
Novel Medium Access Protocols for LoRa Internet of Things Networks

Hanan Jabr Alahmadi

A thesis submitted in partial fulfilment of the requirements of
Edinburgh Napier University, for the award of Doctor of
Philosophy


June, 2024

Authorship Declaration

I, Hanan Alahmadi, confirm that this thesis submitted for assessment is my own work and is expressed in my own words. Any uses made within it of the works of other authors in any form e.g., ideas, equations, figures, text, tables, programs, etc. are properly acknowledged and referenced.

Signed: Hanan Alahmadi

Date: Tuesday 11th June, 2024

Matriculation no: 

Abstract

LoRa, which stands for Long Range, is a network technology that provides a long transmission range while maintaining a low power consumption profile, enabling a wide range of Internet of Things (IoT) applications such as smart cities, smart monitoring, and smart agriculture. However, due to the nodes' ALOHA-based access method, reliable performance in high-density networks may be difficult to achieve, limiting such networks' scalability. Moreover, the Duty Cycle (DC) restrictions imposed on nodes and gateway transmissions to regulate the access to the unlicensed shared ISM band can further constrain network scalability. Furthermore, LoRa networks provide multiple configurable transmission parameters that greatly affect the performance of the overall network. To the best of our knowledge, the optimal combination of these parameters that can allow orthogonal simultaneous transmissions to be successfully decoded by the gateway has not been reported in the literature. Motivated by the aforementioned challenges, this thesis aims to address the main challenges of LoRa networks and conceive comprehensive solutions considering all these challenges such that the network scalability and the energy efficiency are maximized. The main contribution of this thesis is the proposing of decentralized approaches that optimize the network performance without burdening the network with extensive control packets from the network server. To address the challenges mentioned earlier, the thesis first develops a distribution algorithm, named Sensitivity-Aware LoRa (SAL), for LoRa transmission parameters that maximize the network scalability by minimizing the Packet Error Rate (PER). The main contribution of SAL algorithm is doubling the available DC by considering channels from all sub-bands. Then, an autonomous Time-Division Multiple Access (TDMA) protocol named Sector-Based Time-Slotted (SBTS-LoRa) was proposed to minimize the collisions resulting from the ALOHA random channel access and hence improving the network scalability. The collision rate of SBTS-LoRa was enhanced by 49% compared to the legacy LoRaWAN. After that, the SBTS-LoRa was enhanced to consider the dynamic and relatively short time frame sizes, which are presented in the autonomous Adaptive Frame Size (ATS-LoRa) MAC protocol. The main novelty of the proposed protocol is that the time frame size depends on the node density, which uses common transmission parameters to avoid collisions among them. According to that, the network throughput was enhanced three times compared to the SBTS-LoRa protocol. Moreover, all the proposed protocols were evaluated in realistic environmental settings where large-scale and dense networks were considered.

Contents

1	Introduction	1
1.1	Problem Statement and Research Questions	4
1.2	Research Aim and Objectives	5
1.3	Thesis Structure	6
2	Overview of Long Range (LoRa) Networks	9
2.1	Common Long Range Technologies	9
2.1.1	Sigfox	10
2.1.2	INGENU RPMA	11
2.1.3	NB-IoT	11
2.2	LoRa Technological Context	12
2.3	LoRa Physical Layer	14
2.4	LoRaWAN MAC Layer	20
2.5	The Adaptive Data Rate (ADR) Algorithm of LoRaWAN Protocol	21
2.6	LoRa Deployment for Industrial Applications	22
2.7	LoRa Performance Goals	25
2.8	Conclusion	28
3	Assessing the Effectiveness and Reliability of LoRaWAN MAC and Data Link Layers' protocols: A Comprehensive Review	30
3.1	ALOHA Medium Access Mechanism Alternatives	30
3.1.1	Slotted-ALOHA Channel Access: S-ALOHA	31
3.1.2	Time Division Multiple Access Mechanism	38
3.1.3	Contention-Based Medium Access Protocols	50
3.2	Addressing the Limited Duty Cycle	57
3.2.1	Distributed-Based LoRa Configuration	58
3.2.2	Aggregated Acknowledgments	65
3.2.3	Hybrid Solutions	68
3.3	Configuration of Node's Parameters	74
3.3.1	Spreading Factor Configuration	75
3.3.2	Spreading Factor and Transmission Power Configuration	77
3.3.3	Spreading Factor, Transmission Power, and Channel Configuration	81
3.4	Conclusion	88

4	Sensitivity-Aware Configurations for High Packet Generation Rate LoRa Networks	89
4.1	Problem Statement	89
4.2	The Proposed Solution	90
4.2.1	The Initialization Phase	93
4.2.2	SAL Operational Modes	94
4.2.3	Data Transmission Phase	95
4.3	Performance Evaluation	95
4.3.1	Packet Error Rate (PER)	97
4.3.2	The Probability of the Capture Effect	98
4.3.3	End-to-End Delay	98
4.3.4	Packet Delivery Ratio (PDR)	99
4.3.5	Throughput	101
4.3.6	Energy per Bit (EpB)	101
4.4	Conclusion	102
5	SBTS-LoRa: Sector-Based Time-Slotted MAC Protocol for LoRa Networks	104
5.1	Background and Problem Statement	105
5.1.1	Scalability Issue	106
5.1.2	Collisions Due to ALOHA Access Method	106
5.1.3	The Duty Cycle Restrictions	107
5.2	Collision Probability Model of LoRaWAN	107
5.2.1	Calculating $\Pr\{B\}$	109
5.2.2	Calculating $\Pr\{C\}$	111
5.2.3	Calculating $\Pr\{D\}$	113
5.3	Annulus-Based Distribution of LoRa Transmission Parameters	113
5.4	Sector-Based Time-Slotted LoRa (SBTS-LoRa) Protocol Description	119
5.4.1	Overview	119
5.4.2	The Initialization Phase	120
5.4.3	Data Transmission Phase	124
5.4.4	Finding the Optimal Frame Size	126
5.5	Performance Evaluation	127
5.5.1	EXPLoRa-AT Algorithm	128
5.5.2	The Bit Rate Ratio (BRR) Algorithm	129
5.5.3	The Probability of Collision	133
5.5.4	The End-to-End Delay	134
5.5.5	The Throughput	135
5.5.6	The Packet Error Rate (PER)	136
5.5.7	The Energy Consumption (Energy per Bit)	138

5.6	Conclusion	138
6	Autonomous Adaptive Frame Size for Time-Slotted LoRa MAC Protocol	140
6.1	Protocol Description	141
6.1.1	Determining the SF Transmission Parameter	142
6.1.2	Equal Radius-Based Corona (ERBC) Mode	144
6.1.3	Equal Area-Based Corona (EABC) Mode	145
6.1.4	Equal Area-Based Sector (EABS) Mode	148
6.1.5	Frame Sizes Comparison	150
6.2	Performance Evaluation	152
6.2.1	The Probability of Collisions	154
6.2.2	The End-to-End Delay	155
6.2.3	Network Throughput	156
6.2.4	The Energy Consumption	157
6.3	Conclusion	158
7	Conclusion	160
7.1	Concluding Remarks	160
7.2	Future Work Perspectives	164
7.2.1	ALOHA Medium Access Mechanism Alternatives	164
7.2.2	Addressing the Limited Duty Cycle	164
7.2.3	Configuration of Node's Parameters	165

List of Tables

1	The main technical features of Sigfox, LoRa, Ingenu, and NB-IoT LPWA technologies.	13
2	SF estimated values.	16
3	Some large-scale applications of LoRa networks.	24
4	S-ALOHA based research works to enhance LoRa networks.	35
5	Beacon-based TDMA MAC protocols for LoRa networks.	42
6	Synchronization-based TDMA MAC protocols for LoRa networks.	48
7	Contention-based MAC protocols for LoRa networks.	54
8	Distributed-based approaches to address the limited DC in LoRa networks.	62
9	Aggregated ACK approaches to address the limited DC of downlink traffic in LoRa networks.	66
10	Hybrid solutions to overcome DC limit of LoRa networks.	72
11	Enhancing the transmission parameters configurations of LoRa node's.	82
12	Estimation of the maximum distance in meters per each (SF, TX)	91
13	Duty cycle per channels	92
14	Distribution of the transmission power levels on different channels.	93
15	Simulation parameters used to evaluate SAL algorithm.	97
16	Notations and descriptions.	114
17	Transmission parameters distribution among cells.	117
18	Simulation parameters for evaluating SBTS-LoRa protocol	128
19	A summary of performance metrics.	138
20	Simulation parameters used to evaluate the ATS-LoRa.	153
21	MAX eligible distance for each spreading factor SF [1].	154

List of Figures

1	The network architecture of LoRa networks.	10
2	Comparison between Sigfox, LoRa, RPMA, and NB-IoT technologies in terms of the data rate, the maximum range for urban areas, and the maximum payload.	12
3	The effect of spreading factors on the data rate and transmission range [2].	15
4	The relationship between the spreading factor and symbol duration [3].	17
5	The ToA and the BR with SF=10 and BW=125 KHz	18
6	Receiving slots scheduling.	20
7	Performance goals of LoRa networks	26
8	Classification of the related work based on the main limitations of LoRa networks.	31
9	Packet Error Rate (PER) of SAL algorithm.	98
10	The probability of the capture effect of SAL algorithm.	99
11	The end-to-end delay of SAL algorithm.	100
12	The Packet Delivery Ratio (PDR) of SAL algorithm.	100
13	Network throughput of SAL algorithm.	101
14	Energy consumed per bit in SAL algorithm.	102
15	Probability collision model of LoRaWAN	109
16	Partitioning network area into cells and sub-cells.	118
17	Allocation of SF for 4000 nodes: (a) Cell-based and (b) ADR-LoRaWAN.	119
18	Dividing cells into sectors.	122
19	SBTS-LoRa frame structure of a cell partitioned into four sub-cells.	123
20	Polar coordinates of node n	124
21	Flowchart diagram of SBTS-LoRa protocol	125
22	Finding optimal p value	127
23	Scalability limitation of EXPLoRa-AT algorithm.	131
24	Example of the implemented Bit Rate Ratio Algorithm.	132
25	Probability of collision.	133
26	The end-to-end delay.	135
27	The network throughput.	136
28	The packet Error Rate (PER).	137
29	The energy consumption per bit (EpB)	139
30	Dividing the field into overlapping coronas with variable SFs.	143

31	Equal Radius-Based Corona (ERBC) mode example of Corona C_i with radius R_i	145
32	Equal Area-Based Corona (EABC) mode example of Corona C_i with radius R_i	148
33	Equal Area-Based Sector (EABS) mode example of Corona C_i with radius R_i	149
34	Time frames per channel CF.	151
35	Comparison between the frame sizes of ATS-LoRa protocol in ERBC, EABC, and EABS modes and SBTS-LoRa protocol [4].	152
36	Probability of collisions of ATS-LoRa compared to SBTS-LoRa and ADR-LoRaWAN.	155
37	The end-to-end delay of ATS-LoRa protocol.	156
38	Comparison between SF distribution in SBTS-LoRa and ATS-LoRa protocols.	157
39	Network throughput of ATS-LoRa protocol.	158
40	Energy consumption of ATS-LoRa protocol.	159

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor, Dr. Fatma Bouabdallah, for her precious support before and during my PhD studies, for her valuable guidance, availability, and encouragement throughout the entire process. I am profoundly grateful for the immeasurable contributions she made to my development.

I would like to express my deepest gratitude to my director of studies, Professor Ahmed Al-Dubai, for his guidance and advice that carried me through all the stages of my PhD research. Professor Ahmed was always supportive and cooperative, which made my PhD journey go smoothly. I would like to extend my sincere thanks to Dr. Baraq Ghaleb for his valuable comments toward improving my work. I would also like to thank Professor Amir and Dr. Zakwan for their valuable support during the RD6 meetings. I would also like to thank my committee members for letting my defense be an enjoyable moment and for their brilliant comments and suggestions.

Finally, I would like to express my deepest gratitude to my husband Eng. Hani Alahmadi, my daughter Noor, my parents, and my brothers and sisters. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

LIST OF PUBLICATIONS

The following is a list of publications made while working on this thesis.

Peer-reviewed Journal Publications:

1. H. Alahmadi, F. Bouabdallah, and A. Al-Dubai, "A novel time-slotted LoRa MAC protocol for scalable IoT networks", in *Future Generation Computer Systems*, Vol. 134, pp. 287-302, 2022, <https://doi.org/10.1016/j.future.2022.04.003>.
2. H. Alahmadi, F. Bouabdallah, A. Al-Dubai, and B. Ghaleb, "A novel Autonomous Adaptive Frame Size for Time-Slotted LoRa MAC Protocol", in *IEEE Transactions on Industrial Informatics* (accepted).
3. H. Alahmadi, F. Bouabdallah, A. Al-Dubai, and B. Ghaleb, "Efficient and Reliable LoRaWAN MAC and Data Link Protocols: A Survey and Future Prospects" (submitted)

Peer-reviewed Conference Publications:

1. H. Alahmadi, F. Bouabdallah and A. Al-Dubai, "A New Annulus-based Distribution Algorithm for Scalable IoT-driven LoRa Networks," *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500407.
2. H. Alahmadi, F. Bouabdallah, A. Al-Dubai and B. Ghaleb, "A Novel Autonomous Time-Slotted LoRa MAC Protocol with Adaptive Frame Sizes," *2023 International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, Morocco, 2023, pp. 917-922, doi: 10.1109/IWCMC58020.2023.10182638.
3. H. Alahmadi, F. Bouabdallah, B. Ghaleb and A. Al-Dubai, "Sensitivity-Aware Configurations for High Packet Generation Rate LoRa Networks," *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, London, United Kingdom, 2021, pp. 240-246, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00048.

1 Introduction

The massive progress in communication technologies has given birth to a new communication paradigm referred to as the Internet of Things (IoT) which has become an integral part of our daily lives, connecting various devices to the internet and resulting in unprecedented levels of connectivity across the world. At its core, IoT can be defined as a system of smart things or objects that can be uniquely identified and provided with the ability to exchange data autonomously over a network. At the heart of IoT is a collection of interconnected embedded sensor/actuator devices with the capability of communicating with each other facilitating the deployment of a wide range of applications such as smart homes, smart cities, and safety/security applications. Indeed, several IoT applications have stringent requirements including cost effectiveness, low power consumption, and wide coverage communication. These requirements are hard to achieve with short-range wireless technologies (e.g., Bluetooth and Zigbee) or power-hungry cellular technologies (e.g., 4G and 5G). Such unique requirements have pushed the efforts towards pioneering new networking technologies that can facilitate wireless communication over long distances while maintaining low-power consumption profiles, collectively referred to as Low Power Wide Area Networks (LPWANs).

Many LPWA technologies have been invented, which can be divided into LPWA technologies that either operate on Cellular networks or Wireless network. LTE-M [5] and NB-IoT [6] are examples of LPWA technologies for cellular networks, while SigFox [7] and LoRa [8] are examples of LPWA technologies that work on the free unlicensed band [9].

Among LPWA technologies for the unlicensed band, LoRa is taking the lead in the market. LoRa, which stands for Long Range, has attracted both the industrial and the research communities. The main attractive features of LoRa are the low-cost deployment due to the use of the unlicensed ISM band and has low-cost Commercial Off-The Shelf (COTS) devices compared to other LPWA devices that operate in the licensed band [10], the wide coverage due to the proprietary physical layer modulation, and the easy of deployment as the network operator can deploy and manage the IoT devices without a third party. LoRa has an estimated coverage range of up to 5 km in urban areas and up to 15 km in rural areas. The data transmission rate ranges from 0.3 to 37.5 Kbps. The estimated battery lifetime of LoRa devices is up to 10 years [10].

LoRa is a radio communication technique based on the proprietary Chirp Spread Spectrum (CSS) modulation. CSS encodes information on radio waves using chirp pulses making it robust and resilient against noise and

interference. LoRaWAN has been built on top of LoRa modulation to act as its Medium Access Control (MAC) protocol and to provide networking capabilities (i.e., define the network's architecture and manage the network). LoRaWAN organizes its network in a star topology, similar to cellular networks, where end nodes send packets to gateways with a maximum payload of 250 bytes. Gateways relay received packets to the LoRaWAN network server, which manages the entire network.

Different transmission parameters are provided by LoRa physical layer, which highly affect the performance of the overall network. These transmission parameters include different supported BandWidths (BW), different Carrier Frequency (CF) or channels, a number of Spreading Factors (SF), a number of Transmission Powers (TX), and various Code Rates (CR). The BW is typically 125 kHz, 250 kHz, or 500 kHz with the 125 kHz being the most commonly used BW especially for uplink transmissions from nodes to the gateway. The number of available channels depends mainly on the region of node deployment. For example, the European band, which is the one assumed in this study, has eight uplink channels. The Spreading Factors (SF) is a configurable radio parameter ranging between 7 and 12, that defines a payload number of bits that are encoded in a given chirp signal. The lower the SF value, the higher the data rate leading to lower energy consumption and transmission time, however, over shorter distances. On the other hand, transmitting packets with higher SFs allows the signal to travel for longer distances, and hence have longer transmission ranges, as higher SFs have higher sensitivity levels at the gateway. As a consequence, farther nodes would need to use higher SFs to increase their Packet Delivery Rates (PDRs). Furthermore, SFs are orthogonal [11] which means that if two or more simultaneous transmissions are made on the same Channel Frequency (CF) but encoded with different SFs, those transmissions will be successfully received at the gateway providing that the difference between their received power does not exceed a given threshold. Accordingly, each channel in LoRa could accommodate a maximum of six simultaneous transmissions corresponding to the six supported SFs and efficient use of SFs could greatly enhance the network capacity. Transmission powers typically ranges from 2 dBm to 20 dBm, and they greatly depend on the hardware. Coding rates (CR) of the Forward Error Correction (FEC) mechanism are used by the Physical layer and they support the following values $4/5$, $4/6$, $4/7$, and $4/8$.

Nevertheless, LoRa networks have their own limitations and challenges that prevent them from being scalable especially in large-scale deployments. The first limitation is the adopted medium access approach. Indeed, LoRa nodes use the ALOHA medium access mechanism to access the shared medium. Hence, whenever a node generates a packet, it will immediately proceed

with the transmission without any carrier sensing or time regulation mechanisms. Using such a mechanism maintains the power efficiency of the node's transceiver as the ALOHA protocol is simple and does not waste energy in listening to the medium or synchronizing the nodes with other network parties. Nevertheless, the scalability of LoRa networks is constrained by ALOHA channel access, particularly as the number of connected nodes rises. In fact, the maximum achievable channel capacity using ALOHA is only 18% of the total channel capacity as reported in [12].

Another challenge in LoRa networks that affects the network scalability is the limited Duty Cycle (DC) of nodes and gateways. As known, LoRa uses the unlicensed sub-GHz ISM band and in order to regulate the access between the devices of this band, Duty Cycle (DC) is imposed. DC is the fraction of time a device is allowed to transmit in a specific band during a set timeframe. For example, in Europe band, the duty cycle for sensor nodes and gateways for uplink and downlink traffic is limited to 1% for all bands except one band that is used for downlink traffic, which allows a duty cycle of 10%. This implies that if the ToA_i is the Time on Air for node i , then that node must wait at least $99 \times ToA_i$ before transmitting again using that band. Hence, selecting the optimal SF for nodes such that their packets are successfully received at the gateway with minimum ToA is crucial. In fact, the DC limit that is imposed on the downlink traffic from gateways to nodes is more critical than the uplink traffic from nodes to gateways, as LoRa targeted large-scale dense networks with sporadic network traffic [13].

Another challenge is configuring nodes with the appropriate transmission parameters to maximize the network scalability. As explained earlier, LoRa physical layer supports different transmission parameters that greatly affect the performance of uplink communications. One way to control these parameters is by activating the Adaptive Data Rate (ADR) algorithm of LoRaWAN protocol. The ADR algorithm of LoRaWAN tends to enhance the network performance by adapting both the node's data rates, which is mainly affected by the node's spreading factor, and the transmission power levels. To do that, during the initialization phase, nodes select randomly a spreading factor and a transmission power in order to send their "join" requests and packets to the server. When a node sends multiple packets without receiving any acknowledgment packet from the server, the node supposes that their transmissions didn't reach the server. Accordingly, the node will gradually increase its spreading factor and/or their transmission power. Once the server receives packets from nodes, it will record the Received Signal Strength Indicator (RSSI) for each received packet. After the server receives a specific number of packets from a given node, it will estimate the most suitable SF and transmission power level, based on the recorded RSSIs for

that node and transmit them in a downlink communication [14]. Although ADR-LoRaWAN has efficient performance in small- to medium-scale networks, its efficiency dramatically decreases in large-scale dense networks, as demonstrated in Section 5.5. The main reason is the centralized approach of the ADR algorithm, where the server needs to update each node independently by a dedicated downlink communication and this update process could be obstructed due to duty cycle limitation. Consequently, when nodes did not receive downlink communications from the server in their expected periods, it will suppose that their transmissions are not reachable. Hence, it will increase its spreading factor and/or transmission power. Eventually, most of end nodes will end up with the highest spreading factor with the longest time on air, which will further increase the probability of collisions and hence reduce the overall network throughput.

