mobile Computing*, 2018:1–14, Sep 2018.

[94] M. Bor, U. Roedig, T. Voigt, and J. Alonso. Do LoRa Low-Power Wide-Area Networks Scale? In *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 59–67, Nov 2016.

[95] S. Li, U. Raza, and A. Khan. How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2018.

[96] V. Hauser and T. Hégr. Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure. In *IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 85–90, Aug 2017.

[97] M. Slabicki, G. Premsankar, and M. Di Francesco. Adaptive Configuration of LoRa Networks for Dense IoT Deployments. In *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 1–9, Apr 2018.

[98] K. Kousias, G. Caso, Ö. Alay, and F. Lemic. Empirical Analysis of LoRaWAN Adaptive Data Rate for Mobile Internet of Things Applications. In *Wireless of the Students, by the Students, and for the Students Workshop*, pages 9–11, Oct 2019.

[99] K. Abdelfadeel, V. Cionca, and D. Pesch. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. In *IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 15–24, Feb 2018.

[100] J. Lim and Y. Han. Spreading Factor Allocation for Massive Connectivity in LoRa Systems. *IEEE Communications Letters*, 22(4):800–803, Apr 2018.

[101] Q. Zhou, J. Xing, L. Hou, R. Xu, and K. Zheng. A Novel Rate and Channel Control Scheme Based on Data Extraction Rate for LoRa Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, Apr 2019.

[102] 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)*, pages 1–6, May 2017.

[103] G. Premsankar, B. Ghaddar, M. Slabicki, and M. D. Francesco. Optimal Configuration of LoRa Networks in Smart Cities. *IEEE Transactions on Industrial Informatics*, 16(12):7243–7254, Dec 2020.

[104] A. Loubany, S. Lahoud, and R. El Chall. Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways. *Computer Networks*, 182:107491, Aug 2020.

[105] 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 *Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–8, Jun 2018.

[106] A. Hoeller Jr, R. Demo Souza, O. Alcaraz López, H. Alves, M. de Noronha Neto, and G. Brante. Exploiting Time Diversity of LoRa Networks Through Optimum Message Replication. In *International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5, Aug 2018.

[107] L. Amichi, M. Kaneko, N. El Rachkidy, and A. Guitton. Spreading Factor Allocation Strategy for LoRa Networks Under Imperfect Orthogonality. In *IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019.

[108] D. Zorbas, G. Z. Papadopoulos, P. Maillé, N. Montavont, and C. Douligeris. Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations. In *International Conference on Telecommunications (ICT)*, pages 516–520, Jun 2018.

[109] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda. Evaluating LoRa energy efficiency for adaptive networks: From star to mesh topologies. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, Oct 2017.

[110] S. Dawaliby, A. Bradai, and Y. Pousset. Joint slice-based spreading factor and transmission power optimization in LoRa smart city networks. *Internet of Things*, pages 100–121, Sep 2019.

[111] A. Hoeller, R. D. Souza, S. Montejo-Sánchez, and H. Alves. Performance Analysis of Single-Cell Adaptive Data Rate-Enabled LoRaWAN. *IEEE Wireless Communications Letters*, 9(6):911–914, Jun 2020.

[112] A. Farhad, D. Kim, and J. Y. Pyun. Resource Allocation to Massive Internet of Things in LoRaWANs. *Sensors*, 20(9):2645, May 2020.

[113] L. Leonardi, F. Battaglia, and L. Lo Bello. RT-LoRa: A Medium Access Strategy to Support Real-Time Flows Over LoRa-Based Networks for Industrial IoT Applications. *IEEE Internet of Things Journal*, 6(6):10812–10823, Dec 2019.

[114] S. K. Sharma and X. Wang. Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions. *IEEE Communications Surveys & Tutorials*, 22(1):426 – 471, May 2019.

[115] D. Heeger, M. Garigan, and J. Plusquellic. Adaptive Data Rate Techniques for Energy Constrained Ad Hoc LoRa Networks. In *Global Internet of Things Summit (GIoTS)*, pages 1–6, Jun 2020.

[116] J. Park, K. Park, H. Bae, and C. Kim. EARN: Enhanced ADR with Coding Rate Adaptation in LoRaWAN. *IEEE Internet of Things Journal*, PP:1–11, Jun 2020.

[117] M. El-Aasser, T. Elshabrawy, and M. Ashour. Joint Spreading Factor and Coding Rate Assignment in LoRaWAN Networks. In *IEEE Global Conference on Internet of Things (GCIoT)*, pages 1–7, Dec 2018.

[118] A. Peruzzo and L. Vangelista. A Power Efficient Adaptive Data Rate Algorithm for LoRaWAN networks. In *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 90–94, Nov 2018.

[119] D. Zorbas, P. Maillé, B. O'Flynn, and C. Douligeris. Fast and Reliable LoRa-based Data Transmissions. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, Jun 2019.

[120] E. Sisinni, P. Bellagente, A. Depari, P. Ferrari, A. Flammini, S. Marella, M. Pasetti, S. Rinaldi, and A. Cagiano. A new LoRaWAN adaptive strategy for smart metering applications. In *IEEE International Workshop on Metrology for Industry 4.0 & IoT*, pages 690–695, Jun 2020.

[121] N. Benkahla, H. Tounsi, Y. Song, and M. Frikha. Enhanced ADR for LoRaWAN networks with mobility. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 1–6, Jun 2019.

[122] M. Capuzzo, D. Magrin, and A. Zanella. Confirmed traffic in LoRaWAN: Pitfalls and countermeasures. In *Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–7, Jun 2018.

[123] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo. On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In *International Conference on ITS Telecommunications (ITST)*, pages 55–59, Dec 2015.

[124] O. Georgiou and U. Raza. Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wireless Communications Letters*, 6(2):162–165, Apr 2017.

[125] L. Beltramelli, A. Mahmood, P. Osterberg, and M. Gidlund. LoRa beyond ALOHA: An Investigation of Alternative Random Access Protocols. *IEEE Transactions on Industrial Informatics*, PP:1–11, Feb 2020.