1.1 Problem Statement and Research Questions

According to the aforementioned challenges, different research studies have been proposed to address these challenges, as described in Chapter 3. In fact, most of these studies proposed centralized Time-Division Multiple Access (TDMA) approaches, where nodes need extensive downlink traffic from the server to synchronize and disseminate schedules among nodes [15] [16] [17] [18] [19] [20]. In other words, before a node proceeds with its transmission, it must receive some control packets from the server to determine its suitable transmission parameters. In the worst-case scenario, the gateway could reach its duty cycle limit before updating node's transmission parameters. Furthermore, the frame sizes of such TDMA approaches are fixed to a specific number of timeslots or duration ignoring the dynamicity nature of LoRa networks represented in different transmission time for different SFs [21] [22] [23]. Besides the centralized approaches, most of the proposed studies assumes small-scale networks, where each node is eligible to select any SF ignoring hence the various transmission ranges for each SF [24] [25] [26] [27]. Hence, more realistic large-scale dense environments are needed to evaluate the proposed solutions. Furthermore, the proposed solutions address only one challenge of LoRa networks. In other words, the related work studies either deal with the ALOHA channel access by proposing centralized TDMA approaches ignoring hence the limited DC or propose some centralized distribution algorithms of transmission parameters for nodes ignoring the ALOHA random access and the limited DC. To the best of our knowledge, no comprehensive solution was proposed to deal with all LoRa challenges in a simple and applicable manner. Based on these research gaps, the following research questions are to be addressed:

-
- How the configurable transmission parameters can be distributed to maximize the network scalability such that their energy efficiency and server's DC are not compromised?
 - How a Time-Division Multiple Access (TDMA) mechanism can be conceived for LoRa networks considering the special features of the network such as the dynamicity of the network and the limited DC?
 - How does any proposed protocol perform in large-scale dense LoRa networks? and what is its performance compared to the ADR-LoRaWAN protocol?

1.2 Research Aim and Objectives

The main aim of this thesis is to address the main challenges of LoRa networks and conceive comprehensive solutions considering all these challenges such that the network scalability and the energy efficiency are maximized. In fact, MAC protocols, which control the radio status of sensors as well as the access to the transmission medium, play a critical role in managing the battery lifetime of a node, the network scalability, and the variety of applications that are supported by the network. Our research work includes a number of objectives that will help us achieve our ultimate aim, which are described below.

Objective 1: Conduct a thorough review of the technical aspects of LoRa networks.

Objective 2: Carry on a thorough review of the related work to identify the main research gaps.

Objective 3: Develop a distribution algorithm of LoRa transmission parameters that maximizes the network scalability considering the main limitations of LoRa networks.

Objective 4: Replace the ALOHA random channel Access with a Time-Division Multiple Access (TDMA) mechanism such that the network scalability is maximized considering the limited duty cycle of LoRa gateways.

Objective 5: Develop a Time-Division Multiple Access (TDMA) protocol with dynamic time frame sizes and slot duration.

Objective 6: Evaluate the proposed protocols in realistic environment settings, where large-scale and dense networks are considered.

1.3 Thesis Structure

The remainder of this thesis is organized into six chapters as follows:

Chapter 2: Overview of Long Range (LoRa) Networks This chapter describes the main characteristics of LoRa networks by providing a thorough overview of the technical aspects of both the physical and the MAC and data link layers. It also describes how these technical aspects affect the overall performance of the network. Furthermore, a comparison between LoRa and other common Low Power Wide Area (LPWA) networks is provided. The goal of this comparison is to show the technological context of LoRa among other LPWA networks. Moreover, this chapter highlights some large-scale real-deployment applications for LoRa networks to show its effectiveness in the industrial and commercial fields.

Chapter 3: A Survey on efficient and reliable LoRaWAN MAC and data link protocols This chapter reviews the literature that aimed at improving the capacity and scalability of LoRa networks specifically at the MAC (Medium Access Control) and data link layers. Unlike other surveys, this study focuses on these layers because they play a pivotal role in managing collision rates, which significantly impact network scalability. The chapter suggests a comprehensive review of the literature, organizing it based on key limitations that could hinder the network's ability to meet its performance objectives, including scalability, Packet Delivery Ratio (PDR), and energy efficiency.

Chapter 4: Sensitivity-Aware LoRa Configuration (SAL) algorithm for large-scale LoRa networks In this chapter, we introduce Sensitivity-Aware LoRa Configuration (SAL), a new algorithm for efficient autonomous and distributed selection of LoRa physical layer transmission parameters. The aim is to address the limitations of the currently adopted MAC algorithm in LoRaWAN networks (i.e., Adaptive Data Rate - ADR). The selection of the transmission parameters in ADR is done randomly by the gateway, an approach that may result in the gateway reaching its Duty Cycle Limit, consequently hindering it from sending the configuration information to the end points under large-scale networks negatively affecting the network performance. Unlike ADR, SAL uses a decentralized approach to

select node's transmission parameters without any need for gateway's control packets and it only considers a combination of parameters that is guaranteed to be received successfully by the gateway. The performance of the proposed algorithm is validated through extensive simulations under different scenarios and operation conditions. In particular, SAL is compared to LoRaWAN configuration algorithm in terms of Packet Error Rate (PER), capture effect probability, collision probability, end-to-end delay, Packet Delivery Ratio (PDR), throughput and energy consumption showing promising results in this context.

Chapter 5: Sector-Based Time-Slotted (SBTS-LoRa) MAC Protocol for LoRa Networks This chapter presents the Sector-Based Time Slotted SBTS-LoRa MAC protocol that allows nodes to determine their transmission parameters autonomously based on their location relative to the gateway. Furthermore, it regulates the access to the transmission medium by autonomously assigning timeslots to nodes without any downlink transmissions from the gateway. Based on nodes distance to the gateway, they can individually determine their transmission parameters and timeslots. We leverage some tools from the geometry of circles to determine the timeslots of nodes. SBTS-LoRa is targeting large scale networks. The main objective of the proposed protocol is to maximize the network scalability by increasing the network throughput without compromising the energy efficiency. Simulation results show that the proposed protocol significantly enhances the scalability and outperforms its counterparts by maximizing throughput without compromising the energy efficiency. Specifically, the average throughput for dense networks was enhanced 14 times compared to the Adaptive Data Rate ADR-LoRaWAN.

Chapter 6: Autonomous Time-Slotted MAC protocol (ATS-LoRa) with adaptive frame sizes This chapter proposes the Time-Slotted LoRa (ATS-LoRa), a Time-Slotted MAC protocol that allows LoRaWAN nodes to autonomously determine their optimal transmission parameters without extensive downlink transmissions from the gateway. ATS-LoRa utilizes the location information of the nodes and their gateway in a novel way to allow them to determine their appropriate transmission parameters such as the spreading factor (SF), the channel frequency (CF), and the slot ID (SID). Moreover, in ATS-LoRa protocol, any channel has six different frame sizes corresponding to the available six SFs. The frame size has lower and upper boundaries taking into account nodes duty cycles. Furthermore, the frame size is dynamic, and it mainly depends on the node density on a given

channel with a given SF. Simulation results show that the ATS-LoRa protocol achieves a frame size that is only 150 slots with a very large number of connected nodes (4000 nodes). This is a very important feature that highly reduces the end-to-end delay and hence the network throughput is enhanced. ATS-LoRa performance is evaluated through extensive simulations under different operating conditions showing an average throughput of around 47 times better than ADR-LoRaWAN.

Chapter 7: Conclusion This chapter summarizes the primary gaps addressed in this research and clarifies the main approaches employed to resolve such gaps. It also expands on the primary results and highlights critical lessons gained, as well as future research potential and directions.

2 Overview of Long Range (LoRa) Networks

Long Range (LoRa) networks have emerged as a vital solution catering to applications requiring coverage over relatively long distances in the world of Internet of Things (IoT), where connectivity and efficient data transmission are paramount. LoRa's utilization of the unlicensed Industrial, Scientific, and Medical (ISM) band not only underscores its cost effectiveness but also positions it favourably against licensed technologies in terms of deployment cost. According to that, LoRa has been involved in different industrial applications for different IoT scenarios.

LoRa technology consists of two network layers: the physical layer and the MAC and data link layer. LoRaWAN, which is the MAC protocol used by LoRa networks, adopts a star topology, where individual LoRa devices communicate directly with one or more gateways. These gateways act as intermediaries that receive data from devices and forward it to a central network server. The network server then processes and manages the data, enabling bi-directional communication between devices and applications. Fig.1 demonstrates the network architecture of LoRa networks. LoRaWAN supports two communication modes: the uplink and the downlink modes. In the uplink mode, the data is transmitted from end nodes to gateways, while in the downlink mode, the commands or responses are transmitted from the server to the end nodes through gateways.

This chapter first introduces most of the common Low Power Wide Area (LPWA) Networks. Moreover, a comparison between LoRa and other common Low Power Wide Area (LPWA) networks is provided. The goal of this comparison is to show the technological context of LoRa among other LPWA networks. Then, the chapter describes in detail the main characteristics of LoRa networks by providing an overview of the technical aspects of both the physical and the link layers. It also describes how these technical aspects affect the overall performance of the network.

2.1 Common Long Range Technologies

Most LPWA networks use the Sub 1 GHz band, except Ingenu RPMA, because it is less congested than 2.4 GHz band that is used by different protocols such as WiFi, Bluetooth, and ZigBee. Hence, the probability of interference between different technologies is reduced in the sub Ghz band. Modulation techniques play an important role in achieving long range communication. Furthermore, when a communication signal with a lower frequency encoun-

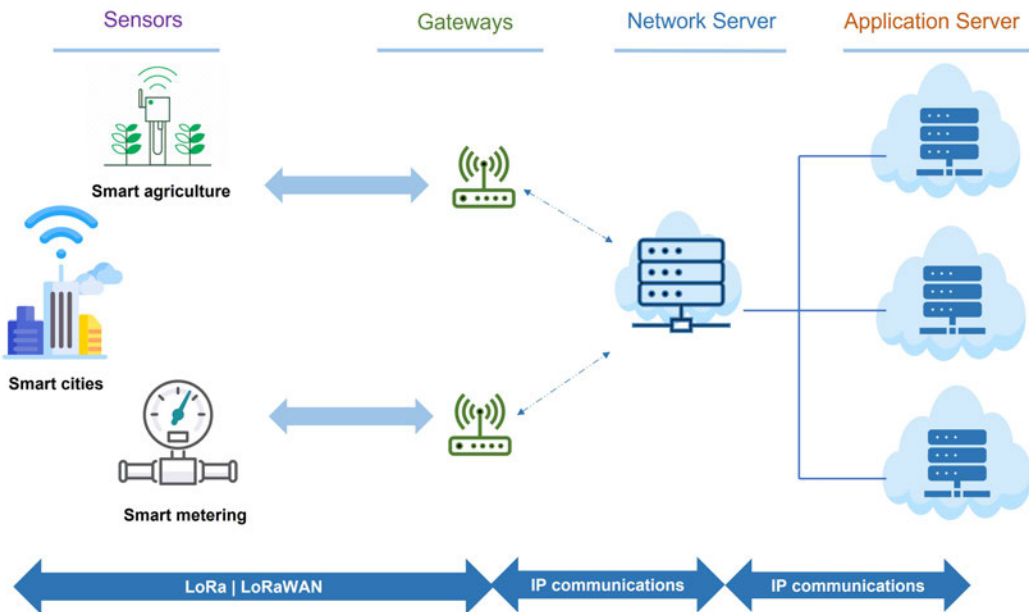


Figure 1: The network architecture of LoRa networks.

ters an obstacle, it experiences multiple path fading and less attenuation. This increases the possibility of correct signal reception. Low power consumption is obtained by adopting simple star network topology, where sensor nodes are connected directly to a base station instead of mesh topology that is adopted in wireless networks. Furthermore, the use of lightweight MAC protocols such as ALOHA or TDMA will not consume much energy compared with the MAC protocols used in wireless and cellular networks. Hence, complex operations are offloaded to the base station to make sensor's hardware simple. Therefore, LPWA technologies have low cost sensors [28]. The following sections provide a brief overview about the most common LPWA technologies, namely SigFox [7], Ingenu RPMA [29], and NB-IoT [6].

2.1.1 Sigfox

Sigfox is an Ultra-NarrowBand (UNB) technology that operates on the unlicensed sub-1 GHz frequencies. It divides the sub-bands (868.180 MHz-868.220 MHz) into 400 distinct 100 Hz sub-bands, forty of which are reserved [28]. Sigfox Network Operators (SNOs) deploy proprietary base stations outfitted with cognitive software-defined radios and connect them to backend servers via an IP network. Sensor nodes communicate with these base stations through the use of Binary Phase Shift Keying (BPSK) modulation [30].

SigFox uses UNB to efficiently utilize bandwidth while experiencing very low noise levels, resulting in high receiver sensitivity, ultra-low power consumption, and an inexpensive antenna design. All these advantages come at the cost of a maximum throughput of only 100 bps, which is the lowest offered data rate among LPWAN technologies as demonstrated in Fig.2. To comply with the duty cycle regulations imposed on the license-free spectrum, the number and size of uplink messages are limited to 140 12-byte messages per day. For downlink transmission, a maximum of 4 8-byte data packets transmitted from base station to nodes. According to that, packet acknowledgments are not supported through this network. Alternatively, to increase the possibility of successful reception at the base stations, a single message is transmitted multiple times (3 by default) in different channels and different time offsets [30]. Users are expected to purchase sensors and subscriptions from network providers in order to connect to regional Sigfox networks without deploying gateways. Sigfox receives and processes sensed data on a cloud server before sending it to a backend server [28]. Hence, deploying private networks using SigFox is not supported.

2.1.2 INGENU RPMA

In contrast to LoRa and SigFox, Ingenu RPMA operates within the 2.4 GHz spectrum and takes advantage of more relaxed regulations applicable in various regions. Notably, the 2.4 GHz band does not impose any Duty Cycle (DC) restrictions, unlike the ISM band utilized by SigFox and LoRa consequently enabling higher network throughput and capacity. Ingenu uses a proprietary MAC protocol named Random Phase Multiple Access (RPMA), which is based on the Code Division Multiple Access CDMA scheme. However, unlike the traditional CDMA, RPMA has a longer slot duration allowing multiple transmitters to share the same time slot while distinguishing their transmissions through introducing random offsets at transmissions start times. Hence, possible overlapping between the transmitters that use the same timeslot is reduced. Base stations spread the signals for individual end devices and then broadcast them using CDMA for downlink communication [30]. Furthermore, by relaying channel conditions within uplink messages, base stations can optimize downlink data rate, capacity, and energy usage [28].

2.1.3 NB-IoT

NB-IoT employs Quadrature Phase Shift Keying (QPSK) modulation and uses the same licensed frequency bands as LTE. Despite the fact that NB-IoT

is integrated into the LTE standard, it simplifies many LTE features to reduce costs and energy consumption, such as exploiting a different bandwidth, using a different cell search process, and adapting a modified random-access technique. Data rate, latency, and spectrum efficiency features result in increased coverage and lower power consumption. Furthermore, by utilizing a narrower band, the price of the NB-IoT chip is reduced, allowing it to be widely deployed [28].

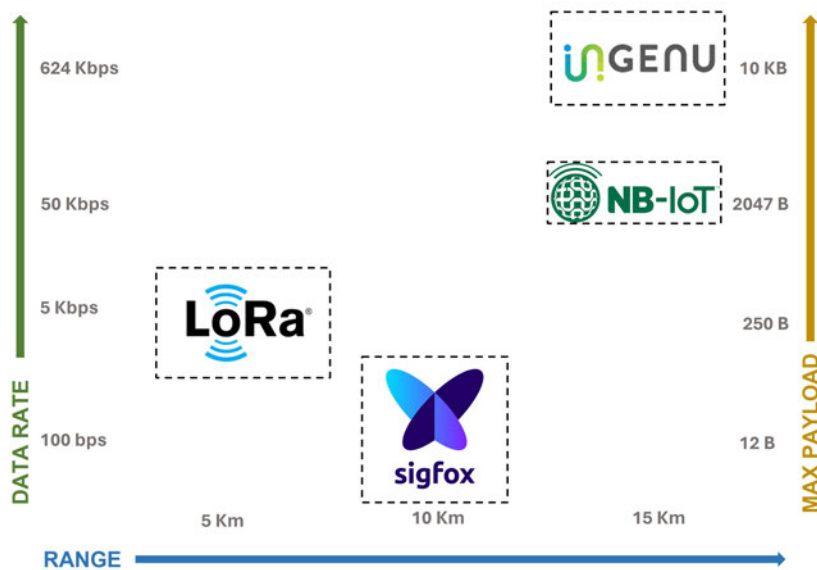


Figure 2: Comparison between Sigfox, LoRa, RPMA, and NB-IoT technologies in terms of the data rate, the maximum range for urban areas, and the maximum payload.

2.2 LoRa Technological Context

This section reviews the main features and characteristics of LoRa compared to other LPWAN technologies. Marini et al. in [31] conduct a simulation-based comparison analysis between the performance of LoRa network and NB-IoT to provide guidance of the most appropriate application domains for each one of them. They emphasized on the fact that although NB-IoT is more reliable and has higher throughput, since it uses more powerful modulation and it has no DC regulations, it suffers from higher energy consumption. However, some of their experiment results showed that the network parameters configuration, such as the use of Adaptive Data Rate (ADR)

with LoRaWAN, could allow a technology to outperform its counterpart. According to their simulation results, LoRaWAN had lower throughput than NB-IoT for low-density networks (< 500) due to the limited DC, especially with high SFs. However, both technologies provided similar performance for dense networks (> 1000). Furthermore, LoRaWAN had much lower and stable energy consumption than NB-IoT regardless of the network size or scale. Ikpehai et al. in [32] carried out another simulation-based comparison between different LPWA technologies. Accordingly, LoRa provided the longest coverage area compared to SigFox and RPMA in both urban and rural environments thanks to its high resistance to obstacles. Furthermore, according to their simulation results, LoRa provided the longest battery life among other LPWA technologies.

From previous discussion we can deduce that careful configuration of LoRa physical layer transmission parameters such as the SF, the transmitted power, the coding rate, and the physical channel may result in better network performance compared to other LPWA networks. In other words, although the theoretical features that are announced about other LPWA technologies may be better than LoRa, with wise selection of LoRa transmission parameters, some simulation studies demonstrated the counterpart [31] [32]. This emphasizes the importance of transmission parameter configuration that will be discussed in Section 3.3. Table 1 shows the main technical features of Sigfox, LoRa, Ingenu, and NB-IoT LPWA technologies.

Table 1: The main technical features of Sigfox, LoRa, Ingenu, and NB-IoT LPWA technologies.

Protocol	Sigfox	LoRa	Ingenu RPMA	NB-IoT
Frequency band	Sub 1-GHz ISM: 868 MHz EU, 902 MHz US	Sub-GHz ISM EU (433, 868 MHz), US (915 MHz), AS (430 MHz)	2.4 GHz	Licensed 700-900 MHz
Modulation	DBPSK (UL), GFSK (DL)	LoRa CSS	RPMA/ DSSS	QPSK

MAC	ALOHA-based	ALOHA-based	RPMA (CDMA-like)	SC-FDMA (UL), OFDMA (DL)
Data rate	100 bps	0.3 -5 kbps	624 kbps UL, 156 kbps DL	50 kbps (UL), 60 kbps (DL)
Range	3-10 km (urban), 20-50 km (rural)	2-5 Km (urban), 10-15 (rural)	15 km	15 km
Bandwidth	100 Hz	125 KHz	1 MHz	180 KHz
Max allowed payload	12 B UL, 8 B DL	250 B	Max 10 kB	2047 B
Tx power	14 (UL), 27 (DL)	14 (UL), 27 (DL)	21 (UL), 27 (DL)	23 dBm
Multiple channels or orthogonal transmissions	360 channels	10 in EUR, 64-8 (UL) and 8 (DL) in US, plus multiple SFs	40 1 MHz channels, up to 1200 signals per channel	Multiple channels
Nodes per GW	50k/cell	40k/cell	500k/cell	200k/cell
DL communication	Very limited	Yes Class dependent	Yes	Yes
FEC	No	Yes	Yes	Yes
Localization support	No	Yes	Need GPS	Need GPS

2.3 LoRa Physical Layer

LoRa physical layer is a critical component of the LoRaWAN (Long Range Wide Area Network) technology, which enables long-range, low-power wireless communication for Internet of Things (IoT) devices. LoRa physical layer is designed to achieve remarkable communication ranges while operating within stringent power constraints. LoRa uses a chirp spread spectrum modulation technique, where data is encoded into chirps (frequency-swept waveforms). This modulation method enables robust communication in chal-

challenging environments, where traditional modulation schemes might fail due to noise and interference. The proprietary modulation allows for longer transmission ranges by multiplying the original data signal with a pre-defined bit pattern, named spreading factor, that has higher data rate. In other words, the overall signal's energy is now spread across a larger frequency range, allowing the receiver to distinguish a signal with a lower signal-to-noise ratio (SNR). The spreading factor determines how the data signal is spread across time and frequency domains during transmission and controls the rate at which the chirp signal changes its frequency and the duration of each chirp. In other words, a higher spreading factor will result in a slower change in frequency (wider chirps) and longer chirp durations. LoRa supports spreading factors from 7 to 12, with higher spreading factors providing longer range but lower data rates and vice versa. Fig.3 demonstrates the effect of spreading factors on the data rate and the transmission range.

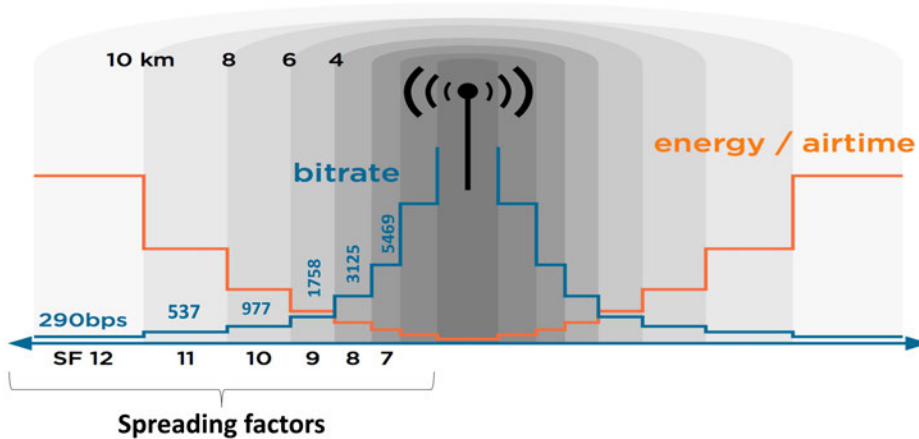


Figure 3: The effect of spreading factors on the data rate and transmission range [2].

In fact, spreading factors have a direct effect on both the data rate and the transmission range. Regarding the effect on the transmission data rate, spreading factors manipulate the transmission data rate by spreading SF bits of data over 2^{SF} symbols. Hence, increasing the SF will double the number of symbols that are transmitted per SF bits of data. As a result, longer Time on Air (ToA) is needed with higher SFs. Consequently, the bit rate R_b is calculated as follows:

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \quad (1)$$

where BW is the channel bandwidth in Hz, SF is the spreading factor of

the signal, and CR is the coding rate that will be explained later. Regarding the effect of the spreading factor on the transmission range, the higher the spreading factor, the longer the distance the signal can travel while still being received without error by the receiver. Hence, each spreading factor has different transmission range resulting from the variety of the sensitivity levels at the receiver side due to the variety of the SNR levels per SF. Table 2 shows the values of the data rates, the SNR levels and their corresponding sensitivity levels, and the Time on Air (ToA) of each SF assuming the bandwidth is 125 KHz and the payload equals 64 bytes [14]. As shown in the table, increasing the SF by one result in decreasing the SNR by almost 2, which will increase the power level of that signal by 5 db [33]. The negative values of SNR indicate the ability of the gateway to decode signals below its noise floor level. The sensitivity level of LoRa receiver is calculated as follows

$$S = -174 + 10 \log(BW) + NF + SNR_{limit} \quad (2)$$

Where -174 dBm is the thermal noise density, which is mostly influenced by the receiver temperature and NF mostly equals 6, which represents the noise figure allowed at the gateway side.

Table 2: SF estimated values.

SF	Data rate (b/s)	SNR_{limit} (db)	Sensitivity (dBm)	Time on Air (ms)
7	5469	-7.5	-123	41
8	3125	-10	-126	72
9	1758	-12.5	-129	144
10	977	-15	-132	288
11	537	-17.5	-134	577
12	293	-20	-137	991

Spreading factors in LoRa networks are considered orthogonal which means they are independent and non-interfering with each other. Indeed, spreading factors in LoRa are designed in a way that the chirp waveforms generated by different spreading factors do not overlap significantly in the time or frequency domain, hence, two or more signals decoded with different SFs and transmitted simultaneously on the same channel will be successfully decoded by the gateway. In fact, signals decoded with different SFs look like noise to each other [34]. Bandwidth (BW) is another important transmission

parameter in LoRa networks that significantly influences communication performance, data rate, and the trade-off between range and robustness. Wider bandwidth allows more data to be transmitted within a given time interval, resulting in a higher data rate. Conversely, a narrower bandwidth decreases the data rate but can improve communication range and robustness. Bandwidth (BW) transmission parameter is also configurable to be one of these values 125, 250, or 500 KHz. As known, increasing the bandwidth increase the data rate of the transmission. In fact, as shown in Eq.2, the increase of the bandwidth actually decreases the sensitivity level at the gateway by almost 3 dB. As a result, the data rate is increased on the cost of decreasing the transmission range [3] [35]. Fig.4 demonstrates the effects of the spreading factor (SF) of a signal on the symbol duration. As shown in the figure, different signals encoded with different SFs and transmitted within the same bandwidth (250KHz in the figure) will have different symbol duration. In other words, increasing the SF of a given signal will increase the symbol duration and hence decrease the symbol rate. On the other hand, increasing the bandwidth of a signal will increase the number of encoded bits per signal per specific period of time. Hence, the transmission bit rate will increase.

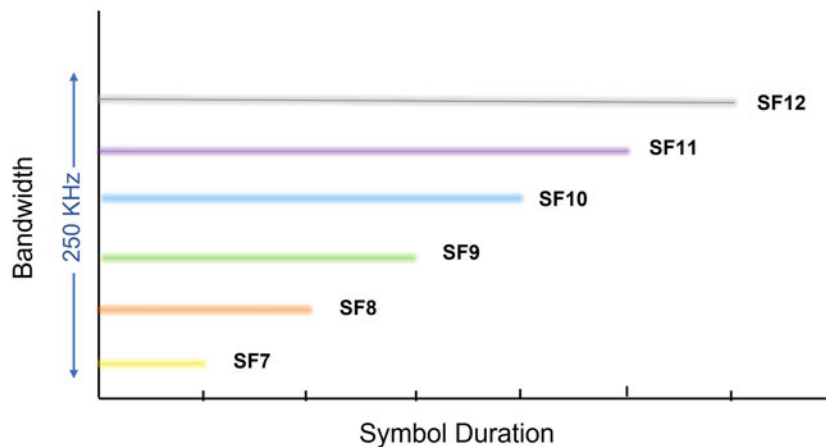


Figure 4: The relationship between the spreading factor and symbol duration [3].

Different transmission power (TX) levels are considered in LoRa networks, which range from 2 dBm to 14 dBm. The common TX values used by LoRaWAN are 2, 5, 8, 11, and 14 dBm. Increasing the transmission power levels increases the transmission range on the cost of increasing the energy

consumption of source-constrained end-node devices. Hence, wise selection of transmission power levels is vital in controlling the energy efficiency of nodes.

Another transmission parameter that affects the bit rate and the quality of the transmission is the Coding Rate (CR). The Coding Rate refers to the ratio of the number of error correction bits added to the original data bits during transmission. These error correction bits help the receiver to detect and correct errors that might occur due to noise or interference in the communication channel. The CR represents the ratio of bits in a data packet that actually carry the real data. Four levels of CR are supported in LoRa, which are $4/5$, $4/6$, $4/7$, and $4/8$. The ratio $4/6$ means for every 4 bits of data six bits are transmitted with 2 bits as redundant bits. Higher CRs offer better error correction against burst interference but can result in longer transmission times that might impact ToA estimation accuracy. Lower CRs prioritize higher data rates but might be more susceptible to burst interference and can provide more accurate ToA estimates due to shorter transmission durations. This trade-off is demonstrated in Fig. 5, where the numbers 1, 2, 3, and 4 on the x-axis refer to the CRs $4/5$, $4/6$, $4/7$, $4/8$ respectively. Fig.5 shows the ToA and the BR with SF=10 and BW=125 KHz. In fact, increasing the CR by one will increase the ToA by almost 90 ms.

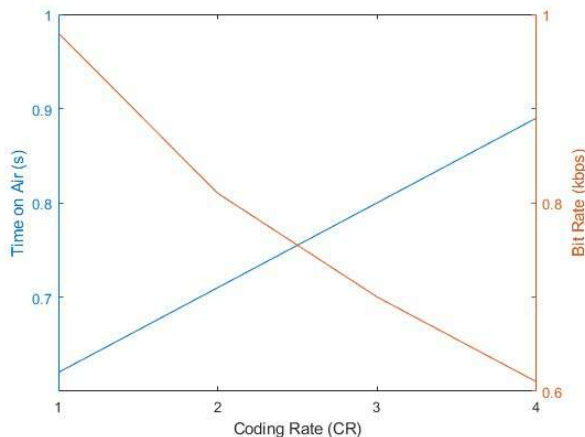


Figure 5: The ToA and the BR with SF=10 and BW=125 KHz

Regarding the impact of the Coding Rate (CR) on the Packet Delivery Ratio (PDR) of transmissions, Park et. al. in [36] performed a pilot experiment to investigate its effect. In their experiment, 20 sensor nodes are distributed around the gateway with a maximum distance of 50m in an in-

door environment. They fixed the SF to 7 and the TX to 14 dBm while they changed the SNR and the packet arrival rate, so there will be different link conditions. A general observation from the experiment is that the PDR is reduced by the SNR. Specifically, at low network traffic rates, the PDR of transmissions with low CRs (4/7 and 4/8) is more stable compared to the PDR of transmissions with high CRs (4/5 and 4/6). However, the state is the opposite with high traffic rates, where high collision rates are presented. In fact, low CRs, and hence longer packet lengths, increase the collision probability, and hence the PDR is badly affected. Accordingly, the effect of the dynamic or adaptive CR is only presented with low traffic rate networks. Moreover, this effect is similar to the one provided by adapting the SFs, however, only on a very small scale compared to the effect of the adaptive CRs. According to that, and since we are targeting large-scale dense networks with high traffic rates, in this research, we only consider the effect of adaptive SF with TX as they are the main factors affecting the SNR, which in turn has a direct effect on the collision rate and the network scalability.

Different channel frequencies are available on LoRa networks. In fact, the number of the available channels depends on the region of the ISM band. For example, In Europe band, there are at least 16 channels supported by EU868MHz. However, only three channels among them must be implemented in every LoRa end device as they are the default channels [37]. Although in LoRaWAN regional parameters document is stated that there are 16 channel frequencies supported in the Europe band, only eight channels are implemented in the ThingsNetwork [38].

Since LoRa uses the unlicensed ISM band, the channel access is regulated through the use of Duty Cycle (DC). Duty Cycle refers to the ratio of time a device is actively transmitting (on-air) to the total time within a certain period. It is usually expressed as a percentage or fraction. For example, a duty cycle of 1% means that a device can transmit for 1% of the total time and must remain idle for the remaining 99% of the time. In EU868, a duty cycle of 1% is imposed on the default band for both the uplink and downlink transmissions. This means that, if node X transmits on a band with a Time on Air ToA_X , that node must wait at least $99 \times ToA_X$ before transmitting again on the same band. Hence, increasing the transmission time will not only affect the energy efficiency of nodes, but it also affects the overall network throughput or capacity as the transmissions of nodes with longer transmission times are constrained with the DC. Hence, assigning optimal transmission parameters to nodes has an important impact on the overall network performance.

2.4 LoRaWAN MAC Layer

As described in the previous section, LoRa provides specifications related to the physical layer. LoRaWAN, on the other hand, provides specifications related to the Link and MAC-based functions. LoRaWAN

LoRaWAN defines three device classes with varying power consumption and latency characteristics named Class A, Class B, and Class C. All end nodes must implement at least the functionalities of Class A. Class A nodes initiate transmissions to the gateway by sending uplink packets as needed. Following their uplink transmissions, nodes open two receiving windows to potentially receive downlink data from the gateway. Hence, in this class, downlink transmissions are always possible after uplink transmissions from nodes. The delay between the receiving windows as well as their start time are determined by the end time of the uplink transmission. Fig.6 shows the timing of the receiving slots for nodes. *RX1* and *RX2* are the first and second receiving windows, respectively. *RECEIVE_DELAY1* and *RECEIVE_DELAY2* are the delays before the first and second receiving windows, respectively. In Class B, besides *RX1* and *RX2*, nodes have scheduled receiving windows called ping slots to receive beacons from the gateway. The goal of this class is to allow the gateway to send downlink transmissions at pre-scheduled timeslots instead of waiting for an uplink transmission from a node to transmit a downlink traffic in its own receiving windows. Class C devices open continuous receiving windows except when they are transmitting. Consequently, this class is better suited for devices equipped with sufficient power resources [37].

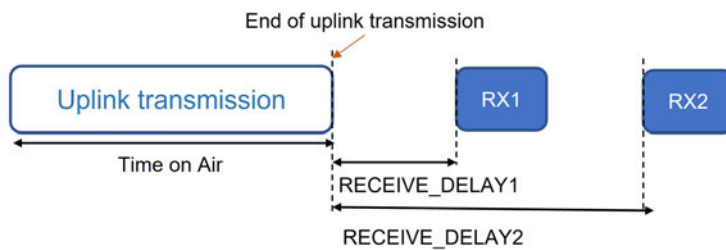


Figure 6: Receiving slots scheduling.

2.5 The Adaptive Data Rate (ADR) Algorithm of LoRaWAN Protocol

The Adaptive Data Rate (ADR) is an algorithm used by the network server to adjust the data rates or the spreading factors of end-devices. As explained earlier, nodes that are closer to the gateway may need smaller transmission ranges and hence they are recommended to use smaller SFs with higher data rates and lower sensitivity levels. On the other hand, nodes that are farther from the gateway are advised to use higher SFs with lower data rates to increase the receiver's sensitivity and hence increasing the transmission range. According to that, the ADR algorithm that runs on the server side allows to adjust the SF of nodes based on their received power, or link budget, at the gateway. The ADR network ends up with nodes that are closer to the gateway using smaller SFs and nodes that are farther from the gateway use higher SFs. In this case, the energy consumption due to the use of unnecessary higher SFs can be mitigated. Furthermore, the network capacity will be enhanced as the SFs are orthogonal and there will be diversity in the SFs among nodes based on their link budget.

When the ADR mode is enabled at the node side, it will wait for an ACK from the server for a pre-defined period. If no ACK is received in that period, it will assume that it lost the connectivity with the gateway, hence it will decrease its data rate, i.e., increase its SF. In dense networks, the server needs to acknowledge large number of ADR-enabled nodes, which is hard to achieve due to the limited DC. As a result, nodes may unnecessarily increase their SFs if no ACK was received from the server. The network will end up with a large portion of the nodes using the lowest data rate as they assume that they have lost the connectivity with the gateway. In this case, high collision rates, high energy consumption, and longer transmission times are expected in the network. Moreover, the ADR algorithm is suitable for static devices only not dynamic ones as the link budget in such devices is hard to predict due to the dynamic environment [14].

Another variant of the ADR algorithm named blind ADR that is used basically for mobile applications [39]. The main idea of blind ADR is to allow nodes to use a set of SFs instead of a single SF, that is selected basically based on node's link budget to the gateway. In [39], they assume a LoRaWAN plus GPS pet tracking application, where a LoRa node, a pet, uses one of the three SFs, SF7, SF10, and SF12 based on its location. SF7 is used when the pet is outdoor, SF10 is used when the pet is indoor, and SF12 is used when the pet is in deep door locations to ensure the connectivity with the gateway. Furthermore, the data periodicity is different based on the used SF. According to their implementation [39], a data packet is transmitted every

60 seconds with SF12, one packet is transmitted with SF10 every 30 seconds, and a data packet with SF7 is transmitted every 20 seconds. The rationale behind the variant of data periodicity is that when the pet is outdoor, more packets are transmitted to determine its location. However, when it is indoor, less packets needed to determine its location as it is safe indoor. According to that, less energy consumption is expected due to the variant of the used data rates and data periodicities.

2.6 LoRa Deployment for Industrial Applications

LoRa networks have seen widespread adoption across a range of applications, each serving distinct purposes and under different network scales. For example, a LoRa network has been deployed in Saudi Arabia to track the workers and vehicles in the Red Sea Giga Project [40]. The goal of this deployment is to enhance the security and ensure worker safety in the event of a medical emergency or prevent workers from becoming lost in remote locations. This deployment covers an extensive area of $35,000\text{km}^2$ and scale up to 39,000 end devices.

Different LoRa networks were deployed to enable smart city scenarios. In the province of Palermo, Italy a LoRa network was deployed to control more than 800 street lightning lamps [41]. The network covers 50km^2 to control 5000 lamps that send 480,000 messages per day. Another LoRa network was deployed with more than 150 thousand LoRa sensors to provide real-time water usage reading to allow customers to be notified about the out of the ordinary consumption size and potentially save costs [42]. A similar LoRa project was deployed in more than fifty cities in Spain to manage water metering with more than 300 thousand devices [43]. The most appealing LoRa features for such projects are the large scale and the easy deployment, the cost effectiveness, and the long-expected battery life. Another example of smart city's application was deployed in the United States to automate car parking [44]. The aim of this project is to implement an automated system for tracking and processing parking fees. Hence, around 1700 device were deployed around 18 gateways to replace the manual paying spots. A nationwide LoRa multi-purpose network was deployed across Czech Republic with more than 580 LoRa gateways [45]. This network comprises several IoT sensors including, for instance, metering and cameras sensors. They have tested the network in two different use cases, namely in water metering and waste management. Another general purpose LoRa network was deployed in Tampa Bay, Florida metro area with a network coverage around 1600 square miles that connect different kinds of sensors such as water meters, lightning controllers, asset trackers, air quality monitors, and trash fill monitors [46].

Regarding smart agriculture, a real-time irrigation control system was deployed in the farms across the South Island of New Zealand [47]. The main goal was to allow farmers to control irrigators, set and manage watering schedules and get data about their water use. The deployed system should have low network costs, low power requirements to last at least seven years in the field, and uplink and downlink message capability to control the device and set schedules to confirm activity. The current number of deployed devices is more than 4000 devices.

Different LoRa deployments are also used to manage buildings. For example, a LoRa network was deployed that scale in all Sweden to manage buildings against mold forming and water leakages [48]. By combining different sensor types such as temperature and humidity sensors and direct water leakage sensors, water leakages as well as mold forming can be discovered early. Hence, building maintenance costs are reduced. Another similar project deployed in Canada to monitor 60 commercial building temperature, humidity, carbon dioxide concentration, and people counting to reduce the energy consumption and provide predictive maintenance. Specifically, LoRa sensors in such project send alarm to the network server if the temperature, the humidity, or the CO_2 levels were beyond specific thresholds. They claim that this project saves more than one million Canadian dollar per year [49]. Another LoRa project in the same country was deployed to ensure the healthiness of the learning and working environment in Quebec schools. The main requirements were the long battery life to last at least 4 years, periodic data transmission with a rate equals to 1 message per 5 minutes, and easy to install [50]. Similarly, a large-scale LoRa network was deployed in different multiple US sites to enhance the cleaning and maintenance capacity of different buildings. With a combination of different occupancy sensors, people counting sensors, leak detection sensors, and indoor quality monitoring, the labor work productivity was enhanced by 15% - 30% as it was directed to the most needed spaces, according to the collected data [51].

The adoption of LoRa technology over other LPWAN options, by companies involved in the projects mentioned earlier is primarily driven by the following considerations:

- Bi-directional communication support allowing gateways to receive data from nodes to inform decision-making, while also enabling gateways to transmit commands and configuration parameters to nodes for control purposes. In fact, downlink communications is not fully supported by some LPWAN technologies such as SigFox [30], where acknowledging each uplink packet or sending frequent commands to nodes are not applicable.

-
- Scaling up is made easy, as companies can conveniently increase their capacity by incorporating additional gateways and sensors.
 - The ability to create large-scale general-purpose IoT network that serve different kinds of sensors and IoT devices.
 - Cost effective networks.
 - Secure connections.

Table 3 summarizes the main features of currently deployed LoRa use cases and projects.