[126] F. Adelantado, X. Vilajosana, P. Tuset, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, pages 34–40, Jun 2017.

[127] E. De Poorter, J. Hoebeke, M. Strobbe, I. Moerman, S. Latre, M. Weyn, B. Lannoo, and J. Famaey. Sub-GHz LPWAN Network Coexistence, Management and Virtualization: An Overview and Open Research Challenges. *Wireless Personal Communications*, 95(1):187–213, Jul 2017.

[128] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch. TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things. *Computer Communications*, 153:1–10, Jan 2020.

[129] R. Trüb and L. Thiele. Increasing Throughput and Efficiency of LoRaWAN Class A. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, pages 54–64, Nov 2018.

[130] R. Piyare, A. L. Murphy, M. Magno, and L. Benini. On-Demand TDMA for Energy Efficient Data Collection with LoRa and Wake-up Receiver. In *International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–4, Oct 2018.

[131] R. Piyare, A. Murphy, M. Magno, and L. Benini. On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT. *Sensors, MDPI*, 18(11):3718, Nov 2018.

[132] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat. Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure. *IEEE Access*, 7:57663–57677, Apr 2019.

[133] S. Gao, X. Zhang, C. Du, and Q. Ji. A Multichannel Low-Power Wide-Area Network With High-Accuracy Synchronization Ability for Machine Vibration Monitoring. *IEEE Internet of Things Journal*, 6(3):5040–5047, Jun 2019.

[134] L. Leonardi, F. Battaglia, G. Patti, and L. Lo Bello. Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications. In *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 4141–4146, Oct 2018.

[135] J. Haxhibeqiri, I. Moerman, and J. Hoebeke. Low Overhead Scheduling of LoRa Transmissions for Improved Scalability. *IEEE Internet of Things Journal*, 6(2):3097–3109, Apr 2019.

[136] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch. $FREE$ - Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection in LoRaWAN. *IEEE Internet of Things Journal*, 7(1):669–683, Jan 2020.

[137] F. Bonafini, A. Depari, P. Ferrari, A. Flammini, M. Pasetti, S. Rinaldi, E. Sisinni, and M. Gidlund. Exploiting localization systems for LoRaWAN transmission scheduling in industrial applications. In *IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–8, May 2019.

[138] C. Wang, L. Liu, H. Ma, and D. Xia. SL-MAC: A Joint TDMA MAC Protocol for LEO Satellites Supported Internet of Things. In *International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pages 31–36, Dec 2018.

[139] Q. L. Hoang, W. Jung, T. Yoon, D. Yoo, and H. Oh. A Real-Time LoRa Protocol for Industrial Monitoring and Control Systems. *IEEE Access*, 8:44727–44738, Mar 2020.

[140] T. To and A. Duda. Timemaps for Improving Performance of LoRaWAN. In *IEEE International Conference on Communications (ICC)*, pages 1–7, Jun 2020.

[141] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund. Using LoRa for industrial wireless networks. In *IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, May 2017.

[142] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp. Traffic Models for Machine Type Communications. In *International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5, Aug 2013.

[143] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi. Modelling of IoT Traffic and Its Impact on LoRaWAN. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2017.

[144] M. O. Farooq and D. Pesch. Evaluation of Multi-Gateway LoRaWAN with Different Data Traffic Models. In *IEEE Conference on Local Computer Networks (LCN)*, pages 279–282, Oct 2018.

[145] W. Hsu, Q. Li, X. Han, and C. Huang. A hybrid IoT traffic generator for mobile network performance assessment. In *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 441–445, Jun 2017.

[146] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella, and K. Mahmood. Comparison of M2M Traffic Models Against Real World Data Sets. In *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6, Sep 2018.

[147] 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*, 23(5):60–67, Oct 2016.

[148] M. Zarrini and A. Ghasemi. Loss and delay analysis of non-Poisson M2M traffic over LTE networks. *Transactions on Emerging Telecommunications Technologies*, 29(2):e3273, Jan 2018.

[149] P. S. Dester, F. Helder C. dos S. Filho, and P. Cardieri. Performance Analysis of Uplink Traffic for Machine Type Communication in Wireless Sensor Networks. In *IEEE Vehicular Technology Conference (VTC Spring)*, pages 1–5, Jun 2018.

[150] D-Y. Kim and S. Kim. Dual-channel medium access control of low power wide area networks considering traffic characteristics in IoE. *Cluster Computing*, 20(3):2375–2384, Sep 2017.

[151] G. Corrales Madueno, Č. Stefanović, and P. Popovski. Reliable and Efficient Access for Alarm-Initiated and Regular M2M Traffic in IEEE 802.11ah Systems. *IEEE Internet of Things Journal*, 3(5):673–682, Oct 2016.

[152] S. Ali, A. Ferdowsi, W. Saad, and N. Rajatheva. Sleeping Multi-Armed Bandits for Fast Uplink Grant Allocation in Machine Type Communications. In *IEEE Global Communications Conference Workshops (GLOBECOM)*, pages 1–6, Dec 2018.

[153] K. A. Lizos. Heterogeneous scheduling in wireless networks: Machine to machine case study. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 675–682, Oct 2015.

[154] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor. Cognitive Hierarchy Theory for Distributed Resource Allocation in the Internet of Things. *IEEE Transactions on Wireless Communications*, 16(12):7687–7702, Dec 2017.

[155] Q. Yuan, J. Shang, X. Cao, C. Zhang, X. Geng, and J. Han. Detecting Multiple Periods and Periodic Patterns in Event Time Sequences. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 617–626, Nov 2017.

[156] T. Andreescu, D. Andrica, and I. Cucurezeanu. *Introduction to Diophantine Equations: a Problem-based Approach*. Birkhauser, Jul 2011.

[157] S. Böcker, C. Arendt, P. Jorke, and C. Wietfeld. LPWAN in the Context of 5G: Capability of LoRaWAN to Contribute to mMTC. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 737–742, Apr 2019.

[158] 3GPP. TR 37.868: Study on RAN Improvements for Machine-type communications. Technical report, 2012.

[159] A. 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)*, pages 1–6, Dec 2017.

[160] K. Wehrle, M. Gnes, and J. Gross. *Modeling and Tools for Network Simulation*. Springer Publishing Company, Incorporated, 1st edition, Sep 2010.

[161] U. Raza, P. Kulkarni, and M. Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, Jan 2017.

[162] LoRa Alliance. LoRa Alliance Homepage & Global Coverage Map. https://lora-alliance.org/, Jun 2020. [Online; accessed 5-Oct-2020].

[163] Tech Crunch. Sigfox Raises $115M To Take Its Internet of Things Network Global. https://techcrunch.com/2015/02/11/sigfox-raises-115m-to-take-its-internet-of-things-network-global/, February 2015. [Online; accessed 5-Oct-2020].

[164] A. Rico-Alvarino, M. Vajapeyam, H. Xu, X. Wang, Y. Blankenship, J. Bergman, T. Tirronen, and E. Yavuz. An overview of 3GPP enhancements on machine to machine communications. *IEEE Communications Magazine*, 54(6):14–21, Jun 2016.

[165] IEEE 802.11 Task Group. IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, Dec 2016.

[166] IEEE 802.15.4 Task Group. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, Jul 2020.

[167] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso. Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review. *IEEE Internet of Things Journal*, 6(2):2225–2240, Apr 2019.

[168] F. Gu, J. Niu, L. Jiang, X. Liu, and M. Atiquzzaman. Survey of the low power wide area network technologies. *Journal of Network and Computer Applications*, 149:102459, Oct 2020.

[169] ETSI. European Standard 300 220-1: Short Range Devices (SRD) operating in the frequency range 25 MHz to 1000 MHz; Part 1: Technical characteristics and methods of measurement, Feb 2017.

[170] J. Haxhibeqiri, A. Shahid, M. Saelens, J. Bauwens, B. Jooris, E. De Poorter, and J. Hoebeke. Sub-Gigahertz Inter-Technology Interference. How Harmful is it for LoRa? In *IEEE International Smart Cities Conference (ISC2)*, pages 1–7, Sep 2018.

[171] B. Reynders, W. Meert, and S. Pollin. Range and coexistence analysis of long range unlicensed communication. In *International Conference on Telecommunications (ICT)*, pages 1–6, May 2016.

[172] 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*, pages 1–11, Oct 2015.

[173] F. Samie, L. Bauer, and J. Henkel. IoT Technologies for Embedded Computing: A Survey. In *IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pages 1–10, Oct 2016.

[174] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. *IEEE Communications Surveys & Tutorials*, 22(1):371–388, Jan 2020.

[175] M. Aref and A. Sikora. Free space range measurements with Semtech LoRa technology. In *International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, pages 19–23, Sep 2014.

[176] Sigfox. Our Story. https://www.sigfox.com/en/sigfox-story, 2020. [Online; accessed 5-Oct-2020].

[177] K. E. Nolan, W. Guibene, and M. Y. Kelly. An evaluation of low power wide area network technologies for the Internet of Things. In *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 439–444, Sep 2016.

[178] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer. Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 197–202, Mar 2018.

[179] Sigfox. Coverage. https://www.sigfox.com/en/coverage, 2020. [Online; accessed 5-Oct-2020].

[180] A. Lavric, A. I. Petrariu, and V. Popa. Long Range SigFox Communication Protocol Scalability Analysis Under Large-Scale, High-Density Conditions. *IEEE Access*, 7:35816–35825, Mar 2019.

[181] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sorensen. Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT. In *IEEE Vehicular Technology Conference (VTC)*, pages 1–5, Jun 2017.

[182] LightReading. French Toast? Sigfox on Skid Row. https://www.lightreading.com/iot/iot-strategies/french-toast-sigfox-on-skid-row/d/d-id/740082, Jan 2018. [Online; accessed 5-Oct-2020].

[183] Y. P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi. A Primer on 3GPP Narrowband Internet of Things. *IEEE Communications Magazine*, 55(3):117–123, Mar 2017.

[184] B. Foubert and N. Mitton. Long-Range Wireless Radio Technologies: A Survey. *Future Internet*, 12(1):13, Jan 2020.

[185] T. Nakamura, S. Nagata, H. Umeda, H. Takahashi, and K. Ando. LTE-Advanced Release 13 Standardization Technology Overview. *NTT DOCOMO Technical Journal*, 18(2):1–7, Sep 2016.

[186] GSMA. GSMA Deployment Map. https://www.gsma.com/iot/deployment-map/, Jun 2020. [Online; accessed 5-Oct-2020].

[187] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert, and J. P. Koskinen. Overview of narrowband IoT in LTE Rel-13. In *IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–7, Oct 2016.

[188] C. Yu, L. Yu, Y. Wu, Y. He, and Q. Lu. Uplink scheduling and link adaptation for narrowband internet of things systems. *IEEE Access*, 5:1724–1734, Feb 2017.

[189] X. Lin, A. Adhikary, and Y. P. Eric Wang. Random Access Preamble Design and Detection for 3GPP Narrowband IoT Systems. *IEEE Wireless Communications Letters*, 5(6):640–643, Dec 2016.

[190] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh. NB-IoT system for M2M communication. In *IEEE Wireless Communications and Networking Conference*, pages 1–5, Apr 2016.

[191] O. Liberg, M. Sundberg, Y.-P. E. Wang, J. Bergman, J. Sachs, and G. Wikström. *Cellular Internet of Things*. Academic Press, second edition, Nov 2019.

[192] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz. A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. *IEEE Access*, 6:3619–3647, Dec 2018.

[193] D. Ismail, M. Rahman, and A. Saifullah. Low-Power Wide-Area Networks: Opportunities, Challenges, and Directions. In *Workshop Program of the International Conference on Distributed Computing and Networking (Workshops ICDCN)*, pages 1–6, Jan 2018.

[194] W. Yang, M. Wang, J. Zhang, J. Zou, M. Hua, T. Xia, and X. You. Narrowband Wireless Access for Low-Power Massive Internet of Things: A Bandwidth Perspective. *IEEE Wireless Communications*, 24(3):138–145, May 2017.