Table 3: Some large-scale applications of LoRa networks.

country	Domain	area	No. of devices	No. of gateways	No. of messages/day
SA [40]	Safety and security	35,000 km^2	39,000	Unknown	unspecified
Province of Palermo, Italy [41]	smart cities	50 km^2	5000	20	480,000
Sweden [48]	smart buildings	Scaling in all of Sweden, expanding internationally	10,000	1000	960,000
Canada [49]	smart buildings	unspecified	4400	440	unspecified
Canada [50]	smart buildings	unspecified	47000	2600	7,331,376

US [51]	Industry	10 million square feet	> 9000	Hundreds of gateways	10 million data points gathered from deployed devices to date
Coventry, Warwickshire, United Kingdom [42]	smart cities	unspecified	150,000	unspecified	unspecified
CA USA [44]	smart cities	Huntington Park	1700	18	unspecified
New Zealand [47]	smart agriculture	farms across the South Island of New Zealand	> 4000 with orders of 7.5k for balance of 2022	> 25 and growing	7.5 million per month and growing
Spain [43]	smart water metering/ smart cities	> 50 cities	> 300,000 devices	unspecified	unspecified
Czech Republic [45]	smart cities	entire country	unspecified	580	unspecified
Florida, US [46]	smart cities	1,600 square miles	unspecified	> 50 tower sites	unspecified

2.7 LoRa Performance Goals

Although different LPWAN technologies share common design goals such as the long transmission range, the low power consumption, and the low deployment cost, each one of them has different performance goals depending on the observed performance of the technology either in real deployments or in simulated environments. While the design goals determine the logic behind

conceiving such a technology, achieving the performance goals maintain a technology alive and viable by guaranteeing an acceptable performance for such a technology. According to that, LoRa performance goals can be summarized into three main goals, namely scalability, energy consumption, and Packet Delivery Ratio (PDR). Fig.7 shows the main performance goals for LoRa networks.

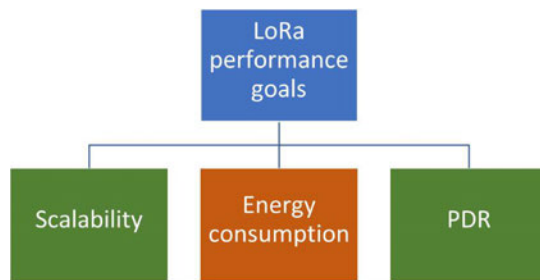


Figure 7: Performance goals of LoRa networks

In the context of LoRa, scalability has two distinct meanings. On one hand, scalability can be seen as the achievable coverage area within a given location with specific characteristics that serve a specific application while meeting a given Quality of Service (QoS) metrics. Under this definition, there are different factors that may affect the scalability with the main factor being the terrain of the environment itself. As demonstrated by different experiments conducted by Liando et al. [52], the rural or semi-urban environments with short building and a possible Line of Sight (LoS) between the sender node and the gateway have longer transmission ranges compared to the urban environments with high and crowded buildings. Other important parameters that have an impact on the transmission range, and hence the coverage area, are the transmission powers (TX) and the spreading factors (SFs) used by a given node. Higher TXs and higher SFs allow nodes to send over greater distances as higher SFs offer improved gateway sensitivity. An experiment has been conducted in [53] to demonstrate the maximum distance for each SF in different environments, including the urban, sub-urban, and rural areas. In the urban scenario, the coverage area was between 2.8 Km, with the highest data rate using SF7, and 6.5 Km with the lowest data rate using SF12. In rural areas, where line-of-sight communication can be achieved, the maximum distance that could be achieved is 9.6 Km and 18.5 Km with SF7 and SF12, respectively. For a given environment terrain, using some techniques to adapt the TXs and SFs of nodes can greatly affect the scalability of the network.

On the other hand, scalability can be defined as the maximum number

of connected devices that serve a specific application while meeting a given Quality of Service (QoS) metrics. In this context, the main factor that affects the scalability from the gateway side, the intended receiver, is its ability to decode simultaneous signals received either from different physical channels or simultaneous signals received on the same physical channel. From the end device perspective, configuring nodes with the most appropriate combination of transmission parameters such as the SF and the communication channel greatly enhances the total number of concurrent successful transmissions at the gateway and hence the scalability of the overall network. Additionally, the medium access mechanisms have a significant impact on enhancing the network's scalability. Effectively managing medium access helps reduce the likelihood of collisions during simultaneous transmissions, ultimately increasing network capacity and scalability[54].

Besides the network scalability, energy efficiency plays a significant role in determining the overall network lifetime. According to [55], the power consumed for packet transmissions is entirely depending on the transmission technology, which is in this case the LoRa technology. In a study described in [52] researchers conducted an experiment aimed at determining the lifespan of nodes in the context of two extreme LoRa parameter configurations. In their experiment, the initial setup involved configuring a node to utilize SF7 with a transmission power of 2 dBm. In the second setup, a node is adjusted to utilize SF12 with a transmission power level of 20 dBm. The first and the second configuration of LoRa parameters consider the least and the most energy consumption settings, respectively. While the claimed average lifetime of LoRa nodes is expected to be around ten years, the study showed that the average lifetimes for the first and second configurations were 4.60 years and 1.37 years, respectively. This suggests that achieving the promised ten-year lifetime that LoRa end nodes require would be achievable only through careful parameter configuration. In this context, resource allocation and duty cycling emerge as the primary determinants influencing nodes' energy consumption and, consequently, their lifespans.

Packet Delivery Ratio (PDR) can be defined as the ratio of the successful delivered packets over the total transmitted packets in the network. Various research studies have focused on enhancing the PDR of LoRa networks by means of decreasing collision rates and interference or the end-to-end delays. Different factors affect the PDR of a LoRa network. Firstly, the spreading factor of a transmitted packet plays a crucial role in its likelihood of being received. Higher spreading factors (SFs) offer increased sensitivity levels at the gateway, resulting in a higher PDR for packets decoded with higher SFs. As demonstrated in [53], the PDR of packets transmitted with SF between 9 and 12 for a distance less than or equal 3 km in urban scenario was al-

most 100%. Hence, one viable strategy to enhance network reliability, and subsequently improve its Packet Delivery Ratio (PDR), is by increasing the Spreading Factor (SF) of transmitted packets. However, higher SFs means higher packet size and hence higher collision rates, especially with ALOHA-based access to the medium. Furthermore, higher SFs means also longer Time on Air (ToA) and hence higher end-to-end delay. Consequently, there is a compromise between the used SFs and the achieved PDR that requires further investigation to find the optimal distribution of SFs among end devices that optimize the PDR. As stated in [53], higher SFs should only be used with far-located nodes ($> 3km$ in urban scenario) as the PDR for nodes located in closer area ($< 3km$) with lower SFs (SF7 and SF8) is almost 100%.

Another important factor that affects the PDR of the network is the Coding Rate (CR). As mentioned in Section 2, LoRa supports four values of CRs, which are, $4/5$, $4/6$, $4/7$, and $4/8$. Practically, the coding rate of Forward Error Correction (FEC) is the ratio of bits of a given packet that actually represent useful data [56]. In other words, loRa adds one or more redundant bits every specific number of raw bits to allow the receiver to decode the bits in case they were corrupted during the transmission. Specifically, $4/5$ coding rate means adding one redundant bit for each 4 raw bits of a packet, hence every 4 raw bits will be transmitted as 5 bits where one redundant bit is included. Similarly, $4/8$ means adding four redundant bits for every 4 raw bits of a packet. As we can notice here, lower coding rates will result in higher link reliability and hence higher PDR. However, this could be at the cost of increased packet size and hence the ToA of a specific packet is increased which may lead to higher collision rates. Here again there is a compromise between the coding rate and the achieved PDR that necessitates further investigation to find the optimal distribution of coding rates among IoT end devices.

2.8 Conclusion

LoRa networks have attractive features that can be summarized in the long-range, the low cost, the low energy consumption, and the low data rates. These features make them appealing for a variety of IoT applications where the massive connectivity over long transmission ranges is the most important thing they need. According to LoRa alliance, LoRa networks are able to connect hundreds of thousands of nodes that have low traffic rate over long transmission ranges. However, practical experiments and simulations have shown that the claimed performance is far from being achieved in large scale networks that have large number of connected nodes. This is mainly because of the high interference, resulting from the use of the shared free unlicensed

band and the high collision rates due to the use of the ALOHA channel access. Furthermore, LoRa networks are constrained with the duty cycle feature of the open band and this limitation is critical especially for downlink traffic. According to that, the next chapter introduces the main limitations that are identified in LoRa networks that prohibit satisfying the aforementioned performance goals.

3 Assessing the Effectiveness and Reliability of LoRaWAN MAC and Data Link Layers' protocols: A Comprehensive Review

Despite LoRa immense promise in reshaping IoT connectivity, LoRa does have some shortcomings and challenges that yet to be addressed by the research community to unleash its full potential. These constraints have triggered substantial attention from diverse entities, including research institutions, organizations, and industry stakeholders. This chapter surveys the literature aimed at improving the capacity and scalability of LoRa networks specifically at the MAC (Medium Access Control) and data link layers. Unlike other surveys, this study focuses on these layers because they play a pivotal role in managing collision rates, which significantly impact network scalability. The chapter suggests a comprehensive review of the literature, by organizing it based on key limitations that could hinder the network's ability to meet its performance objectives, including scalability, Packet Delivery Ratio (PDR), and energy efficiency.

Although LoRa networks have been conceived to achieve the previously mentioned performance goals, there are still some limitations that prohibit them from achieving these goals especially in large networks. In this chapter, the related work is classified based on the limitation that prohibit LoRa networks from achieving their performance goals. Three main limitations are identified in LoRa networks. The first limitation is the ALOHA based medium access mechanism. The second limitation is the limited Duty Cycle (DC) imposed especially on the downlink traffic from the gateway to nodes. The third observed limitation is the proper configuration of transmission parameters for LoRa nodes. Fig.8 shows the classification of the related work based on these limitations. Section 3.1, 3.2, and 3.3 discuss the related works that address these limitations.

3.1 ALOHA Medium Access Mechanism Alternatives

As known, LoRa nodes use the ALOHA medium access mechanism to access the shared medium. Hence, whenever a node generates a packet, it will immediately proceed with the transmission without any carrier sensing or time regulation mechanisms. Using such a mechanism maintains the power efficiency of the node's transceiver as the ALOHA protocol is simple and does not waste energy in listening to the medium or synchronizing the nodes with other network parties. Nevertheless, the scalability of LoRa networks

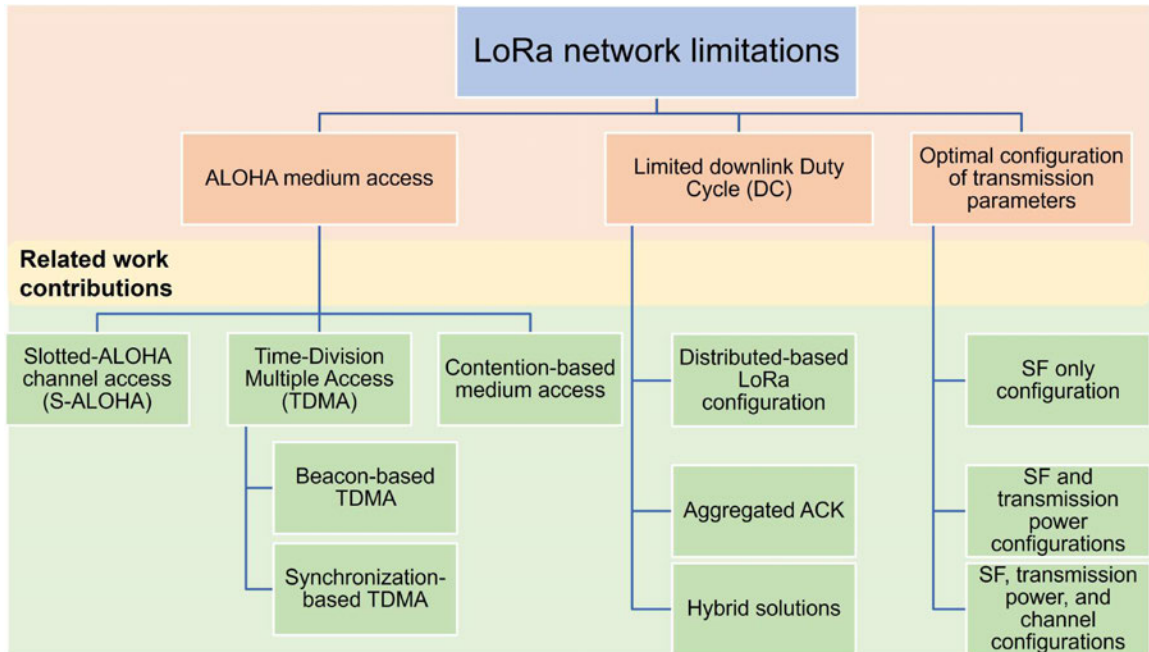


Figure 8: Classification of the related work based on the main limitations of LoRa networks.

is constrained by ALOHA channel access, particularly as the number of connected nodes rises. In fact, the maximum achievable channel capacity using ALOHA is only 18% of the total channel capacity as reported in [12]. Therefore, different studies investigate different alternative approaches to ALOHA channel access that can be summarized into three main approaches: 1) slotted ALOHA (S-ALOHA), 2) Time-Division Multiple Access (TDMA-based) approaches, and 3) contention-based channel access mechanisms. The following sections discuss these approaches.

3.1.1 Slotted-ALOHA Channel Access: S-ALOHA

In the S-ALOHA channel access medium, the time is divided into time slots of length T where each slot consists of two main periods, the transmission period and the guard period. Each node is allowed to only transmit at the beginning of a given slot. In other words, no transmissions are initiated at the middle or at the end of a given slot. Most importantly and as opposed to TDMA access where a node is assigned a specific timeslot for its transmissions, S-ALOHA enables nodes to independently and randomly pick a slot for their transmissions. By adopting S-ALOHA, the maximum achievable capacity of a channel is improved to 37% as shown in [12] where they implemented the

S-ALOHA on top of LoRaWAN layers. They divide the time into fixed-size slots with a length of 2 seconds. With 1.6 s for the transmission period and 0.4 s as the guard duration. To synchronize nodes with the gateway, both nodes and gateways attach an 8-byte timestamp to their packets at the end of the transmission time of a packet. Hence, once a node receives an ACK from the GW, it will retrieve the attached timestamp to synchronize with the gateway. The synchronization in this algorithm is implemented as a service at the application layer without changing the underlying LoRaWAN layers [57]. However, a drawback arises when nodes struggle to synchronize if the ACK is missed due to collisions. Furthermore, applying fixed timeslot intervals is not optimal for LoRaWAN networks that have different transmission durations based on different configuration parameters. Loh et al. in [58] performed a simulation study to determine the most appropriate S-ALOHA parameters, (i.e., slot length and the guard time length) under different SFs and payload sizes. They found that for optimal slot length, the slot duration should be set equal to the Time on Air (ToA) of a given node. In fact, the capacity of the network is getting worse with the increase of the slot length even with small SFs and payloads. They found that small guard time is recommended for small payloads and SFs. Consequently, they recommend implementing an adjustable guard time proportional to the slot duration to enhance the network's overall capacity. Their performance evaluation demonstrates that S-ALOHA outperforms pure ALOHA albeit only in cases involving lower SFs, demonstrating a maximum potential network capacity improvement of 22.4%.

Chasserat et al. in [59] modified Class B LoRaWAN devices through the use of S-ALOHA. They named the proposed solution as Class S, where the gateway sends beacons to nodes in the specified beacon interval, similar to Class B. However, the duration between beacon transmission periods is further divided into equal-size slots with a duration that accommodates the largest possible ToA in the network. Any slot can be used in sending or receiving data from/to nodes. Hence, a possible unused time within a timeslot could result in shorter ToA transmissions. With a number of nodes around nine thousand, the throughput was around 0.28 while Class A devices' throughput was 0.06 for the same number of nodes.

Beltramelli et al. in [60] proposed an S-ALOHA medium access with an out-of-band synchronization using FM-radio data system (FM-RDS) to increase the scalability and the energy efficiency. They leverage the clock time and date (CT) group to send synchronization information to nodes every one minute. Nodes, on the other side, must be equipped with FM-RDS receivers to receive such messages. In order for a node to proceed with its transmission, it must receive two CT-group packets and wait for a randomly

selected time slot for its transmission. Their results show that the success probability of the proposed protocol were below 50%. However, the proposed protocol outperforms the legacy LoRaWAN in terms of the energy efficiency and success probability only with higher SFs and payloads, as nodes do need to listen frequently to transmit small-sized frames.

Beltramelli et al. in [61] conceived an analytical model that compares the performance of LoRa network under S-ALOHA and non-persistent carrier sensing multiple access, NP-CSMA, channel access approaches. The simulation results illustrates that the network success probability was enhanced by 16% with S-ALOHA compared to pure ALOHA, P-ALOHA, used by legacy LoRaWAN. Furthermore, the coverage probability of S-ALOHA is better than P-ALOHA with the increase in the number of nodes due to the decrease in collisions. Specifically, S-ALOHA outperforms NP-CSMA and P-ALOHA when nodes are located far from the gateway and use larger SFs. Furthermore, S-ALOHA is more energy efficient than P-ALOHA, especially for far-located nodes.

Valkanis et al. in [62] proposed a reinforcement learning-based algorithm to dynamically determine the parameters of the backoff algorithm in S-ALOHA channel access. Specifically, they modify the size of the Congestion Window (CW) value based on the observed packet successful ratio in a given traffic period. The CW in this context means the duration for which a node waits before transmitting its packet. They use Learning Automata (LA) algorithm to adjust the minimum boundary of the CW. In other words, when the server observes a higher number of received packets in the previous period, it will decrease the CW size and conversely, it will expand the CW size when the number of received packets in the previous period decreases. The updated CW size is transmitted to nodes by including it in the ACK packets. However, the proposed algorithm is based on a centralized approach, where the network server observes the network, updates the parameters, and delivers them to nodes. In case the server has reached its DC limit, it cannot update the nodes with the modified parameters, which will affect the network integrity. Furthermore, the proposed algorithm assumes periodic fixed data generation rate for all nodes, which is not appropriate with sporadic LoRa network scenarios.

Tsakmakis at al. in [63] replaced the ALOHA with either an S-ALOHA or TDMA MAC according to the network traffic rate. In other words, they used the Reinforcement Learning (RL) mechanism, in a cluster-based network, in order to dynamically select the MAC protocol based on the traffic rate to be transmitted by the nodes. In other words, an agent in the server selects between S-ALOHA or TDMA MAC protocol to be used by the node at the beginning of each frame to minimize the delay of packets in the event-driven

network. Specifically, based on the traffic that needs to be transmitted in the previous frame, an agent at the server selects either S-ALOHA or TDMA MAC scheme to be used in the next frame. TDMA approach is used if high traffic is calculated in the previous frame while S-ALOHA is used if low traffic rate is observed in the previous frame. Subsequently, the server initiates the transmission by sending a beacon request to the cluster head. The cluster head in turn broadcasts a wake-up packet requesting the nodes to send their data if any. Their simulation results show that the maximum achievable throughput using the proposed approach is 29%.

Yapar et al. in [64] proposed a slotted-ALOHA protocol with aggregated acknowledgment. They divide the channel into six overlapping superframes where each superframe is assigned to a different SF. Then, they divide each superframe into groups of uplink slots followed by one downlink slot. The duration of the downlink slot is different based on the used SF on a given superframe. Nodes need first to get schedule information from the server to proceed with their transmissions. Subsequently, each node calculates its own transmission schedule. However, the authors did not clarify how nodes determine their appropriate SF and hence their superframes. The downlink transmissions are scheduled in specific slots per superframe, unlike LoRaWAN where two downlink slots are opened after each uplink transmission. The proposed protocol assumes single-channel communication. Furthermore, the superframe duration as well as the duration for uplink traffic in each superframe is the same regardless of the network size. This could result in limiting the scalability of the network.

Triantafyllou et al. in [22] proposed a time-slotted protocol called FCA-LoRa that divides the time on all supported channels into fixed-size frames that are constructed by a set of consecutive timeslots. Each time slot is further divided into beacon reserved interval and beacon period interval, where the beacons from the gateway and uplink transmissions from nodes are sent during the first and second intervals, respectively. At the beginning of each frame, gateways send beacons containing synchronization information on all supported channels to ensure that nodes receive beacons regardless of the channel they are listening to. During the beacon period, nodes select randomly a slot and perform Listen Before Talk (LBT) for a random offset before transmission to mitigate collisions among nodes. High energy consumption is expected as nodes are enforced to listen to the medium most of the simulation time either to receive beacons or to check the medium before each packet transmission. Table 4 summarizes S-ALOHA based works by providing the main idea, the enhancement goal, and the enhancement level of each algorithm.

Table 4: S-ALOHA based research works to enhance LoRa networks.

Ref	Year	enhanced goal	methodology	Pros	Cons	Enhancement level
[12]	2018	PDR	Implement S-ALOHA instead of P-ALOHA	Time synchronization is implemented as an independent service at the application layer	The evaluation of the performance used few number of nodes (20) with fixed transmission parameters	Medium. The network throughput was enhanced by 2.
[63]	2022	PDR	Use Reinforcement Learning (RL) to select the appropriate MAC	Adaptive MAC	Fixed timeslot	Low. The Packet Success Ratio (PDR) was improved by 30%.
[64]	2019	PDR	divides the channel into 6 super frames. each super frame is further divided into m groups. each group consists of n UL slots and 1 DL slot.	Agg ACK, different UL slot duration among each super frame.	Limited number of groups per super frame and it is not related to the network size.	Medium. The PDR was improved by 40% with 5000 nodes and 20% with 7500 nodes.