[195] Nokia Networks. LTE-M - Optimising LTE for the Internet of Things.

[196] Nwave. Nwave: Smart Parking Technology. https://www.nwave.io/parking-technology/, 2020. [Online; accessed 5-Oct-2020].

[197] Ingenu. An Educational Guide: How RPMA Works. https://www.ingenu.com/portfolio/how-rpma-works-white-paper/.

[198] Weightless SIG. LPWAN Technology Decisions: 17 critical features. http://www.weightless.org/membership/hvVs4ZGQqr5dwCDlBiYX, 2016. [Online; accessed 5-Oct-2020].

[199] Telensa. Smart City Solutions. https://www.telensa.com/solutions/, 2020. [Online; accessed 5-Oct-2020].

[200] WAVIoT. NB-Fi Specification. https://waviot.com/technology/nb-fi-specification/, 2020. [Online; accessed 5-Oct-2020].

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

[202] G. Ergeerts, M. Nikodem, D. Subotic, T. Surmacz, B. Wojciechowski, P. D. Meulenaere, and M. Weyn. DASH7 Alliance Protocol in Monitoring Applications. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 623–628, Nov 2015.

[203] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. Prévotet. Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility. *IEEE Communications Surveys & Tutorials*, 21(2):1561–1581, Oct 2019.

[204] W. Ayoub, F. Nouvel, A. E. Samhat, J. Prevotet, and M. Mroue. Overview and Measurement of Mobility in DASH7. In *International Conference on Telecommunications (ICT)*, pages 532–536, 2018.

[205] O. Cetinkaya and O. B. Akan. A DASH7-based power metering system. In *IEEE Consumer Communications and Networking Conference (CCNC)*, pages 406–411, Jan 2015.

[206] A. Marquet, N. Montavont, and G. Z. Papadopoulos. Investigating Theoretical Performance and Demodulation Techniques for LoRa. In *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, Jun 2019.

[207] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers. A Multi-Channel Software Decoder for the LoRa Modulation Scheme. In *INSTICC International Conference on Internet of Things, Big Data, and Security*, pages 41–51, Jan 2018.

[208] M. Knight and B. Seeber. Decoding LoRa: Realizing a Modern LPWAN with SDR. *GNU Radio Conference*, 1(1):1–5, Sep 2016.

[209] LoRa Alliance Technical Marketing Workgroup. LoRaWAN - What is it?: A Technical Overview of LoRa and LoRaWAN.

[210] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[211] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[212] R. Pickholtz, D. Schilling, and L. Milstein. Theory of Spread-Spectrum Communications - A Tutorial. *IEEE Transactions on Communications*, 30(5):855–884, May 1982.

[213] Semtech. AN1200.02 - LoRa Modulation Basics, May 2015.

[214] A. Springer, W. Gugler, M. Huemer, L. Reindl, C. C. W. Ruppel, and R. Weigel. Spread spectrum communications using chirp signals. In *IEEE/AFCEA Information Systems for Enhanced Public Safety and Security (EUROCOMM)*, pages 166–170, May 2000.

[215] Y. Tsai and J. Chang. The feasibility of combating multipath interference by chirp spread spectrum techniques over Rayleigh and Rician fading channels. In *IEEE International Symposium on Spread Spectrum Techniques and Applications (ISSSTA)*, pages 282–286, Jun 1994.

[216] C. Bernier, F. Dehmas, and N. Deparis. Low Complexity LoRa Frame Synchronization for Ultra-Low Power Software-Defined Radios. In *IEEE Transactions on Communications*, volume 68, pages 3140–3152, Feb 2020.

[217] T. T. Nguyen, H. H. Nguyen, R. Barton, and P. Grossetete. Efficient Design of Chirp Spread Spectrum Modulation for Low-Power Wide-Area Networks. *IEEE Internet of Things Journal*, 6(6):9503–9515, Dec 2019.

[218] L. Vangelista. Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters*, 24(12):1818–1821, Dec 2017.

[219] O. Afisiadis, M. Cotting, A. Burg, and A. Balatsoukas-Stimming. On the Error Rate of the LoRa Modulation With Interference. *IEEE Transactions on Wireless Communications*, 19(2):1292–1304, Feb 2020.

[220] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance. *IEEE Communications Letters*, 22(4):796–799, Apr 2018.

[221] H. Mroue, A. Nasser, B. Parrein, S. Hamrioui, E. Mona-Cruz, and G. Rouyer. Analytical and Simulation study for LoRa Modulation. In *IEEE International Conference on Telecommunications (ICT)*, pages 655–659, Jun 2018.

[222] T. Elshabrawy and J. Robert. Closed-Form Approximation of LoRa Modulation BER Performance. *IEEE Communications Letters*, 22(9):1778–1781, Sep 2018.

[223] D. Bankov, E. Khorov, and A. Lyakhov. Mathematical model of LoRaWAN channel access with capture effect. In *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5, Oct 2017.

[224] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke. LoRa Scalability: A Simulation Model Based on Interference Measurements. *Sensors*, 17(6), May 2017.

[225] T. Elshabrawy and J. Robert. Analysis of BER and Coverage Performance of LoRa Modulation under Same Spreading Factor Interference. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6, Sep 2018.

[226] A. A. Doroshkin, A. M. Zadorozhny, O. N. Kus, V. Y. Prokopyev, and Y. M. Prokopyev. Experimental Study of LoRa Modulation Immunity to Doppler Effect in CubeSat Radio Communications. *IEEE Access*, 7:75721–75731, May 2019.

[227] LoRa Alliance. LoRaWAN Specification v1.1. https://lora-alliance.org/resource-hub/lorawanr-specification-v11, Oct 2017. [Online; accessed 5-Oct-2020].

[228] Ladislav Polak and Jiri Milos. Performance analysis of LoRa in the 2.4 GHz ISM band: coexistence issues with Wi-Fi. *Telecommunication Systems: Modelling, Analysis, Design and Management*, 74(3):299–309, Jul 2020.

[229] J. R. Cotrim and J. H. Kleinschmidt. LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication. *Sensors*, 20(15):4273, Jul 2020.

[230] A. Osorio, M. Calle, J. D. Soto, and J. E. Candelo-Becerra. Routing in LoRaWAN: Overview and Challenges. *IEEE Communications Magazine*, 58(6):72–76, Jun 2020.

[231] M. Costa, T. Farrell, and L. Doyle. On energy efficiency and lifetime in Low Power Wide Area Network for the Internet of Things. In *IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 258–263, Sep 2017.

[232] D. Bankov, E. Khorov, and A. Lyakhov. Mathematical model of lorawan channel access. In *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, Jun 2017.

[233] Semtech. LoRaWAN - simple rate adaptation recommended algorithm. https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f5 Oct 2016. [Online; accessed 5-Oct-2020].

[234] The Things Network. The Things Network Technology Stack. https://www.thethingsnetwork.org/tech-stack, 2020. [Online; accessed 5-Oct-2020].

[235] ChirpStack. ChirpStack open-source LoRaWAN Network Server . https://www.chirpstack.io/network-server/, 2020. [Online; accessed 5-Oct-2020].

[236] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub. Toward Massive Machine Type Cellular Communications. *IEEE Wireless Communications*, 24(1):120–128, Feb 2017.

[237] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy. Massive Machine-Type Communications in 5G: physical and MAC-layer solutions. *IEEE Communications Magazine*, 54(9):59–65, Sep 2016.

[238] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang. Large-Scale Measurement and Characterization of Cellular Machine-to-Machine Traffic. *IEEE/ACM Transactions on Networking*, 21(6):1960–1973, Jul 2013.

[239] E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour. A Survey of Traffic Issues in Machine-to-Machine Communications Over LTE. *IEEE Internet of Things Journal*, 3(6):865–884, Dec 2016.

[240] 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 (WNS3)*, pages 61–68, Jun 2018.

[241] T. To and A. Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.

[242] G. Callebaut, G. Ottoy, and L. van der Perre. Cross-Layer Framework and Optimization for Efficient Use of the Energy Budget of IoT Nodes. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, Apr 2019.

References for LPWAN Use Cases Table

[243] R. K. Kodali, K. Y. Borra, S. S. G. N., and H. J. Domma. An IoT Based Smart Parking System Using LoRa. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 151–1513, Oct 2018.

[244] T. Perković, P. Šolić, H. Zargariasl, D. Čoko, and Joel J.P.C. Rodrigues. Smart Parking Sensors: State of the Art and Performance Evaluation. *Journal of Cleaner Production*, 262:121181, Mar 2020.

[245] T. Nguyen Gia, J. Pena Queralta, and T. Westerlund. Exploiting LoRa, edge, and fog computing for traffic monitoring in smart cities. In *LPWAN Technologies for IoT and M2M Applications*, pages 347 – 371. Academic Press, Mar 2020.

[246] J. Santa, R. Sanchez-Iborra, P. Rodriguez-Rey, L. Bernal-Escobedo, and A. F. Skarmeta. LPWAN-Based Vehicular Monitoring Platform with a Generic IP Network Interface. *Sensors*, 19(2), Jan 2019.

[247] A. Aneiba, B. Nangle, J. Hayes, and M. Albaarini. Real-Time IoT Urban Road Traffic Data Monitoring Using LoRaWAN. In *International Conference on the Internet of Things (IoT)*, Oct 2019.

[248] L. Zhao, Q. Gao, R. Wang, N. Fang, Z. Jin, N. Wan, and L. Xu. Intelligent Street Light System Based on NB-IoT and Energy-saving Algorithm. In *International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6, Jun 2018.

[249] N. Sukhathai and T. Tayjasanant. Smart Street Lighting System with Networking Communication. In *IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pages 2826–2831, May 2019.

[250] F. Sánchez Sutil and A. Cano-Ortega. Smart Public Lighting Control and Measurement System Using LoRa Network. *Electronics*, 9(1):124, Jan 2020.

[251] D. Croce, D. Garlisi, F. Giuliano, A. L. Valvo, S. Mangione, and I. Tinnirello. Performance of LoRa for Bike-Sharing Systems. In *International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pages 1–6, Jul 2019.

[252] Á. Lozano, J. Caridad, J. F. De Paz, G. Villarrubia González, and J. Bajo. Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization. *Sensors*, 18(5), May 2018.

[253] T. Boshita, H. Suzuki, and Y. Matsumoto. IoT-based Bus Location System Using LoRaWAN. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 933–938, Nov 2018.

[254] Y. Li, L. Yang, S. Han, X. Wang, and F. Wang. When LPWAN Meets ITS: Evaluation of Low Power Wide Area Networks for V2X Communications. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 473–478, Nov 2018.

[255] P. Ferrari, A. Flammini, S. Rinaldi, M. Rizzi, and E. Sisinni. On the use of LPWAN for EVehicle to grid communication. In *AEIT International Annual Conference*, pages 1–6, Sep 2017.

[256] Y. Chou, Y. Mo, J. Su, W. Chang, L. Chen, J. Tang, and C. Yu. i-Car system: A LoRa-based low power wide area networks vehicle diagnostic system for driving safety. In *International Conference on Applied System Innovation (ICASI)*, pages 789–791, May 2017.

[257] A. Ouya, B. M. De Aragon, C. Bouette, G. Habault, N. Montavont, and G. Z. Papadopoulos. An efficient electric vehicle charging architecture based on LoRa communication. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 381–386, Oct 2017.

[258] O. Jo, Y. Kim, and J. Kim. Internet of Things for Smart Railway: Feasibility and Applications. *IEEE Internet of Things Journal*, 5(2):482–490, Apr 2018.

[259] 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*, 19:1510, Mar 2019.

[260] T. Polonelli, D. Brunelli, M. Guermandi, and L. Benini. An accurate low-cost Crackmeter with LoRaWAN communication and energy harvesting capability. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 671–676, Sep 2018.