[22]	2021	PDR	Time is divided into fixed-period slots, with a reserved slot for beacon receiving at the beginning of each frame. Nodes select a random slot and perform LBT on that slot before transmission.	Less down-link traffic as the gateway sends beacons at the beginning of each super frame period on different channels considering the DC of GW as well as channel.	Fixed number of timeslots, fixed slot duration, and fixed frame sizes regardless of the network size. High energy consumption as nodes are enforced to listen to stay active and listen to the medium most of the simulation time either to receive beacons or to check the medium before each packet transmission	Medium
[62]	2021	PDR	Use RL to determine S-ALOHA parameters	dynamic Contention Window (CW) size	Fixed SF. Small-scale networks.	Medium. The normalized throughput was improved by 57%.

[58]	2022	Capacity	Perform simulative experiment to determine the most appropriate parameters for S-ALOHA, namely the slot length and the guard time duration with different SFs and payloads	Dynamic SFs and dynamic payloads.	The maximum enhancement of the network performance using S-ALOHA is 22.4%.	performance analysis study
[59]	2022	Scalability	Implement S-ALOHA for Class B devices	Replace the random access with S-ALOHA	Fixed slot duration regardless of the used SF	High. The throughput was enhanced by 366% compared to the pure ALOHA.

[60]	2021	Energy efficiency	propose an S-ALOHA medium access with an out-of-band synchronization using FM-radio data system	Enhancing the FM-RDS to be more accurate by allowing nodes to receive 2 CT-groups packets and then a node can estimate its synchronization information. Nodes select randomly a slot for their transmission. Hence, no downlink transmission is needed from the server	Nodes must be equipped with FM-RDS receivers to receive such messages. Inappropriate for applications with sporadic transmissions with small payloads in terms of energy efficiency.	Medium. Less than 50%.
------	------	-------------------	---	--	--	------------------------

3.1.2 Time Division Multiple Access Mechanism

Another alternative approach to access the channel in LoRa networks instead of the ALOHA random access is the Time-Division Multiple Access (TDMA). Various research studies have put forth diverse TDMA strategies for LoRa

networks, and in this section, we delve into the latest and most promising ones. Indeed, the majority of TDMA-based algorithms suggested in existing literature employ centralized methodologies where nodes ascertain their suitable transmission parameters by receiving direct downlink information from the server. In other words, before a node proceeds with its transmission, it must receive some control packets from the server to determine its suitable transmission parameters, namely its slot identifier.

We further divide the centralized TDMA approaches into beacon-based approaches and synchronization-based approaches. In beacon-based approaches, the gateway sends beacons to nodes containing control information at the beginning of each frame. Nodes use these beacons to determine their transmission parameters and timeslots. Hence, the communication in this approach is always initiated by the server rather than the nodes. In the synchronization-based approaches, nodes initiate the communication by sending synchronization requests to the server requesting synchronization information. This approach is more compatible with LoRaWAN Class A devices, where the communication is initiated first by nodes, not the server.

Beacon-based centralized TDMA algorithms Leonardi et al. in [65] proposed a beacon-based TDMA MAC protocol that supports both stationary and mobile nodes. The proposed protocol divides the time into super-frames, which are further partitioned into five main periods. The first period is dedicated for beacon broadcasting to synchronize nodes with the server. The second period is dedicated for aperiodic traffic, where nodes compete to access the channel using ALOHA, similar to LoRaWAN. The third period is dedicated for the periodic traffic where this interval is further divided into equal-sized timeslots. Each node is assigned a timeslot along with a channel and a given SF. Channels and SFs are assigned randomly to nodes, while timeslots are assigned offline. An Enhanced version of the proposed algorithm was presented in [15] where they propose a protocol, named RT-LoRa, that allows the server to send multiple beacons with multiple SFs to let nodes recognize the most appropriate SF for their transmissions. RT-LoRa supports both real-time periodic traffic and non-real-time periodic traffic by assigning different time intervals in the frame for each one of them. Nodes access the channel to transmit periodic traffic using S-ALOHA. On the other hand, the TDMA approach is used to access the channel for real-time periodic traffic. Similar to [65], authors did not clarify how nodes are assigned their timeslots ID without compromising the limited DC of the server. Furthermore, sending multiple beacons with different SFs could result in longer end-to-end delays and more energy consumption, especially with large SFs.

Lee et al. in [16] proposed another beacon-based TDMA MAC protocol in which the uplink transmission period (UTP) is separated from downlink transmission period (DTP). At the beginning of each frame, each gateway uses a different SF to transmit its beacons to nodes. After the beacon period and during the UTP, nodes access the channel randomly to transmit their packets. When UTP elapses, the DTP starts, where gateways send group ACK (GACK) using the same SF used in the uplink transmission. The DTP is divided into timeslots to avoid collisions between GACK packets. The timeslots and SFs are assigned to gateways such that at a given time no two gateways use the same SF and timeslot. The GACK contains the addresses of the devices to be acknowledged. The number of devices acknowledged in a given GACK packet depends on the SF used to transmit the GACK. In other words, higher SFs implies a lower number of acknowledged nodes.

Hoang and Oh in [17] proposed another beacon-based TDMA protocol that supports both periodic and aperiodic data traffic. Unlike [65] and [15], where the interval of periodic traffic and aperiodic traffic are isolated in two different durations inside the superframe, in [17], the timeslots of periodic traffics and aperiodic traffics are interchanged. In other words, the server assigns timeslots for nodes with periodic traffic using the same approach used in [66]. Then, the remaining unassigned slots are used for aperiodic traffic by allowing nodes to compete using CSMA approach. Further details regarding the contention-based access are explained in Section 3.1.3. In this approach, they take advantage of the empty slots between the scheduled slots to be used for aperiodic traffic reducing hence the waiting time for event-driven packets.

Triantafyllou et al. in [22] proposed a beacon-based TDMA MAC protocol, named FCA-LoRa, where the gateway sends multiple beacons with different SFs on different channels in order to ensure beacon delivery to nodes. The frame in FCA-LoRa [22] is divided basically into beacon interval and beacon window. During the beacon interval, nodes are listening to receive beacons from the gateway. However, the time in the beacon window interval is divided into fixed-sized slots. A node selects a random time slot for its transmission and performs Listen Before Talk (LBT) to mitigate collisions among other nodes. Although the proposed protocol is considered centralized, gateways broadcast beacons only at the beginning of each frame interval. However, energy consumption is high as nodes remain in the listening mode most of the time either to receive beacons or before transmitting their packets.

Triantafyllou et al. in [67] proposed a recent beacon-based TDMA protocol with variable slot duration depending on the used SF and payload. The time is divided into frames, with a beacon interval at the beginning of each

frame. Each frame is further divided into $6 \times K$ timeslots, where 6 corresponds to the number of supported SFs and K corresponds to the number of supported payloads. In order to avoid collisions among nodes that use the same SF with different payloads, each payload size range is assigned a different channel. In other words, a payload size range is assigned for each channel and that specific payload size can be used with different 6 frames in that channel corresponding to the six supported SFs. The nodes determine their timeslot ID based on the DevAddr parameter that was sent during the node's joining phase to the network, similar to [23]. Hence, the same slot ID is used by a given node regardless of the used SF or the payload size. Although the slot duration depends on the SF and the payload, the frame duration is fixed. Consequently, the number of available timeslots per frame is actually the frame duration divided by the slot duration. Hence, large SFs frames have less number of timeslots. This may limit the capacity of the network as the number of available slots is fixed for any SF no matter how many nodes are using a given SF.

Hoang et. al. in [66] proposed a TDMA-based protocol that assigns logical slot indices to the time slots of each frame. The server collects first the number of slots needed per node based on their traffic. Then, it constructs a frame, where their slots are tagged with logical indices besides the physical ones. The logical indices are sequential when assigned to nodes while their corresponding physical indices are not. The main idea behind it is that when a node needs to transmit multiple packets, it uses physically nonsequential timeslots to comply with its DC constraints. Indeed, the server needs to decide and disseminate the logical indices to nodes. Furthermore, the server needs to disseminate control packets during the downlink timeslot of each frame to inform nodes of their scheduling information, which they use to anticipate their own logical indices for their transmissions. Subsequently, nodes will send their packets in the physical indices that correspond to the logical calculated ones. Extensive downlink traffic is needed to inform nodes of their schedule. The proposed protocol assumed fixed packet transmission time and fixed frame duration. Furthermore, they assume in their algorithm that all SFs and a number of channels are supported. However, during the performance evaluation, they only assume 2 SFs in the analysis evaluation and one SF in the experimental evaluation. Table 5 summarizes the beacon-based TDMA MAC protocols discussed in this section.

Table 5: Beacon-based TDMA MAC protocols for LoRa networks.

Ref	Year	enhanced goal	Main idea	Enhancement level	Notes
[65]	2018	PDR	Support both aperiodic event-based traffic and periodic real-time traffic	Very low	no packet lost in CFP. In CAP as follows: stationary located at 250 meter the Packet Loss Ratio (PLR) is 95%. No comparison with other protocols.
[15]	2019	PDR	Support both aperiodic event-based traffic and periodic real-time traffic	Low	PLR enhancement compared to [65] was 26%
[16]	2021	scalability	provide Time-Slotted approach for sending group ACK	High	The maximum number of supported nodes is 2500 nodes compared to a maximum number of only 500 nodes in LoRaWAN.

[17]	2022	PDR	The timeslots of periodic traffics and aperiodic traffics are interchanged	Very high	Improvements compared to ILoRa [15] = 288% Improvements compared to RT-LoRa [65]= 133%
[22]	2021	PDR	Combine S-ALOHA with LBT to improve collision rate	Medium	Scalability is low as it is evaluated on very small area with small number of nodes (max 500).
[67]	2022	PDR	Variable slot durations based on SF/payload	Low	The average enhancements in the PDR with $N = 500$ compared to the ALOHA is 37%.

[66]	2020	PDR	Tag the timeslots with logical indices to simplify the process of disseminating the slot IDs to nodes	Low	The average improvements of the PDR in the proposed scheme compared to S-ALOHA is 22.5%. The evaluation assumes only one SF while the algorithm supports all SFs.
------	------	-----	---	-----	---

Synchronization-based centralized TDMA algorithms In the synchronized-based TDMA approach, nodes initiate communication with the server by sending synchronization requests either randomly or according to a predetermined schedule. In [18], nodes transmit synchronization requests that include, among other parameters, their traffic periodicity, allowing the server to determine the number of slots needed to assign to each node. The server responds with a synchronization reply that includes the timeslot indices for each node during the specified synchronization period. To minimize the size of the synchronization reply packets, the timeslots indices are embedded in a probabilistic data structure. Each node maintains an index of the current timeslot identifier. Nodes investigate the received vector by checking if the next timeslot index is set to one in the received data structure. If affirmative, they will initiate their transmission during the timeslot; otherwise, they will postpone it until the following traffic period. In other words, a node searches for its timeslot indices by incrementing and comparing its time slot index to the indices received in the reply packet, which may consume time and energy. Moreover, some nodes may be prevented from timeslots to communicate if the number of connected nodes exceeds the available frame size. This will result in limited network scalability.

Different synchronized-based algorithms are proposed in the literature [68] [19] [69] that provide bulk data transmission, where nodes buffer their

packets and transmit them in a pre-defined period known by both the server and nodes. Zorbas et al. in [68] proposed a synchronization-based TDMA offline scheduling algorithm, where nodes communicate their buffered data size and the minimum SF they are entitled to within their synchronization requests. The server responds individually to each node with a tuple consisting of the node ID, the SF, and the slot ID for each transmission. Consequently, one can anticipate lengthy reply packets, particularly when dealing with large buffer sizes. This scenario may raise some issues related to the limited DC of the gateways. Furthermore, the algorithm assumes that the number of connected nodes as well as their appropriate SF are known to the gateway, which are unrealistic assumptions. The work was extended in [19] by formatting the bulk data transmission in a linear algorithm to minimize the data collection time without violating the DC constraints by minimizing the size of the frames. However, the high complexity of the minimization function with a large number of decision variables can not be practical with a large number of connected nodes. Hence, they only compute the optimal function for a maximum network size of 50 nodes. The proposed algorithm has less data collection time compared to [68] at the expense of high computation time.

In [69], nodes send first their synchronization request that includes the amount of buffered data. The server responds accordingly with the appropriate SF, TX, channel, and slot identifier in the synchronization reply. This process is repeated at the beginning of each data collection period. They propose an algorithm for SF allocation with two different objectives, either to minimize the energy consumption or the collection time. Although the algorithm has high reliability thanks to the support of global ACK, it suffers from limited scalability as each node should send “join request” and should receive “join accept” packets from the gateway at the beginning of each data collection period. This will increase the overhead of the network especially if the data collection period is not large enough to compensate for the synchronization overhead.

Xu et al. in [70] proposed a synchronization-based TDMA protocol named S-MAC algorithm based on the assumption of SF orthogonality and traffic periodicity of nodes. The main idea is to use a maximization algorithm to assign the nodes with the same SF different channel or time offset to mitigate collisions among them. The server predicts the traffic periodicity and the SF based on the received uplink packets from nodes. Hence, the server performs and disseminates the appropriate channels using a channel optimization scheduling algorithm. Thus, the server recommends to nodes the channel and potentially the transmission offset while the SF is fixed for each node. To accomplish this, the server attaches the channel configuration

information in a reply packet and sends it during the node's receive windows.

Garrido-Hidalgo et al. in [20] proposed a synchronization-based TDMA protocol based on an online multi-agent approach to assign transmission parameters to nodes. Specifically, nodes exchange with the server four up-link/downlink packets to synchronize and get their transmission parameters before proceeding with their data transmissions. The proposed algorithm is totally centralized, where at the warm-up period it collects data from nodes. Subsequently and according to the collected information, different agents for each SF determine the appropriate SF, channel, and timeslot for each node. However, the proposed algorithm did not provide a clear methodology of how the schedule will be disseminated among nodes with the limited DC of gateways.

Haiahem et al. in [71] proposed two variants of synchronization-based TDMA algorithms to avoid collisions among nodes. The first approach is TDMA-based, where nodes are divided into clusters and each cluster is assigned a specific duration in the time frame such that different clusters are transmitting in sequential. Inside a cluster, a maximum of six nodes are assigned to each cluster, which corresponds to the number of supported SFs in the network. Nodes belonging to the same sub-cluster are transmitting on parallel without collisions as they use different SFs. The slot duration is fixed and based on the highest ToA of the largest SF supported by a given sub-cluster. Hence, there will be wasted time in the frame, especially with small SFs. The second algorithm is a Frequency-Division Multiple Access, FDMA approach, where each cluster is assigned a specific channel and SF. Nodes belonging to different clusters transmit in parallel, whereas nodes belonging to the same cluster transmit in sequential. Furthermore, assigning nodes to clusters and sub-clusters is performed at the server side. However, with a large number of connected nodes, the network server might not be able to inform all nodes of their clustering information due to the limited DC.

Pullmann and Macko in [72] proposed a synchronization-based TDMA-based approach that takes advantage of the periodicity of node's traffic and their packet size to provide a specific schedule for each node. According to [72], the time frame is divided into segments, and each segment is further divided into two main intervals where the first interval is dedicated to the periodic traffic for nodes and the second interval is dedicated to the aperiodic traffic. Each interval is further divided into timeslots that accommodate one or more transmissions. A unique timeslot is assigned by the gateway to each node in the periodic traffic interval while S-ALOHA is used during the aperiodic interval. To assign timeslots to nodes, nodes exchange with the gateway four packets during the joining phase, which includes their traffic

periodicity and packet size. The gateway is fully responsible for planning the schedule and disseminating it among nodes. The paper did not clarify how the gateway is capable of responding with 2 packets per node during the joining phase without violating its DC. Also, the paper did not clarify how other parameters, such as the SF and channels, are distributed among the nodes. The overhead of the protocol is expected to be high, as the gateway, besides the two packets sent to each individual node during the joining phase, has to send periodic beacons to synchronize nodes.

Chinchilla-Romero et al. in [21] proposed a synchronization-based TDMA algorithm, named Collision Avoidance Resource Allocation (CARA) algorithm for LoRa networks. They are taking advantage of the multichannel support and the orthogonality of SFs by defining a set of Resource Blocks (RB) consisting of a unique and different SF and channel. Hence, in Europe, there are 48 different RB since it has 6 SFs and 8 channels. The gateway is responsible for determining the initial RB and the set of eligible SFs for each node, as well as delivering them in the “joint-accept” message. The gateway maintains a matrix where rows represent the devices and columns represent the timeslots. A given matrix entry stores the RB of a node in a given timeslot. Hence, collision-free transmissions are only achieved if the number of nodes is less than the number of RBs. Furthermore, the proposed algorithm assumes a fixed timeslot duration, which may not be optimal with the dynamic nature of LoRa networks.

Leonardi et al. in [73] proposed a synchronized multi-hop TDMA protocol to increase the coverage area of LoRa networks. The nodes are organized in a tree structure, where the gateway represents the root of the tree. Nodes are organized in layers such that nodes belonging to the closest layer to the gateway will use the smallest SF and nodes belonging to the farther layer use larger SFs. Different kinds of nodes are supported in this network. The end node is the leaf node that is responsible for submitting its own packets. However, nodes in the upper layers are responsible for either submitting their own packets or forwarding packets coming from the lower layer. The time frame is divided into transmission and reception slots, where a transmission slot in an upper layer corresponds to a receiving slot in the lower layer. So, the transmissions among related layers are synchronized. Nodes’ assignment to the layers as well as the assignment of the parameters such as the SF, TX, and timeslots are done offline during the setup period of the network. Hence, the proposed protocol assumes that the status of the nodes as well as the total number of nodes are fixed and known in advance, which renders the protocol unsuitable for dynamic networks. Furthermore, the number of idle slots in the frame increases with the increase of the tree layers, which will worsen the end-to-end delay. Table 6 summarizes the Synchronization-based

TDMA MAC protocols discussed in this section.

Table 6: Synchronization-based TDMA MAC protocols for LoRa networks.

Ref	Year	enhanced goal	Main idea	Enhancement level	Notes
[18]	2019	PDR	The server sends to nodes the indices of the timeslots based on their traffic needs encoded in probabilistic structure to reduce the downlink packet size.	Low	For non-saturated LoRaWAN network and the synchronization was performed in a separate channel, the PDR was 7% (SF7) to 30% (SF12).
[68]	2019	PDR	Time-Slotted bulk data transmission	high	With N=1000, the PDR of LoRaWAN was 12%, where the PDR of the proposed algorithm was 100%. However, the network scaled for only 1 km.
[19]	2021	Energy efficiency/ collection time	Time-Slotted bulk data transmission	low	Hard to compute for large number of nodes. It is not applicable.

[69]	2020	Energy efficiency/collection time	Time-Slotted bulk data transmission	Very high	88% better in energy efficiency compared to the LoRaWAN.
[70]	2020	PDR	Propose a channel optimization scheduling algorithm among nodes that use the same SF.	Medium	The average improvements of S-MAC PDR compared to the ALOHA PDR is 51%.
[20]	2023	Capacity	Online resource allocation using multi-agent approach	High	Increase the network capacity for up to 667%.
[71]	2020	Capacity	Divide nodes into clusters and sub-clusters with different transmission parameters to avoid collisions.	High	The proposed algorithm achieves a PDR equals 1 for up to 2000 nodes in TDMA approach and 5000 nodes in FDMA approach.

[72]	2019	Capacity	Use TDMA-based for periodic traffic and S-ALOHA for aperiodic traffic	high	With N=2500 nodes the average of successful transmissions in LoRaWAN equals 23.90% while the Proposed equals 98.01%.
[21]	2021	Capacity	Exploit multichannel support and orthogonality of SFs to provide collision-free transmissions.	medium	Increase the capacity by almost 40% assuming a realistic propagation model.
[73]	2023	Scalability	Using multi-hop tree structure to enhance the coverage area.	medium	They assume 3% DC.

3.1.3 Contention-Based Medium Access Protocols

Different MAC approaches have been proposed to mitigate collisions and hence increase the scalability of LoRa networks by applying some Listen Before Talk (LBT) or Carrier Sense Multiple Access (CSMA) mechanisms. Unlike S-ALOHA or TDMA-based MAC protocols, CSMA does not need any synchronization with the server. However, the main concern that makes the CSMA mechanism not favorable for LoRa is the hidden node issue as LoRa

networks connect nodes in large-scale areas. In severe situations, when all nodes are hidden, the performance of CSMA becomes similar to P-ALOHA. LoRa supports a Channel Activity Detection (CAD) mechanism to detect preamble chirps. However, Gamage et al. in [74] performed an experimental study and showed that CAD detects also the payload chirps with an accuracy of 95%. This section investigates the most common approaches found in the literature.

To and Duda in [75] studied the performance of LoRa networks by applying two variants of CSMA approaches. The first approach is the regular CSMA approach, where nodes before each transmission select a random channel and perform Clear Channel Assessment (CCA) and start transmission only if the channel is clear. Otherwise, it will backoff at a random time. The second approach is called CSMA-x, where nodes perform CCA for x long period. According to their results, CSMA and CSMA-x outperform the P-ALOHA in terms of the PDR, the collision rate, and energy consumption with high network densities ($> 5000nodes$). With high traffic rates, nodes most of the time are backing off to avoid collisions. Hence, it has higher energy efficiency compared to LoRaWAN. However, concerns related to the hidden node, especially with large coverage area, are not discussed.

Kouvelas et al. in [76] implemented Persistent-Carrier Sense Multiple Access (P-CSMA) as an alternative to the P-ALOHA channel access. The main possible advantage is the mitigation of collisions and hence an increase in the PDR, especially with large number of connected nodes, is achieved. However, the main issue of CSMA approaches is how to avoid the hidden terminal problem especially when the wireless node range is high in a large scale network. Kouvelas et al. in [76] suggested to use persistent CSMA, such that a node will transmit with random probability p when the medium is sensed idle. Simulation results show that lower p values result generally in higher PDR. However, as the number of connected nodes increases, the PDR decreases.

Rochester et al. in [77] proposed a Lightweight Carrier Sense (LCS) mechanism as an alternative to the P-ALOHA to enhance the network PDR without extra overhead in terms of the energy efficiency. LCS exploits the Channel Activity Detection (CAD) mechanism that is implemented in most LoRa Class A devices. Hence, when a node has a packet for transmission, unlike CSMA, it will listen to the channel for a period of one symbol period instead of a fixed sensing time. If the channel is busy, the node will discard the packet instead of performing a backoff like in the CSMA. Hence, nodes with LCS have less burden compared to the legacy CSMA. With $N=5000$ nodes, the energy efficiency of LCS was doubled compared to the ALOHA. This is because the additional energy consumed due to the CAD in this ap-

proach is less than the energy consumed in case of collisions in ALOHA. To achieve 90% PDR, the maximum supported nodes with LCS in SF7 LoRa networks increased by 8% compared to the ALOHA network with the same SF. Although the proposed approach shows an enhanced performance in terms of the PDR and the energy, it is only evaluated in small scale area of around 0.5 Km. Hence, the hidden node problem may not be experienced. Gamage et al. in [74] proposed three variant versions of CSMA-based MAC named LMAC-1, LMAC-2, and LMAC-3. In LMAC-1, a node randomly selects a channel and a SF for the upcoming transmission. LMAC-1 uses Distributed Inter-Frame Space (DIFS) approach, where during a given DIFS, a fixed number of CADs is performed, on the selected channel and SF, followed by a random Back-Off (BO), in case of an idle channel. In LMAC-2, each node stores status information of channels and SFs, which can be collected during the CADs in the DIFS or BO periods. Hence, instead of randomly selecting channel/SF, nodes use this information to select the less busy channel/SF. Hence, better channel load balancing can be achieved. In LMAC-3, instead of storing information at the nodes level regarding the status of the channel/SF, the server broadcasts beacons containing global information regarding the status of each supported pair channel/SF. While LMAC-3 provides the most promising version among the others, it is only compatible with Class B LoRaWAN devices. The estimated LMAC overhead was about 25%.

Zhong and Springer in [78] proposed a time-slotted with Persistent Carrier Sense Multiple Access (P-CSMA). The time is divided into timeslots that accommodate the largest possible payload in LoRa. Each timeslot is divided into two intervals, the Contention Access Period (CAP) and the Packet Transmission Period (PTP). Each interval is further divided into sub slots called Backoff Period (BP) that lasts for 2 LoRa symbols. Each node before proceeding with its transmission performs a random BP and then senses the channel. If the channel is idle, the node will submit its packet in the next BP with a probability p . If the channel is busy, it will repeat the channel sensing with probability $(1 - p)$. As a main limitation of this work is that the proposed algorithm assumes a single SF network.

Herrería-Alonso et al. in [79] proposed a MAC protocol that combines the CSMA with S-ALOHA in LoRa networks. According to [79], the time is divided into equal-sized slots that are large enough for one packet transmission with the highest ToA. Unlike S-ALOHA, where node's transmissions always start at the beginning of a given timeslot, the packet transmission in the proposed approach starts at any given moment but should end at the end of the timeslot. Hence, transmissions with longer ToA start earlier than transmissions with shorter ToA. Hence, collisions could be mitigated as de-

vices with shorter packets duration can detect the transmissions with longer packet transmissions.

Hoang and Oh in [17] proposed a contention-based TDMA protocol that enables nodes with periodic data traffic and aperiodic data traffic to send data in different interchanged timeslots. For aperiodic data traffic, nodes compete to access the unassigned data slots using a two-level collision avoidance scheme. According to the first level collision avoidance scheme, a node first selects a slot on a given channel such that this timeslot is not assigned to any periodic traffic. In the second level of the collision avoidance design, the slots for aperiodic traffic are further divided into delay slots, in which each node selects a random delay slot index for its transmission. Consequently, if two nodes with the same timeslot on the same channel uses two different delay slot indices, the one with the higher delay slot index (the node that will transmit later) can overhear the transmission of nodes with lower delay slot index (the node that transmits earlier). In this case, the collisions between nodes with same slot can be mitigated. However, authors did not evaluate the energy efficiency of the proposed protocol as nodes need to stay awake and listen until its delay slot index before proceeding with the transmission. Furthermore, the performance evaluation shows promising results only with small number of nodes (≤ 300). Hence, the protocol performance remains unknown with high number of connected nodes, which is the common case for LoRa networks.

Beltramelli et al. in [61] proposed a model that compares the performance of LoRa network under S-ALOHA and non-persistent NP-CSMA channel access approaches. According to their analysis, two main factors affect the performance of CSMA for LoRa. The first factor is the sensing threshold. Higher sensing threshold lowers the performance and make it similar to P-ALOHA, while lower sensing threshold makes the performance outperforms P-ALOHA, especially with short coverage areas. The second factor is the size of the annulus that uses the same SF. Devices located farther are more exposed to the hidden nodes. Hence, they experience either unnecessary backoff or more collisions depending on the sensing threshold. Hence, NP-CSMA is preferable only with devices that are located close to the gateway and use small SFs. NP-CSMA outperforms the S-ALOHA in terms of energy efficiency only when nodes density is small and close to the gateway.

Pham and Ehsan in [80] conducted an experiment to evaluate the performance of the Channel Activity Detection (CAD) technique that is supported by most LoRa devices. According to their experiment, the CAD can reveal not just the preamble symbols, as stated by Semtech, but also the entire packet. However, the CAD performance decreases with the increase of the node's transmission range. Although the receiver successfully receives the

packets at several kilometers, it fails to detect them using CAD when the distance is above 1.3 Km. This is due to the fact that LoRa receivers can receive packets below the noise floor. Hence, adopting CAD mechanism might not be reliable as it cannot resolve the hidden node problem. To avoid the hidden node problem, Pham and Ehsan in [80] propose to use Request To Send (RTS) packets before data transmissions. When a node has a packet for transmission, it either listens first for any RTS request in the medium with probability $(1 - p)$ or sends immediately its RTS packet without prior listening with probability p . Hence, p percent of nodes will transmit their RTS packets, while $1 - p$ percent will listen before they proceed with their RTS packet transmissions. In both cases, a backoff mechanism is used before the transmission of both the RTS packet and the data packet. Further investigation is needed regarding the performance of the proposed mechanism in terms of the network overhead and the energy efficiency. Table 7 summarizes the contention-based MAC protocols discussed in this section.

Table 7: Contention-based MAC protocols for LoRa networks.

Ref	Year	enhanced goal	Main idea	Enhancement level	Notes
[74]	2023	PDR	Use CAD in DIFS and BO periods.	High	$2.2\times$ goodput improvement and $2.4\times$ reduction of radio energy per successfully delivered frame.
[78]	2023	PDR	Use of P-CSMA with S-ALOHA	High	The average throughput with N equals 500 for ALOHA was 37% and the proposed equals 90%.

[79]	2022	Scalability	Use CSMA with S-ALOHA	Medium	Assume single SF. The average throughput of p-ALOHA equals 15%, S-ALOHA equals 30%, and the proposed equals 50%
[61]	2021	Scalability	Develop a model to compare S-ALOHA with CSMA in LoRa	Medium	Assume fixed channel frequency and fixed transmission power level.
[75]	2018	PDR/ Energy efficiency	Implement CSMA and CSMA-x instead of P-ALOHA.	High	with N=10k, the PDR of CSMA-x was enhanced 6x compared to the ALOHA, while the energy efficiency was enhanced by 50%.

[76]	2018	PDR	Implement P-CSMA with variant P to mitigate the hidden nodes problem.	Low	The Maximum PDR with $N=80$ nodes was 38%. No performance comparison with other protocols.
[77]	2020	Scalability/ Energy Efficiency	Use CAD mechanism without back off.	Low	With SF10, energy consumption of ALOHA is 18% more than that of Lightweight Carrier Sense (LCS), while LCS is still able to support 2% more devices.
[80]	2021	PDR	Sending and listening to RTS packets with probability p before data transmissions	Medium.	PDR between 91% and 96%. However, they implemented in 100 m distance only.

[17]	2022	PDR	The time-slots of the aperiodic traffic is divided into multiple delay slots to mitigate the hidden node problem	Very high	With N=300 nodes, the PDR of the proposed algorithm was improved 2x compared to the RT-LoRa [65] and 3.8x compared to ILoRa [15].
------	------	-----	--	-----------	---

3.2 Addressing the Limited Duty Cycle

As known, LoRa uses the unlicensed sub-GHz ISM band and in order to regulate the access between the devices of this band, Duty Cycle (DC) is imposed. DC is the fraction of time a device is allowed to transmit in a specific band during a set timeframe. For example, in Europe band, the duty cycle for sensor nodes and gateways for uplink and downlink traffic is limited to 1% for all bands except one band that is used for downlink traffic, which allows a duty cycle of 10%. This implies that if the ToA_i is the Time on Air for node i , then that node must wait at least $99 \times ToA_i$ before transmitting again using that band. Hence, selecting the optimal SF for nodes such that their packets are successfully received at the gateway with minimum ToA is crucial. In fact, the DC limit that is imposed on the downlink traffic from gateways to nodes is more critical than the uplink traffic from nodes to gateways, as LoRa targeted large-scale dense networks with sporadic network traffic. Therefore, this section is dedicated to investigating the studies that focus on dealing with the limited downlink traffic from gateways to nodes. More importantly, when the gateway is obliged to disseminate individual configuration parameters or acknowledgments to nodes in a specific period. The gateway may reach its DC limit before updating the targeted nodes, especially in large networks, which will inevitably affect the network integrity and its performance.

Different approaches were proposed in the literature to deal with the limited DC of downlink traffic in LoRa networks. Some studies conceived

distributed approaches, where nodes autonomously determine their appropriate transmission parameters. Other studies proposed hybrid solutions, where LoRa networks are combined with other non-DC-limit technologies to bypass the DC limitation. Furthermore, some studies proposed techniques to aggregate and broadcast the downlink traffic instead of sending individual packets to individual nodes. This section explains the most recent studies related to these approaches.

3.2.1 Distributed-Based LoRa Configuration

Two different approaches have been proposed that overcome the DC limitation of downlink traffic by using distributed-based configurations. The first approach relies on gateways broadcasting global configurations to guide node behavior [81] [23] [82], while the second approach empowers nodes to make autonomous decisions based on their local environment [4].

Zorbas and O’Flynn in [81] exploited the “DevEUI” feature of devices to derive the timeslot IDs for nodes. DevEUI is a unique identifier provided by the chip manufacturer. This parameter is exchanged with the gateway during node’s joining phase. Besides that, gateways need to determine the number of nodes that use a given SF to determine the frame length. Then, using the DevEUI and the frame length that is sent by the server, nodes can individually determine their slot IDs using the modulo operator. The algorithm assumes a fixed and known number of devices per SF, which is impractical, especially with continuously expanding or dynamic networks. Furthermore, the proposed algorithm may result in empty unused slots if the actual number of nodes using a given SF is less than the assumed one. Note that these empty unused slots are increasing with the increased number of nodes per SFs since the assumed number will also increase. The proposed algorithm performs well only with a few numbers of nodes. However, when the number of connected nodes increases, the frame sizes get larger, which will highly increase the nodes’ waiting time to proceed with sending, and hence the network scalability is compromised. The proposed algorithm was enhanced in [23] by using the DevAddr parameter instead of the MAC-based DevEUI parameter along with the slot assignment procedure. In LoRaWAN, when the server receives a “join-request” packet from a node, it will generate a random address for the node, named DevAddr, and include it in the “joining accept” packet. TS-LoRa [23] exploits the DevAddr parameter to generate unique timeslots for nodes that use a given SF. In [23], the server will not generate the random DevAddr and immediately include it in the joining reply, which is the case of LoRaWAN. Instead, the server in TS-LoRa generates a DevAddr and checks if the modulo operator of that DevAddr results in an

unused timeslot for the joining node. If not, the modulo process is repeated until an unused timeslot is generated for the joining node. Obviously, the server must keep track of the used timeslots for each frame per SF. Once the node receives the “joining-accept” packet, it will calculate its timeslot using the DevAddr and the frame size that is broadcast by the gateway, using the same modulo operator. However, at the node side, the modulo operator is performed only once.

Alahmadi et al. in [4] proposed a novel approach to autonomously determine the transmission parameters of nodes. The network area is firstly partitioned into virtual annuli and assigned a specific SF for each annulus. Then, each annulus is further divided into a set of sectors. The number of sectors corresponds to the node density of a given annulus. Each node must be supplied with the (X,Y) coordinates of the associated gateway as well as its coordinates to determine the appropriate SF based on the distance between the gateway and the node. Subsequently, nodes determine their appropriate sector ID, which is its slot ID, by calculating its geometric angle to the gateway. Through this process, all transmission parameters are autonomously derived eliminating the need for additional packet exchanges with the gateway that could quickly deplete the duty cycle.

Moy and Besson in [83] proposed a distributed Reinforcement Learning mechanism that is implemented at the node side for channel allocation to mitigate the overhead that is resulting from the centralized approaches. In [83], the channel access problem is formulated as Multi-Armed Bandit (MAB) problem. LoRaWAN downlink traffic is exploited as feedback from the environment that will allow a node to determine its cumulative reward or success ratio. Nodes track the success ratio of each individual channel. Hence, they will select the channel with the highest packet success ratio. Although the algorithm is simple to implement, it requires acknowledging each uplink transmission in order to let the node decide whether the channel is congested or not which is possible when the server successfully receives the uplink transmission. However, due to the limited DC imposed on the downlink traffic, the server may not be able to acknowledge the reception of all uplink transmissions. Hence, nodes will have partial knowledge about the status of the channel.

Similarly, Azari and Cavdar in [84] proposed to use a distributed learning approach called multi-agent multi-arm bandit (MAB) to let nodes individually solve a maximization function that is a tradeoff between the reliability and the energy efficiency. This procedure allows each node to autonomously derive its SF, channel, and transmission power. Accordingly, they use a wider set of actions in order to update all the transmission parameters, unlike [83] where the algorithm aims at selecting only the transmission channel. The

calculation of a node's binary reward is contingent on whether it receives an acknowledgment from the server or not. In other words, if the ACK packet is received, the reward is set to one; otherwise, it is set to zero.

Sandoval et al. in [85] proposed a distributed transmission parameters configuration for nodes based on Markov Decision Process (MDP). In this model, the states of the MDP are defined by the remaining DC of the node at time t and the priority of the generated event. The action set is defined as pairs of Coding Rate (CR) and SF. The corresponding reward is the expected Packet Reception Rate (PRR), which is directly related to the selected action (selected pair of CR/SF). Higher CR or SF means higher PRR and hence higher rewards. Although the system is implemented individually on each node, the system did not model the collisions among the concurrent transmissions on the same channel with the same SF. In other words, although a node may get as a reward a high PRR since it selects high SF or CR, it may actually have a low PRR at the gateway due to collisions with concurrent transmissions on the same channel.

Liu et al. in [86] proposed a low complexity channel allocation algorithm by formulating the channel allocation as a joint optimization problem. They decompose the problem into two main phases, the channel allocation phase and the power allocation phase. The proposed channel allocation algorithm is based on many-to-one matching model. They assumed that users using the same channel would use different SFs to avoid collisions among them. Furthermore, they assumed that the channel statuses of all channels are known by all nodes, which are broadcasted in the downlink data timeslots at the end of each frame. Based on these channel statuses, a preference list is initiated for each node at the gateway. Then, each node selects the first available channel from that list. In the evaluation, each channel is assigned a maximum of six nodes that correspond to the available SFs, which is a very small number compared to the theoretical support limit of LoRa. Furthermore, the algorithm assumes fixed and static LoRa networks, which is impractical.

Kaburaki et al. in [87] proposed an autonomous Q-learning-based algorithm to avoid collisions for event-based traffic networks. When a node detects an event, it will calculate a time offset based on the Q-learning algorithm. Then, it will transmit that packet based on a calculated probability. The probability of sending a packet is calculated at the node side as a ratio between the total number of received Acks to the total number of transmitted packets. Hence, when an event happens, only a portion of nodes that have high success probability will be informed about it, leading to the lowering of the collision rate. Although the proposed scheme does not need any synchronization from the server, it assumes a fixed assignment of channels and

a fixed packet duration. Furthermore, the reward function mainly depends on the Ack that is sent by the server, which is an impractical solution with dense Duty-Cycled constrained networks since successfully received packets may be not acknowledged due to the duty cycle limitation.

Lasri et al. in [82] proposed that connected nodes autonomously control their traffic to mitigate collisions in high-density networks. To accomplish this, they operate under the assumption that the server possesses advance knowledge of the number of required transmissions per node within a pre-determined timeframe and each node initiates transmissions with a specified probability. During a given period, if the number of received packets from the node is lower than what is expected by the server, the server instructs the node to increase its probability through a downlink message. Similarly, an instruction to decrease the used probability is issued by the server when the number of transmitted packets from a node is higher than what is expected by the server. Hence, with a high number of connected nodes, the server might not update all nodes with the required uplink messages due to its limited DC. Furthermore, the proposed algorithm is not fully distributed, as nodes need periodical feedback from the server to adjust their transmission traffic.

Wu et al. in [88] proposed a distributed queuing-based algorithm for LoRa networks. The time is divided into consecutive beacon periods where each period is divided into three main intervals. The first interval is the Contention Window (CW), where nodes use S-ALOHA to compete for their transmissions. The second interval is optional in the frame, and it is a collision-free data slot. The third interval is used by the gateway to broadcast feedback to nodes. In the Contention Window (CW), nodes randomly pick a slot and transmit a small 2-symbol preamble. After the CW period, nodes receive feedback from the server about the status of the transmission during the CW. If a collision is detected during the CW, collided nodes enter the collision resolution queue and compete again in the next frame. If no collision occurs during the CW, the transmitting nodes enter the data transmission queue in the same order of the slots they choose during the CW to transmit their packets during the data transmission period. In other words, only one node accesses the medium and transmits its packet according to its order in the DTQ. The proposed protocol assumes Class B LoRa devices, where nodes listen to the medium after each CW to determine the status of their transmission. Table 8 summarizes the related works that address the Limited DC using distributed-based approaches.

Table 8: Distributed-based approaches to address the limited DC in LoRa networks.

Ref	Year	enhanced goal	methodology	Enhancement level	Notes
[81]	2019	PDR	Using node's MAC address to derive its timeslot ID.	Low	Not works well with hundreds of nodes.
[23]	2020	PDR	Using DevAddr and frame size to generate unique timeslot IDs	Medium	The average improvements in the PDR for the proposed algorithm compared to the ALOHA was 47%.
[4]	2022	Scalability	Use circle geometry to determine node's timeslot IDs	High	The average throughput was enhanced by 14x compared to the LoRaWAN.
[83]	2019	PDR	Use RL to let node autonomously select its channel	Low	Use only the three default channels.
[84]	2018	Energy efficiency	Use RL to let nodes individually select their parameters	High	The average improvements in the energy efficiency compared to the ALOHA was 79%.

[85]	2019	PDR	Use MDP to model node's transmissions	Medium	The Average Discounted Reward of the proposed algorithm compared to the theoretical limit was 18%. However, only 13 combination of (SF/CR) were considered.
[86]	2019	PDR	Formulating channel/power assignment as optimization problem using many-to-one matching	Medium	It was not evaluated with large-scale networks.
[87]	2021	PDR	Adopt Q-learning algorithm to offset the transmission time	High	PDR was enhanced by 66% compared to the ALOHA.
[82]	2023	PDR	Adjust node's traffic to mitigate collisions.	High	The average success probability of the proposed scheme was enhanced 64x compared to the LoRaWAN.