[261] M. Sidorov, P. V. Nhut, Y. Matsumoto, and R. Ohmura. LoRa-Based Precision Wireless Structural Health Monitoring System for Bolted Joints in a Smart City Environment. *IEEE Access*, 7:179235–179251, Dec 2019.

[262] A. Perles, E. Pérez-Marín, R. Mercado, J. D. Segrelles, I. Blanquer, M. Zarzo, and F. J. Garcia-Diego. An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage. *Future Generation Computer Systems*, 81:566 – 581, Jun 2018.

[263] M. T. Marshall. Interacting with Heritage: On the Use and Potential of IoT Within the Cultural Heritage Sector. In *International Conference on Internet of Things: Systems, Management and Security*, pages 15–22, Oct 2018.

[264] T. Addabbo, A. Fort, M. Mugnaini, E. Panzardi, A. Pozzebon, and V. Vignoli. A city-scale IoT architecture for monumental structures monitoring. *Measurement*, 131:349–357, Aug 2019.

[265] H. Zhang, L. Li, and X. Liu. Development and Test of Manhole Cover Monitoring Device Using LoRa and Accelerometer. *IEEE Transactions on Instrumentation and Measurement*, 69(5):2570–2580, May 2020.

[266] Y. Ban and M. Kai. IoT Equipment Structure with Reduced Risk of Damage on Attachable to Manhole Cover. In *IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, pages 1363–1364, Jul 2019.

[267] S. J. Johnston, P. J. Basford, F. Bulot, M. Apetroaie-Cristea, N. Easton, C. Davenport, G. Foster, M. Loxham, A. Morris, and S. Cox. City Scale Particulate Matter Monitoring Using LoRaWAN Based Air Quality IoT Devices. *Sensors*, 19(1):209, Jan 2019.

[268] B. Braem, S. Latre, P. Leroux, P. Demeester, T. Coenen, and P. Ballon. Designing a smart city playground: Real-time air quality measurements and visualization in the City of Things testbed. In *IEEE International Smart Cities Conference (ISC2)*, pages 1–2, Sep 2016.

[269] A. Candia, S. Represa, D. Giuliani, M. Angel Luengo, A. Porta, and L. Armando Marrone. Solutions for SmartCities: proposal of a monitoring system of air quality based on a LoRaWAN network with low-cost sensors. In *Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI)*, pages 1–6, Nov 2018.

[270] L. Chen, H. Huang, C. Wu, Y. Tsai, and Y. Chang. A LoRa-Based Air Quality Monitor on Unmanned Aerial Vehicle for Smart City. In *International Conference on System Science and Engineering (ICSSE)*, pages 1–5, Jun 2018.

[271] T. Addabbo, A. Fort, M. Mugnaini, L. Parri, A. Pozzebon, and V. Vignoli. Smart Sensing in Mobility: a LoRaWAN Architecture for Pervasive Environmental Monitoring. In *IEEE International forum on Research and Technology for Society and Industry (RTSI)*, pages 421–426, Sep 2019.

[272] R. Yasmin, J. Petäjäjärvi, K. Mikhaylov, and A. Pouttu. Large and Dense LoRaWAN Deployment to Monitor Real Estate Conditions and Utilization Rate. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6, Sep 2018.

[273] B. Ngom, M. Diallo, B. Gueye, and N. Marilleau. LoRa-based Measurement Station for Water Quality Monitoring: Case of Botanical Garden Pool. In *IEEE Sensors Applications Symposium (SAS)*, pages 1–4, Mar 2019.

[274] S. R. Niya, S. S. Jha, T. Bocek, and B. Stiller. Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN. In *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 1–4, Apr 2018.

[275] Y. Chen and D. Han. Water quality monitoring in smart city: A pilot project. *Automation in Construction*, 89:307–316, Feb 2018.

[276] C. Dupont, M. Vecchio, C. Pham, B. Diop, C. Dupont, and S. Koffi. An Open IoT Platform to Promote Eco-Sustainable Innovation in Western Africa: Real Urban and Rural Testbeds. *Wireless Communications and Mobile Computing*, 2018:1–17, Jun 2018.

[277] M. E. E. E. Alahi, N. Pereira-Ishak, S. C. Mukhopadhyay, and L. Burkitt. An Internet-of-Things Enabled Smart Sensing System for Nitrate Monitoring. *IEEE Internet of Things Journal*, 5(6):4409–4417, Dec 2018.

[278] T. A. Balushi, A. Al Hosni, H. A. Theeb Ba Omar, and D. Al Abri. A LoRaWAN-based Camel Crossing Alert and Tracking System. In *IEEE International Conference on Industrial Informatics (INDIN)*, pages 1035–1040, Jul 2019.

[279] E. D. Ayele, N. Meratnia, and P. J. M. Havinga. Towards a New Opportunistic IoT Network Architecture for Wildlife Monitoring System. In *IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Feb 2018.

[280] W. Guibene, J. Nowack, N. Chalikias, K. Fitzgibbon, M. Kelly, and D. Prendergast. Evaluation of LPWAN Technologies for Smart Cities: River Monitoring Use-Case. In *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–5, Mar 2017.

[281] W. Xu, J. Zhang, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu. The Design, Implementation, and Deployment of a Smart Lighting System for Smart Buildings. *IEEE Internet of Things Journal*, 6(4):7266–7281, Aug 2019.

[282] T. Polonelli, D. Brunelli, A. Bartolini, and L. Benini. A LoRaWAN Wireless Sensor Network for Data Center Temperature Monitoring. In *Applications in Electronics Pervading Industry, Environment and Society*, pages 169–177, May 2019.

[283] W. Li, G. Shen, and J. Zhang. An Indoor Environmental Monitoring System for Large Buildings Based on LoRaWAN. In *Conference on Research in Adaptive and Convergent Systems (RACS)*, pages 34–38, Sep 2019.

[284] J. Petäjäjärvi, K. Mikhaylov, M. Hamalainen, and J. Iinatti. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In *International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–5, Mar 2016.

[285] G. Yang and H. Liang. A Smart Wireless Paging Sensor Network for Elderly Care Application Using LoRaWAN. *IEEE Sensors Journal*, 18(22):9441–9448, Nov 2018.