[88]	2020	PDR	Using Distributed Queue in LoRa networks	Dis- in net-	High	The average enhancements of the network throughput of the proposed algorithm was 2.5x better than the ALOHA.
------	------	-----	--	--------------	------	--

Discussion The preceding section demonstrates that three main methods were adopted to tackle the constrained duty cycle (DC) of downlink traffic, all involving distributed mechanisms. The first approach exploits some features in the nodes that will help in determining some of their transmission parameters [81] [23] without requiring extra packets from the gateway. In other words, using nodes' MAC addresses or device addresses to individually derive slotIDs for nodes. Although this approach is simple, it may generate large frame sizes and may not guarantee a unique slotID especially if nodes have heterogeneous manufacturers [81]. Furthermore, nodes must be aware of the frame size for their assigned SF, which must be broadcasted by the gateway at the end of each frame.

The second approach relies on the location information of both the associated gateway and the node itself to calculate its distance and angle to the gateway. Based on its distance, it will determine its SF and channel, and based on its angle, it will determine its slotID. Although this algorithm is simple and fully autonomous as it does not require any downlink traffic, it does not guarantee a unique timeslot for all nodes [4].

The third approach adopts some machine learning algorithms to be implemented individually at the node side [83] [84] [87] [82]. The main issue related to this approach, apart from the computational complexity imposed on the node side, is the requirement of instant feedback from the environment to help nodes make their decisions. The feedback is mainly sent by the server, which once again cannot update all the connected nodes, especially in high-density networks, due to its limited DC. Furthermore, the algorithms in this category must be evaluated in terms of their complexity and storage

constraints, as they are implemented in resource-limited nodes.

3.2.2 Aggregated Acknowledgments

An alternative method to address the gateway's constrained duty cycle involves employing the aggregated acknowledgment approach. In aggregated ACK, instead of acknowledging each individual node with a separate ACK message, the server aggregates a set of ACKs for a group of nodes in a single ACK. Consequently, by aggregating ACKs, the server may not reach its duty cycle limit before acknowledging or updating all connected nodes. Nonetheless, there is a need for more research to determine the optimal size and content of the aggregated message, as well as the required processing operations for the dedicated ACK packet at the node's side.

In this regards, Zorbas et al. in [23] combined the synchronization and the acknowledgments of nodes in one packet, which is transmitted at the end of each frame. The acknowledgment of node transmissions during the current frame is sent at the end of the frame. The server includes in the ACK packet a sequence of ones and zeros, where the length of this sequence corresponds to the number of assigned slots for nodes. A zero bit means unsuccessful reception of the packet that was transmitted in the corresponding timeslot position. Obviously, the size of the acknowledgment packet is proportional to the number of nodes that use a given SF and hence are assigned to a given SF frame. The server due to its limited DC may not be able to send the ACK packet especially if the number of nodes using a given SF is extremely high.

Lee et al. in [16] divided the time frame into an uplink transmission period (UTP) and a downlink transmission period (DTP). During DTP, the gateway acknowledges the nodes with an aggregated ACK packet that uses the same SF used by the nodes in their uplink transmissions. The aggregated ACK contains the number of devices to be acknowledged and the addresses of the acknowledged devices. Due to the maximum limit of packet sizes, there is a maximum number of acknowledged nodes per ACK packet. That number mainly depends on the used SF. In other words, ACK packets transmitted with larger SFs include a very small number of node addresses. With SF 10, only two nodes can be acknowledged in a given time frame. If a node does not receive its ACK, it will retransmit the packet in the next time frame. Obviously, this will limit the scalability of the network, as only a small number of nodes, 107 nodes, can be acknowledged per a time frame. Moreover, some successfully transmitted packets may be retransmitted just because the gateway wasn't able to send the ACK due to the time constraints of the DTP.

In [89], an aggregated acknowledgment is broadcasted periodically by

the server that includes an aggregated ACK of multiple users and multiple packets. To receive the aggregated ACKs, nodes open their receiving windows synchronously at a pre-configured time, for example, open a receiving window after each ten seconds regardless of the transmission time of the packet. Obviously, this will consume extra energy as nodes must wake up periodically and listen for a given period of time. Furthermore, issues related to the size of the aggregated ACK may arise when high number of connected nodes are deployed or high traffic rates are presented, as the aggregated ACK contains some bits for each packet and each node ID.

Lee et al. in [90] proposed an algorithm that supports the confirmed traffic by grouping the acknowledgment of multiple packets transmitted with the same SF. In [90], the server assigns nodes with different SFs to the same channel, however with different time slots. Nodes with the same SFs are assigned the same timeslots, however on different channels. In other words, nodes with the same SF have identical timeslots on different channels to synchronize their transmissions. The server simultaneously receives different packets with the same SF on different channels. Hence, the server aggregates the ACKs of the received packets and sends them in one channel. Obviously, the maximum parallel transmission corresponds to the number of available channels. Furthermore, the proposed algorithm did not clarify how the ACKs are aggregated and what is the maximum length of the ACK packet.

Yapar et al. in [64] used Naïve Aggregation (NA) method to aggregate multiple ACKs in one downlink traffic. In this approach, the frame is divided into a set of timeslots for uplink traffic followed by a single time slot dedicated for the downlink traffic. The aggregated ACK packet consists of a bit sequence of four bits that identifies nodes. Nodes inspect the received ACK packet that was received during the common DL period to check if it contains their IDs. Consequently, a maximum of 16 nodes can be acknowledged on each frame, which is a very limited number of nodes. Table 9 summarizes the related works that use aggregated ACK to overcome the limited DC of downlink traffic in LoRa networks.

Table 9: Aggregated ACK approaches to address the limited DC of downlink traffic in LoRa networks.

Ref	Year	enhanced goal	Methodology	Enhancement level	Notes

[81]	2020	PDR	Broadcast ACK at the end of the frame contains a sequence of 0s and 1s to acknowledge node's transmissions	Low	The SACK mechanism caused a roughly 50% increase in the total energy consumed.
[16]	2021	Scalability	Include device addresses in the ACK packet	Low	A maximum of 107 nodes can be acknowledged by a given frame.
[89]	2019	PDR	Aggregate ACKs for multiple users and multiple packets	Low	AggACK provides a thirty times higher data rate at maximum.
[90]	2018	PDR	Aggregate ACKs of different nodes that use the same SF.	High	The average improvements in the PDR with N=10k and 8 channels compared to the ALOHA was 227%.

[64]	2019	Scalability	Aggregate the subscription IDs of nodes in one Downlink packet	Medium	The average enhancements in the probability of successfully received packets with N=10k compared to the ALOHA was 25%.
------	------	-------------	--	--------	--

3.2.3 Hybrid Solutions

Another approach to mitigate the limited duty cycle challenge in LoRa networks is to integrate LoRa with other technologies that do not have duty cycle constraints. These blended solutions are referred to as hybrid approaches in this study. Different technologies have been combined with LoRa in literature. The majority of research in this domain involves combining LoRa with short-range technologies that are not bound by duty cycle limitations such as the Wake up Radio (WuR) [91]- [92], ZigBee [93], IEEE 802.11s [94], and FM radio [95]. They mainly adopt cluster topology, where nodes inside the cluster use a short-range technology. However, there is one study that combines LoRa with long-range non-DC-constrained, which is NB-IoT [96].

Various protocols utilize short-range wake-up technology to distribute the schedule among nodes, thereby circumventing the limitations of the duty cycle [91]- [92]. They use ultra-low power wake-up radios (WuR) for downlink communication with nodes. WuR can continuously monitor the wireless channel while consuming orders of magnitude less power compared to the radio hardware commonly used in wireless sensor platforms [97]. All protocols propose cluster-based TDMA protocols that adopt both the long-range LoRa technology and the short-range wake-up radios. The network in these protocols consists mainly of a sink (gateway) that supports only LoRa communication and a set of clusters. Each cluster comprises a Cluster Head (CH) and a set of nodes, all equipped to support LoRa communication for communication with the sink and short-range communication for intra-cluster

communication. The communication is always initiated by the gateway by sending a LoRa packet to the CH requesting data collection.

In both [91] and [98], the cluster head sends WuR message to nodes inside the cluster to wake them up. Then, each node selects a timeslot that corresponds to its ID for its LoRa uplink transmission. Additionally, In [98], if a node has no packet to send, it will send a LoRa packet back to the CH to inform it so it can assign its timeslot to another node. In this case, timeslots can be reused among nodes.

Unlike [91] and [98], in [99], the cluster head sends the nodes a short-range message containing the schedule of node's short-range packets. According to that schedule, if a node has a LoRa packet to the gateway, it will reply to the cluster head with another short-range packet using the previously set schedule. Hence, the cluster head can determine which nodes have LoRa packets to the sink to assign a timeslot for them in the next frame. The time frame and hence the waiting time for a node to transmit a LoRa packet for the sink is reduced since the frame accommodate timeslots for nodes that have packet for transmission. While the frame size is dynamic in [99], the slot duration is fixed based on the SF used by the nodes on a given cluster. All nodes of a given cluster use the same SF in [91] and [99]. However in [98], nodes of a given cluster use two different SFs such that nodes closer to the sink use the lower SF.

Unlike the previous studies, the CH in [100] supports LoRa Class B or C for long-range transmissions with the server, Gaussian Frequency Shift Keying (GFSK) and WuR for short-range in-cluster communications. Hence, the server can request periodic data by sending its request to the CH as LoRa command. The CH wake-up the targeted node using WuR beacon to send its LoRa packets to the server. Furthermore, nodes can send their periodic data immediately to the server without a prior request.

Instead of statically selecting a node to be a cluster head for each cluster, Djidi et al. in [92] proposed an opportunistic selection of cluster head by exploiting the receiving windows in LoRaWAN protocol. In thier approach,, once a gateway receives a LoRa packet from a node, it can send a request for a data packet from another node in one of the receive windows that is opened right after the transmission at the sender node. In this case, once a node receives such a request from the server, it will send that request to the targeted node using its WuR radio. That node in turn will send its data packets to the gateway using LoRa modulation. Hence, each node in the cluster after sending its LoRa packet can act as a CH for other vicinity nodes. However, unicast on-demand requests for data packets may be challenging with the limited DC at the gateway.

In fact, adopting this scheme requires that all nodes as well as cluster

heads to be equipped with WuR transceivers. Furthermore, using WuR transceivers inside the clusters imposes further challenges in terms of the range and the data rate, as WuR supports very small ranges, in the order of a few meters. The energy efficiency of [99] is not evaluated as nodes and the cluster head must have always-on wake-up radios to exchange short-range packets. Furthermore, the proposed protocols are evaluated only with one cluster that uses fixed SF and channel and contains a very small number of nodes. Moreover, the frame size in [91] and [98] is proportional to the number of nodes. Thus, with a large number of connected nodes, the protocol may have extremely large frame sizes.

Jiang et al. in [101] proposed a multi-hop mesh network, which combines long-range connectivity through LoRa with short-range connectivity using ANT to increase the scalability of the network. ANT is an ultra-low power short-range communication protocol. In their work, the network topology consists of a set of subnetworks, where each subnetwork consists of a LoRa node and a set of short-range nodes. Short-range nodes are organized within a subnetwork in hub-and-spoke mode, where one node in a listening mode collects data from other nodes on the network and forwards it to the LoRa node in the same subnetwork. The LoRa node inside a subnetwork forwards this data to other LoRa nodes to deliver it to the gateway. Hence, LoRa nodes are organized in a mesh topology while NAT nodes are organized in a star topology inside subnetworks. In order for a LoRa node to send its own packets or forward packets of neighbor subnetworks without collision, the transmissions are regulated using TDMA approach. Accordingly, each LoRa node builds a routing table and broadcasts it to other LoRa nodes in order for the LoRa hub nodes to build their schedule tables. Subsequently, each LoRa node either sends or receives or sleeps depending on their schedule. When LoRa nodes receive a schedule of a neighbor LoRa node, it registers the RSSI of the received packet so it can adjust the transmission power when transmitting to it. However, the proposed protocol assumes a single SF network. Moreover, the proposed algorithm did not specify how the dissemination of the routing table among LoRa nodes or data forwarding of other LoRa nodes are accomplished without violating the limited DC of LoRa networks.

Zhang et al. in [96] proposed a network architecture that combines LoRa technology with NB-IoT technology. They classify the connected nodes into main nodes and sub-nodes. Sub-nodes are equipped with LoRa technology only and they are responsible for collecting the data and sending it to the main node either using single-hop communication or multi-hop communication. Main nodes, on the other hand, are equipped with both LoRa and NB-IoT technologies and are responsible for sending their data as well as the

data of vicinity sub-nodes to the server using NB-IoT. In other words, the main node can be seen as a LoRa and NB-IoT gateway.

Another hybrid solution is proposed in [102], where a LoRa gateway is installed on top of a Low Earth Orbit (LEO) satellite. Nodes communicate with the gateway using SF12 to increase the scalability of the network. To avoid collision among nodes, they propose a scheduling algorithm that assigns slots to nodes based on the satellite coverage area and the visit time for each node. In other words, as the gateway moves to cover a new area, and hence new nodes, two strategies are used. The first one is called First Come First Served (FCFS), where the nodes located at the beginning of the gateway's covered area will transmit first. The second strategy is called fair strategy, where nodes that transmit fewer than others will have the priority to transmit, hence all nodes have a chance to transmit. However, disseminating the schedule among the nodes and updating the schedule is not discussed in the work, which is a critical metric in DC-constrained LoRa networks. Furthermore, Using SF12 consumes large ToA and hence higher energy consumption.

Davoli et al. in [94] proposed a hybrid mesh network that combines the IEEE 802.11s with LoRa networks to form a base communication for Unmanned Aerial Vehicles (UAVs). In their proposal, they tried to take advantage of both network technologies and avoid their limitations. In their proposal, the priority of the communication between the UAVs in a swarm and between the UAVs and the ground station is set to IEEE 802.11s since it is more reliable and provides higher data rates. However, if the communication is hard due to the limited range, the UAVs will use LoRaWAN to communicate directly to the ground station using the star topology. If the latter case is even harder to achieve, a UAV will broadcast its packet to the neighbor nodes in the swarm using a mesh time-slotted approach to avoid the collision between them. Each node performs a Listen Before Talk (LBT) at the beginning of each slot before proceeding with its transmission. All nodes use SF7 for their transmissions.

Truong et al. in [93] proposed to combine both the Zigbee and LoRa networks to increase the network scalability. According to their network architecture, there are two Zigbee clusters that are connected to a Zigbee-LoRa converter which in turn is connected to a LoRa gateway. They use the token ring protocol in the Zigbee clusters. Furthermore, the network architecture also has two LoRa clusters that are connected directly to the LoRa gateway. They use a polling mechanism in LoRa clusters where nodes send their packets upon a prior request from the gateway. According to their experimental results, the optimal distance that ensures a minimum packet loss rate between a Zigbee node and the ZigBee-to-LoRa converter is 630 m.

Furthermore, the optimal distance between a LoRa node and a gateway is 3.7 Km with SF7, CR=1, and a bandwidth of 125 KHz. However, they did not specify the size of each cluster. Moreover, allowing nodes to send their packets based only on the gateway request may limit the network throughput and increase the waiting time of nodes as the LoRa gateway has limited DC. Table 10 summarizes the related work that uses hybrid solutions to overcome the DC limit of LoRa networks.

Table 10: Hybrid solutions to overcome DC limit of LoRa networks.

Ref	Year	enhanced goal	Methodology	Enhancement level	Notes
[91]	2018	PDR	Use the WuR to deliver schedule to nodes	Low	The PDR was enhanced by 13% compared to the Listen Before Talk (LBT) mechanism.
[99]	2022	PDR	Use the WuR to deliver schedule to nodes	Low	No comparison with other protocols.
[98]	2022	PDR	use the WuR to deliver schedule to nodes	Low	9 nodes only used for performance evaluation
[92]	2021	Energy efficiency	Opportunistic selection of cluster head by exploit node's receive Windows.	High	They focus on reducing the downlink latency. With N=50 nodes the average latency proposed = 10000s LoRa=100000s Enhancements = 0.9

[100]	2018	Energy efficiency	Use Class B/C LoRa cluster head with WuR to trade-off the latency and the energy efficiency.	Medium	No comparison with other protocols.
[101]	2021	Scalability	Using LoRa mesh topology to connect short-range star networks.	Medium	13 LoRa nodes, distributed in a 1.1-km by 1.8-km area of Purdue University (in mode C).
[96]	2019	PDR	Equipped the gateway with LoRa and NB-IoT to communicate with nodes through LoRa and communicate with the server through NB-IoT	Low	Used small number of nodes. The maximum distance achieved with point-to-point communication was 1 km, while the maximum distance achieved with multi-hop communication was 1.6 km for harsh environments.

[102]	2022	Scalability	Mobile gateway installed on LEO	Low	No comparison with other protocols. The number of received packets per node when N=500 was 9 packets.
[94]	2021	Scalability	Use LoRa and IEEE 802.11s for communications in UAVs	High	Messages are received in range between 30 -60 Km with PRR = 95.83%.
[93]	2021	Scalability	Set up a ZigBee-LoRa based network	Medium	The Packet Loss Rate was < 0.5% when the communication rang of Zigbee network was 630 m and communication rang of LoRa network was 3.7 km.

3.3 Configuration of Node's Parameters

As explained earlier, LoRa physical layer supports different transmission parameters that greatly affect the performance of uplink communications. Among these parameters, the Spreading Factor (SF) which controls the data rate of the transmissions, Transmission power (TX) which controls the transmitted power of the packet, and the Channel Frequency (CF) that plays an

important role in determining the link quality between the nodes and the gateway. Different studies propose different methodologies to control these parameters in order to enhance the overall network performance. This section discusses the most recent studies related to the configuration of the transmission parameters at end nodes. The studies are further divided into studies that focus on the configuration of SF only (Section 3.3.1), SF in combination with TX (Section 3.3.2), and SF in combination with TX and CF (Section 3.3.3).

3.3.1 Spreading Factor Configuration

Ullah et al. in [103] proposed an algorithm to determine the annulus range of each SF using the k-means clustering algorithm. In their implementation, the algorithm starts by assigning the highest SF, SF12, to the nodes located at the outer side of the covered area. To do so, the algorithm calculates the range of the outer annulus. Then, all the nodes located on that annulus are assigned to SF12. Subsequently, the algorithm iterates through these steps five times, progressively determining the ranges for nodes with SF11 and other SFs until all SF ranges are defined.

Lim and Han in [104] proposed a model to estimate the range for each SF such that the overall Packet Delivery Ratio (PDR) is maximized. In the proposed model, each node selects an SF that meets two conditions: 1) the power received at the gateway of a transmission using a given SF must exceed that gateway's sensitivity level and 2) the Signal to Interference (SIR) of the signal received at the gateway must be above a certain level. In fact, the model sacrifices the first condition if a node fails to satisfy both, which is expected in dense networks. In their study, they found that if the second condition is met, the PDR will be higher than if the first condition is met. Therefore, even if the PDR is high, some nodes may have transmissions that are below the gateway's sensitivity level. This will increase the packet retransmission rates for those nodes, which will impact their energy efficiency.

Saluja et al. in [105] proposed a distance-based model to estimate the range of each SF, named windows, by defining an exponential factor a that defines the width of each window. The value of the exponential factor depends on the calculated Packet Success Probability (PSP) of nodes at the server side. Similar to [104], a node is assigned a given SF that must satisfy both conditions mentioned earlier. According to their analysis, when the value of a is less than one, more nodes are assigned to larger SFs, which is the case in high-density networks. On the other hand, when the value of a is greater than one, more nodes are assigned to smaller SF, which is the case in low-density networks. However, when a value equals one, all windows have

the same width, which equals the field radius divided by six. The proposed scheme provides high PSP with low-density networks. However, the PSP is less than 20% with high network density. Moreover, issues related to the distribution of the value of the window among nodes are not discussed, which is critical in the DC-constrained LoRa networks.

Marini et al. in [106] proposed a Collision Aware Adaptive Data Rate (CA-ADR) that adjusts node's SF such that the collision probability is minimized. In CA-ADR, instead of considering only the link quality in assigning a given SF for a node, similar to the ADR algorithm in LoRa WAN, it considers the packet success probability P for a node that uses a given SF by considering the set of nodes assigned the same SF. To do that, the algorithm first determines a maximum number of nodes to be assigned a given SF without compromising the collision probability. Then, the server computes the average of the received power of that k packets and selects the minimum SF that has a sensitivity lower than this value. If the number of nodes assigned to that SF does not reach the maximum, that SF will be assigned to the node. Otherwise, that node will be assigned the current $SF + 1$. If the algorithm reaches SF12 and there are some nodes that have not been assigned yet an SF, the algorithm runs again using a lower P . The algorithm needs to run every time whenever nodes are added to the network or when the topology changes. Furthermore, the proposed algorithm is executed centrally at the server side. Hence, the possibility of reaching the DC limit could happen before updating all connected nodes.

Zorbas et al. in [24] proposed a mathematical model that estimates the percentage of nodes that can use a given SF such that the packet delivery ratio meets a given threshold. To do that, they compute the average success probability for each set of nodes that use a given SF as a function of the node density. In their mathematical model, they consider both the collisions between transmissions of the same SF, named Intra-SF, and the collisions between the transmissions of different SFs, named Inter-SF. However, the model assumes that all nodes can select any SF ignoring the possible outage resulting when a transmission of a node with a specific SF might be received below the sensitivity level of the gateway.