[286] F. Wu, T. Wu, and M. Yuce. An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications. *Sensors*, 19(1):21, Jan 2019.

[287] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel, and B. Freisleben. IoT-based health monitoring via LoRaWAN. In *IEEE International Conference on Smart Technologies (EUROCON)*, pages 519–524, Jul 2017.

[288] J. Pena Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund. Edge-AI in LoRa-based Health Monitoring: Fall Detection System with Fog Computing and LSTM Recurrent Neural Networks. In *International Conference on Telecommunications and Signal Processing (TSP)*, pages 601–604, Jul 2019.

[289] D. Fernandes Carvalho, P. Ferrari, E. Sisinni, P. Bellitti, N. F. Lopomo, and M. Serpelloni. Using LPWAN Connectivity for Elderly Activity Monitoring in Smartcity Scenarios. In *Applications in Electronics Pervading Industry, Environment and Society*, pages 81–87, Mar 2020.

[290] C. Tan and H. Tan. Evaluation of Sigfox LPWAN for sensor-enabled homes to identify at risk community dwelling seniors. In *IEEE Conference on Local Computer Networks (LCN)*, pages 26–33, Oct 2019.

[291] J. M. Howerton and B. L. Schenck. The Deployment of a LoRaWAN-Based IoT Air Quality Sensor Network for Public Good. In *Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6, Apr 2020.

[292] A. Sales Mendes, D. Jiménez-Bravo, M. Navarro-Cáceres, V. Reis Quietinho Leithardt, and G. Villarrubia González. Multi-Agent Approach Using LoRaWAN Devices: An Airport Case Study. *Electronics*, 9(9):1430, Sep 2020.

[293] P. Fedchenkov, A. Zaslavsky, and I. Sosunova. Enabling Smart Waste Management with Sensorized Garbage Bins and Low Power Data Communications Network. In *International Conference on the Internet of Things (IoT)*, Oct 2017.

[294] M. Cerchecci, F. Luti, A. Mecocci, S. Parrino, G. Peruzzi, and A. Pozzebon. A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context. *Sensors*, 18(4):1282, Apr 2018.

[295] S. Dong, S. Duan, Q. Yang, J. Zhang, G. Li, and R. Tao. MEMS-Based Smart Gas Metering for Internet of Things. *IEEE Internet of Things Journal*, 4(5):1296–1303, Oct 2017.

[296] A. Abrardo, A. Fort, E. Landi, M. Mugnaini, E. Panzardi, and A. Pozzebon. Black Powder Flow Monitoring in Pipelines by Means of Multi-Hop LoRa Networks. In *Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*, pages 312–316, Jun 2019.

[297] M. Saravanan, A. Das, and V. Iyer. Smart water grid management using LPWAN IoT technology. In *Global Internet of Things Summit (GIoTS)*, pages 1–6, Jun 2017.

[298] M. de Castro Tomé, P. H. J. Nardelli, and H. Alves. Long-Range Low-Power Wireless Networks and Sampling Strategies in Electricity Metering. *IEEE Transactions on Industrial Electronics*, 66(2):1629–1637, Feb 2019.

[299] C. Paolini, H. Adigal, and M. Sarkar. Upper Bound on LoRa Smart Metering Uplink Rate. In *IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4, Jan 2020.

[300] F. G. Reck, M. Sperandio, C. H. Barriquello, F. Gabriel Carloto, L. Maziero, and W. D. Vizzotto. Fault Indication in Distribution Systems Using an LPWAN Network. In *IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, pages 1–6, Sep 2019.

[301] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang. Smart Choice for the Smart Grid: Narrowband Internet of Things (NB-IoT). *IEEE Internet of Things Journal*, 5(3):1505–1515, Jun 2018.

[302] Y. Song, J. Lin, M. Tang, and S. Dong. An Internet of Energy Things Based on Wireless LPWAN. *Engineering*, 3(4):460 – 466, May 2017.

[303] K. Mikhaylov, A. Moiz, A. Pouttu, J. M. Martín Rapún, and S. A. Gascon. LoRaWAN for Wind Turbine Monitoring: Prototype and Practical Deployment. In *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–6, Nov 2018.

[304] L. Germani, V. Mecarelli, G. Baruffa, L. Rugini, and F. Frescura. An IoT Architecture for Continuous Livestock Monitoring Using LoRa LPWAN. *Electronics*, 8(12):1435, Dec 2019.

[305] N. Zinas, S. Kontogiannis, G. Kokkonis, S. Valsamidis, and I. Kazanidis. Proposed Open Source Architecture for Long Range Monitoring. The Case Study of Cattle Tracking at Pogoniani. In *Pan-Hellenic Conference on Informatics (PCI)*, Sep 2017.

[306] A. Grunwald, M. Schaarschmidt, and C. Westerkamp. LoRaWAN in a rural context: Use cases and opportunities for agricultural businesses. In *ITG-Symposium; Mobile Communication - Technologies and Applications*, pages 1–6, May 2019.

[307] G. Codeluppi, A. Cilfone, L. Davoli, and G. Ferrari. LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture. *Sensors*, 20(7), Apr 2020.

[308] Y. S. Chang, Y. Hsiung Chen, and S. K. Zhou. A smart lighting system for greenhouses based on Narrowband-IoT communication. In *International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, pages 275–278, Oct 2018.

[309] M. R. Ramli, P. T. Daely, D. Kim, and J. M. Lee. IoT-based adaptive network mechanism for reliable smart farm system. *Computers and Electronics in Agriculture*, 170:105287, Feb 2020.

[310] W. Zhao, S. Lin, J. Han, R. Xu, and L. Hou. Design and Implementation of Smart Irrigation System Based on LoRa. In *IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, Dec 2017.

[311] C. Cambra, S. Sendra, J. Lloret, and L. Garcia. An IoT service-oriented system for agriculture monitoring. In *IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[312] M. Á. Guillén-Navarro, F. Pereñíguez-García, and R. Martínez-España. IoT-based System to Forecast Crop Frost. In *International Conference on Intelligent Environments (IE)*, pages 28–35, Aug 2017.

[313] S. Kontogiannis, G. Kokkonis, S. Ellinidou, and S. Valsamidis. Proposed Fuzzy-NN Algorithm with LoRa Communication Protocol for Clustered Irrigation Systems. *Future Internet*, 9(4):78, Nov 2017.

[314] M. Usmonov and F. Gregoretti. Design and implementation of a LoRa based wireless control for drip irrigation systems. In *International Conference on Robotics and Automation Engineering (ICRAE)*, pages 248–253, Dec 2017.

[315] L. Joris, F. Dupont, P. Laurent, P. Bellier, S. Stoukatch, and J. Redouté. An Autonomous Sigfox Wireless Sensor Node for Environmental Monitoring. *IEEE Sensors Letters*, 3(7):01–04, Jul 2019.

[316] P. Spachos. Towards a Low-Cost Precision Viticulture System Using Internet of Things Devices. *IoT*, 1(1):5–20, Feb 2020.

[317] A. Renehan, B. Rombach, A. Haikl, C. Nolan, W. Lupton, E. Timmons, and R. Bailey. Low Power Wireless Networks in Vineyards. In *Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6, Apr 2020.

[318] A. Valente, S. Silva, D. Duarte, F. Cabral Pinto, and S. Soares. Low-Cost LoRaWAN Node for Agro-Intelligence IoT. *Electronics*, 9(6):987, Jun 2020.

[319] E. Pietrosemoli, M. Rainone, and M. Zennaro. On Extending the Wireless Communications Range of Weather Stations Using LoRaWAN. In *EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs)*, pages 78–83, Sep 2019.

[320] O. Debauche, M. El Moulat, S. Mahmoudi, S. Boukraa, P. Manneback, and F. Lebeau. Web Monitoring of Bee Health for Researchers and Beekeepers Based on the Internet of Things. *Procedia Computer Science*, 130:991–998, Apr 2018.

[321] D. Brunelli, A. Albanese, D. d'Acunto, and M. Nardello. Energy Neutral Machine Learning Based IoT Device for Pest Detection in Precision Agriculture. *IEEE Internet of Things Magazine*, 2(4):10–13, Dec 2019.

[322] S. Gil-Lebrero, F. J. Quiles-Latorre, M. Ortiz-López, V. Sánchez-Ruiz, V. Gámiz-López, and J. J. Luna-Rodríguez. Honey Bee Colonies Remote Monitoring System. *Sensors*, 17(1):55, Jan 2017.

[323] R. Pueyo Centelles, F. Freitag, R. Meseguer, L. Navarro, S. F. Ochoa, and R. M. Santos. A LoRa-Based Communication System for Coordinated Response in an Earthquake Aftermath. *International Conference on Ubiquitous Computing and Ambient Intelligence (UCaMI)*, 31(1):73, Nov 2019.

[324] P. Boccadoro, B. Montaruli, and L. A. Grieco. QuakeSense, a LoRa-compliant Earthquake Monitoring Open System. In *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–8, Oct 2019.

[325] S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez. LoRaWAN Network for Fire Monitoring in Rural Environments. *Electronics*, 9(3):531, Mar 2020.

[326] T. Brito, A. Pereira, J. Lima, and A. Valente. Wireless Sensor Network for Ignitions Detection: An IoT approach. *Electronics*, 9(6):893, May 2020.

[327] J. Kang and S. Adibi. Bushfire Disaster Monitoring System Using Low Power Wide Area Networks (LPWAN). *Technologies*, 5(4):65, Oct 2017.

[328] M. Antunes, L. M. Ferreira, C. Viegas, A. P. Coimbra, and A. T. de Almeida. Low-Cost System for Early Detection and Deployment of Countermeasures Against Wild Fires. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 418–423, Apr 2019.

[329] E. Leon, C. Alberoni, M. Wister, and J. Hernández-Nolasco. Flood Early Warning System by Twitter Using LoRa. *International Conference on Ubiquitous Computing and Ambient Intelligence (UCaMI)*, 2(19):1213, Oct 2018.

[330] A. Douinot, A. Dalla Torre, J. Martin, J. Iffly, L. Rapin, C. Meisch, C. Bastian, and L. Pfister. Prototype of a LPWA Network for Real-Time Hydro-Meteorological Monitoring and Flood Nowcasting. In *Ad-Hoc, Mobile, and Wireless Networks*, pages 566–574, Sep 2019.

[331] M. Ragnoli, G. Barile, A. Leoni, G. Ferri, and V. Stornelli. An Autonomous Low-Power LoRa-Based Flood-Monitoring System. *Journal of Low Power Electronics and Applications*, 10(2):15, May 2020.

[332] S. M. Karunarathne, M. Dray, L. Popov, M. Butler, C. Pennington, and C. M. Angelopoulos. A technological framework for data-driven IoT systems: Application on landslide monitoring. *Computer Communications*, 154:298–312, Feb 2020.

[333] M. El Moulat, O. Debauche, S. Mahmoudi, L. Aït Brahim, P. Manneback, and F. Lebeau. Monitoring System Using Internet of Things For Potential Landslides. *Procedia Computer Science*, 134:26–34, Jul 2018.

[334] R. Fekih Romdhane, Y. Lami, D. Genon-Catalot, N. Fourty, A. Lagréze, D. Jongmans, and L. Baillet. Wireless sensors network for landslides prevention. In *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 222–227, Jun 2017.

[335] I. F. Priyanta, F. Golatowski, T. Schulz, and D. Timmermann. Evaluation of LoRa Technology for Vehicle and Asset Tracking in Smart Harbors. In *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 4221–4228, Oct 2019.

[336] O. Poenicke, M. Kirch, K. Richter, and S. Schwarz. LoRaWAN for IoT Applications in Air Cargo – Development of a GSE Tracking System for DHL Air Cargo Hub Leipzig. In *European Conference on Smart Objects, Systems and Technologies (Smart SysTech)*, pages 1–6, Jun 2018.

[337] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke. LoRa indoor coverage and performance in an industrial environment: Case study. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Sep 2017.