Sandoval et al. in [25] proposed another mathematical model that works centrally at the receiver side and maximizes the average throughput of each node under a specific set of parameters. The maximization problem takes into account the Packet Reception Rate (PRR) and the collision probability of the transmissions under a given configuration. The server keeps a vector of each node, which contains the percentages of packets generated by that node under different configuration parameters. The algorithm considers a combination of SF and CR for each configuration. In other words, the algorithm changes only

the SF and the CR while fixing other transmission parameters to simplify the algorithm. Since the gateway needs to keep a vector for each node, creating a matrix with a number of rows equal to the number of connected nodes, the proposed algorithm might be applicable to small networks with a small number of nodes. Hence, the maximum number of nodes used in their evaluation is only 200 nodes.

Casals et al. in [26] proposed three different approaches to avoid the *SF12 well* issue in the Adaptive Data Rate (ADR) algorithm of LoRaWAN. The *SF12 well* issue happens when all nodes with ADR mode turned on eventually use SF12 for their transmissions. In the ADR algorithm, when a node sends a packet and does not receive an ACK from the gateway, it increases its SF gradually and hence decreases its data bitrate. This behavior may result in congestion as all nodes may end up using the same and the largest SF, SF12. To avoid that, they evaluate three different approaches. In the first approach, a single SF network is proposed, where all nodes use only SF7 and no change even if the ACK was not received. In the second approach, nodes use the ADR of LoRaWAN but if they reach SF12 and do not receive back ACK for two consecutive transmissions, it will reset to SF7 to mitigate any possible congestion. Similiar to the second approach, the third approach uses the ADR, however, when a node receives an ACK, it will decrease its SF. Although the proposed approaches provide higher PDR compared to the ADR algorithm, it is only applicable for small-scale areas as all nodes start initially using SF7 and may end up using it.

Soy in [107] implemented an autonomous and dynamic selection approach of SF parameter named blind ADR. In other words, nodes use the RSSI of the received beacons from the gateway to estimate its appropriate SF. In their implementation, the time is divided into frames in which at the beginning of each frame there is a reserved period to broadcast beacons by the gateway. Once a node receives a beacon, it will measure the RSSI of the received beacon. Based on the RSSI value, nodes change their SFs accordingly. The proposed algorithm targets mobile applications, as nodes can change their SFs at the beginning of each frame based on the RSSI of the received beacon. The proposed algorithm assumes a single channel with six different frames corresponding to the six available SFs. With 1000 nodes, the PDR is 70%. However, with a large number of connected nodes, more than 7000 nodes, the PDR was less than 20%.

3.3.2 Spreading Factor and Transmission Power Configuration

Abdelfadeel et al. in [108] introduced a method for achieving fairness in the distribution of Spreading Factors (SFs) among nodes, which was inspired

by a prior study [109] that proposed a ratio-based SF distribution method. However, they generalize the distribution ratio to include different Bandwidths (BW) and different Coding Rates (CRs). In their algorithm, the server calculates the percentage of nodes using a specific SF to ensure that each group of nodes utilizing that SF experiences an equal collision rate. Furthermore, they propose to balance the received power at the gateway side by tuning the node's transmission power to mitigate the near-far effect. Specifically, the server computes the recommended transmission power for each node such that the signal of closer nodes will not suppress the signal of farther nodes. Hence, the gateway can receive packets from both close and far nodes ultimately leading to the achievement of fairness among nodes.

Jeon and Jeong in [110] proposed a decentralized approach to adapt node's SF and TX taking into account the link quality of a given transmission configuration. In their algorithm, the server initially estimates the SF based on the received power of the *Join_Request* packet from the node and includes it in the *Join_Accept* packet that is transmitted back to the node. The node also estimates the initial transmission power from the received power of the *Join_Accept* packet. After that, each node maintains two counters that count the number of successive successful and failure packets using a given configuration. Nodes consider a failure transmission if they did not receive an ACK during the first receiving window and hence increase their failure counter. If the failure counter reaches a threshold, that node will increase its SF. On the other hand, if the number of received ACK reaches a given threshold, the SF for that node will be decreased. Although the proposed algorithm is simple to be implemented at the node side, the failure counter could be increased due to the limited DC by the server and not because the uplink packet was missing, which will result in an inaccurate update of node's configurations.

de Jesus et al. in [111] enhanced the performance of the Adaptive Data Rate (ADR) algorithm used by LoRaWAN. Specifically, they enhanced the accuracy of the link quality by dynamically changing the link margin parameter. In ADR algorithm of LoRaWAN, changing the SF and/or the TX depends mainly on the Received Signal Strength Indicator (RSSI) of the received packets. Hence, if there is an outage such that packets of a given node are not received because the received power was below the sensitivity level of the gateway due to a low TX or SF, those packets are not counted in the measured RSSI at the server. Hence, ADR algorithm does not accurately calculate the RSSI and thus the transmission parameters for nodes. They propose ADRx, that uses a dynamic link margin parameter instead of a fixed one that is used in the ADR of LoRaWAN. According to ADRx, all nodes start with the same margin parameter and once the server receives K pack-

ets, it will adjust the margin parameter by comparing the Data Extraction Rate (DER) of the K packets to a reference DER.

Li et al. in [27] proposed an algorithm that dynamically determines the SF and TX parameters for nodes in sporadic low-traffic networks such that the energy efficiency is maximized. According to [27], the gateway computes the Signal to Noise Ratio (SNR) for the last K packets received from a given node. Then, it uses this value to predict the most energy-efficient combination of SF/Tx for that node using a prediction model. At the node side, the node changes its SF/TX parameters as suggested by the server and tracks the connectivity to the server. If the number of missed ACK exceeds a specific threshold, it will reset their configuration to the maximum SF/TX. Due to the limited DC of the downlink traffic, nodes could end up tuning to the maximum SF/TX, especially in dense networks as the server may not be able to update all connected nodes.

Sandoval et al. in [112] extended the model proposed in [25] by bounding the set of eligible configurations of each node to include only the configurations that have minimum energy consumption. Furthermore, instead of mathematically calculating the PRR as a function of the SF and CR as in [25], they derive the PRR based on an estimated model of the current deployed environment. To do that, each node initially sends a predetermined number of packets under different configurations. Then, the server calculates the PDR for each node under each used configuration. Moreover, the server sets up an upper and lower bound for the expected PRR, hence it deduces the PRR of some configurations that were not used by a node based on the calculated PRR of some configurations used by the same node (past knowledge) (as if it is a supervised learning). Consequently, the number of required transmitted packets per node per configuration is minimized.

Djoudi et al. in [113] used Reinforcement Learning (RL) algorithms to find the optimal transmission parameters for nodes that maximize the data rates of uplink transmissions. To do that, three main stages are defined. Firstly, nodes randomly transmit multiple packets with different transmission parameters to build knowledge at the server side about the quality of each transmission configuration. Then, the transmission configurations are clustered based on their qualities into patterns. After that, they use these quality patterns to define the states of the Markov Decision Process (MDP) matrix. Simulation results show optimistic results regarding the PDR only with a small number of nodes (less than 3000). However, the PDR with a large number of nodes (< 8000) does not exceeds 30%.

Ilahi et al. in [114] used a centralized deep reinforcement learning mechanism to determine the appropriate combination of SF and TX for each node such that the PDR is maximized without compromising the energy efficiency.

In their work, they used Double Deep Q-Networks (DDQN) algorithm run at the server side and implemented it as a Markov Decision Process (MDP). The states represent the allocated actions and node's distances to the gateway. The action is the combination of SF and TX. The reward function considers the PDR, the time on air, and the power consumed by each action. Although the set of actions is assumed to be a combination of SF/TX, the evaluation process considers only a single transmission power. Moreover, they stated that the convergence time of the proposed algorithm is 200 khours, which is a very long time for convergence. Issues related to how to disseminate the configuration among nodes with the limited DC of gateways were not discussed.

Valach and Macko in [115] replaced the well-known LoRaWAN protocol in the link layer with another protocol named LoRa@FIIT [116] MAC protocol. LoRa@FIIT protocol has different features that overcome the drawbacks of LoRaWAN. Among these features, it has an optional acknowledgment, which means nodes are not obliged to open the two receiving windows after each transmission. Hence, it offers better energy efficiency compared to LoRaWAN. Furthermore, it uses shorter headers than LoRaWAN. Specifically, 12B of control data is needed to send 1B of payload, as opposed to LoRaWAN, where 29B is needed to send 1B of payload. Consequently, better usage of the node's limited duty cycle is achieved. Moreover, the node's transmissions are tagged with sequence numbers. Hence, the network server can identify if there are packets that are not received due to bad link quality. All these features come with the cost that LoRa@FIIT protocol does not support roaming. This means that the owner of the network must be the owner of LoRa sensors. This may limit the usability of such architecture.

Valach and Macko in [115] modified the centralized Adaptive Data Rate (ADR) algorithm in LoRa@FIIT protocol by replacing it with a Reinforcement-Learning algorithm that can work in centralized mode at the server side or in decentralized mode at the node side in case the acknowledgment was enabled. Specifically, instead of determining a specific combination of SF/TX per node like the ADR-LoRaWAN, each node has a set of combinations of SF/TX configuration that are associated with a specific reward. The configuration with a higher reward is more likely to be used by a node. The reward is updated at the node's side based on whether it receives an acknowledgment of its transmitted packets using a given configuration or not. They use the Upper Confidence Bound (UCB) algorithm at the node's side due to its simplicity. Moreover, the server also keeps track of the reward model using the sequence number of packets, the SNR, and the RSSI values. However, it updates nodes only when the environment or the link quality were changed. The simulation demonstrates that the suggested model increases the Packet

Delivery Ratio (PDR) by a factor of five when compared to the conventional ADR algorithm.

3.3.3 Spreading Factor, Transmission Power, and Channel Configuration

Reynders et al. in [117] proposed a beacon-based algorithm in which the gateway determines the set of eligible SFs and transmission powers for each possible channel. This information is carried out to nodes through beacons at the beginning of each frame. Once a node receives a beacon, it can synchronize itself with the gateway, and determine its transmission power and SF. In such a way, each channel has a specific set of SFs and transmission powers such that the capture effect is mitigated. After the beacon interval elapsed, nodes randomly transmit their uplink packets using ALOHA. Hence, collisions among uplink packets are not completely resolved.

Alahmadi et al. in [118] proposed a decentralized algorithm to distribute node's transmission parameters by considering the Duty Cycle (DC) limit of each channel. Nodes determine their appropriate set of tuples SF, TX, and CF based on their distances to the gateway. The proposed algorithm assumes that the DC per sub-band is distributed among the channels of that band. Hence, the DC per channel is affected by the number of channels for each band. Two operational modes are used, the Round-Robin approach and the Random approach. In the round-robin approach, nodes keep using a specific tuple of transmission parameters until no more DC on the channel for that tuple. In Random mode, a new tuple is randomly selected with each new transmission.

Abdelfadeel et al. in [69] proposed a distribution of spreading factors among channel frequencies based on the interference thresholds of each SF. In other words, spreading factors that have higher interference thresholds are isolated in dedicated channels from SFs that have lower interference thresholds. Therefore, the transmissions with lower SFs will not surpass the transmissions with higher SFs. Specifically, SF7, the smallest SF, has a higher interference threshold, hence it is allocated for a dedicated channel. Other SFs are assigned to channels such that small SFs, like SF8 or SF9, are combined with higher SFs, like SF11 or SF12, but with degrading their transmission power by one dBm to avoid the capture effect among the concurrent transmissions.

Table 11 summarized the related works that focus on enhancing the configuration of transmission parameters in LoRa networks.

Table 11: Enhancing the transmission parameters configurations of LoRa node's.

Ref	Year	enhanced goal	Methodology	Enhancement level	Notes
[103]	2019	PDR	Determine SF ranges using K-means clustering algorithm	Medium	The success probability was enhanced by 16% compared to the baseline model.
[106]	2021	PDR	Using packet success probability in assigning SFs to nodes.	Medium	With N=500 nodes, the proposed algorithm enhances the PDR by 1.33x compared to the ADR-LoRaWAN.
[104]	2021	PDR	Using packet success probability in assigning SFs to nodes.	Low	With N=2000, the average improvements in the PDR compared to the Equal Area-Based SF distribution was 29%.

[105]	2022	Scalability	Define SF windows using an exponential factor based on the PSP.	Medium	With N=10k nodes, the $PSP < 14\%$, while when N=2500 nodes, the $PSP = 50\%$
[24]	2018	PDR	Estimate the percentage of nodes that use each SF such that the PDR is maximized	Low	The Maximum number of nodes was 1000. 77% of nodes use SF7, while 33% use SF8.
[25]	2019	PDR	Maximize the average throughput for each node by estimating the percentage of usage for each parameter configuration	Medium	When N=200 nodes, the Proposed improvements was 50%.

[26]	2021	PDR	Propose three approaches to mitigate the SF12 well issue	Low	With N=100 nodes and the area is 142 m, the average PDR for Model1 was 90%, Mode2 was 62%, and Mode3 was 33%, where the average PDR for the ADR-LoRaWAN was 13%.
[107]	2023	PDR	Use the RSSI of beacons to estimate the appropriate SF	Medium	With 10K nodes, the average PDR for Blind ADR was 12%, while it was 4% for the ADR-LoRaWAN.
[108]	2018	PDR	Fairness distribution of SF/TX	Low	The Data Extraction Rate (DER) was 15% with N=4000 nodes.

[110]	2020	PDR	Adapt the SF/TX autonomously based on the estimating link quality of ACK	Medium	With 50 nodes, The PDR was 52%.
[111]	2021	PDR	Using dynamic link margin parameter for ADR	High	58% of nodes that used the ADRx achieved an ADR equals 80%, while 20% of nodes reach that ADR with the ADR-LoRaWAN.
[27]	2022	Energy efficiency	Predicting the SF/TX for nodes based on SNR of packets	Low	The average energy efficiency was improved by 41.2%.

[112]	2020	PDR	Use pounded model to estimate PRR based on the current used environment. The PRR use to estimate the optimal configuration of nodes	Medium	With 5 nodes in real testbed, the PRR almost equals 1 with the proposed algorithm, while it was 85% with the ADR-LoRaWAN. Hence, the average improvements was almost 18%.
[113]	2022	PDR	Use RL and MDP to maximize the DR of uplink traffic	Medium	the PDR with a large number of nodes (less than 8000) does not exceeds 30%.
[114]	2020	PDR	Use Double Deep Q-Networks (DDQN) algorithm at the server to estimate node's SF/TX	Medium	The model can achieve a PDR greater than 0.9 in a network containing 100 EDs in a single channel environment.

[115]	2022	PDR	Use LoRa@FIIT protocol with UCB algorithm instead of ADR algorithm	High	The average PDR of the proposed protocol was 80.85% compared to the ADR-LoRaWAN that was 11.91%
[117]	2018	Scalability	Provides two-step of selection for the transmission parameters	Low	With N=3500 nodes, the gain in the average Packet Error Rate was 20%.
[118]	2021	PDR	Determine optimal configurations based on node's distance to the GW. Distribute DC among the channels of a sub-band	High	The PDR with packet inter-arrival time= 16 min was 77%, compared to the ADR-LoRaWAN that was 15%.
[69]	2020	PDR	Distribute SFs among channels based on their interference thresholds	High	The Data Delivery Ratio (DDR) was almost 1 regardless of the network size.

3.4 Conclusion

This chapter reviewed the related works that have been proposed to mitigate or overcome the main limitations of LoRa networks. These limitations are the ALOHA medium access method, the limited duty cycle of the ISM band, and configuring LoRa nodes with optimal transmission configuration. Furthermore, this chapter focused on the research works that have been proposed at the MAC and Link LoRa network layers as it has a major role in controlling the interference and the collision rates. Furthermore, these two layers are open source for the research community unlike the physical layer which is a property of the LoRa alliance. In order to configure nodes with appropriate transmission parameters, the next chapter introduces an autonomous distribution of these parameters considering the limited duty cycle of LoRa nodes and the variant sensitivity levels of spreading factors.

4 Sensitivity-Aware Configurations for High Packet Generation Rate LoRa Networks

This chapter proposes Sensitivity-Aware LoRa Configuration (SAL) algorithm that autonomously allows nodes to select its transmission parameters in order to maximize the Packet Delivery Ratio (PDR). In other words, nodes independently determine the set of transmission parameters that ensure the transmitted packets using these parameters will not be received below the sensitivity level of the gateway. Assuming that each node knows its coordinates as well as the gateway ones, they can independently determine the optimal combination of Spreading Factors (SFs), Transmission Power (TX), and Channel Frequencies (CFs) based on their distance from the gateway. Furthermore, since the proposed algorithm uses all CFs from all ISM sub-bands and not just the CFs of the default sub-band, the proposed algorithm at least doubles the available Duty-Cycle (DC) by exploiting the channels of other sub-bands. To the best of the authors knowledge, this is the first LoRa algorithm that considers the distribution of (SF, TX, CF) taking into account the duty cycle of channel's sub-bands. Extensive simulation has been performed on OMNET++ [119] under FLoRa framework [120] showing promising results especially in terms of PDR and throughput as shown in section 4.3

The rest of the chapter is structured as follows: section 4.1 highlights the motivation and the problem statement. Section 4.2 describes in details the proposed algorithm. Section 4.3 summarizes the performance evaluation of SAL algorithm. Finally, section 4.4 concludes the chapter and provides insights about possible future research works.

4.1 Problem Statement

Indeed, according to LoRa, each node has a set of eligible spreading factors (SFs) where the transmission of packets using such factors is not received below the sensitivity of the gateway [8]. However, in the legacy LoRaWAN, nodes select randomly a channel (CF), a transmission power level (TX), and a spreading factor (SF) for each new transmission without considering whether the transmission using the selected SF will be received by the gateway. In fact, the number of available frequencies and transmission power levels depends on the region in which the LoRaWAN network is deployed. In Europe region, the assumed deployment region in this study, there are five transmission power levels, eight channels, and six SFs providing us with a search space of $(8CFs \times 6SFs \times 5TXs)$ or 240 different combinations for each node.

However, LoRaWAN uses three channels only by default leaving us with 90 options out of the 240 available. In addition, depending on the node's distance from the gateway, some of these options are not eligible to be used by a given node, as packets using these combination of transmission parameters might be received below the gateway's sensitivity. In other words, LoRaWAN nodes are not aware of whether the selected combination of (SF and TX) is eligible or not. Indeed, reducing the research space to include only the combination of transmission parameters (SFs, TXs) that are eligible according to node's distance to the gateway is vital to decrease the number of packets received below the sensitivity at the gateway. Consequently, enhancing the throughput as well as the energy efficiency. LoRaWAN uses a centralized approach called the Adaptive Data Rate (ADR) to choose a combination of (SF, TX) such that the Received Signal Strength Indicator (RSSI) of packets are above a pre-determined threshold. However, with a large number of connected nodes, the gateway may reach soon its Duty Cycle Limit (DCL) and cannot send control packets specifying nodes optimal transmission parameters. To overcome such an issue, this paper proposes a decentralized approach of determining node's transmission parameters without any need for gateway's control packets. The following section describes in details the proposed SAL algorithm.

4.2 The Proposed Solution

As mentioned earlier, a LoRa network has 240 different transmission parameters combinations with some that can worsen the network performance as they may result in packet transmissions below the sensitivity of the gateway. Thus, a more efficient approach is to have the network acts wisely and only considers a combination of parameters that is guaranteed to be received by the gateway. In order to specify the sensitivity level S for each SF, Eq.2 in section 2.3 is used, where the SNR values used in the equation are declared in Table 2.

In order for the gateway to successfully decode a packet, the received power of that packet must be higher than the receiver sensitivity for a given SF. Indeed, the received power depends on the transmission power and the path loss due to the signal attenuation and shadowing. In this study, we use the well-known log-distance path loss model with shadowing [121], which is used by different studies in LoRa [120] [122] [123][109] [108]. Eq.3 shows the path loss PL formula

$$PL(d) = \overline{PL}(d_0) + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma \quad (3)$$

where $\overline{PL}(d_0)$ is the mean path loss for distance d_0 , n is the path loss exponent, and X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ . The values of these variables are declared in table 15, which are the same values used in [124]. Knowing that the received power is the node's transmission power subtracted by the path loss PL , we can estimate the maximum distance d such that the resulting received power will be above the sensitivity level of the gateway. In fact, two transmission parameters affect the received power at the gateway, the transmission power parameter (TX) and the spreading factor parameter (SF). Table 12 shows the estimation of the maximum distance in meters in which it is eligible to use a given combination of SF and TX. For example, if the distance between a node and the gateway $d = 2500m$, then we have the following eligible set of transmission parameters (SF7,TX14), (SF8,TX11), (SF9,TX8), (SF10,TX5), (SF11,TX2), and (SF12,TX2). By using any of these combinations, the received power of the packet will be above the gateway sensitivity. Note that, we use the first maximum distance greater than the target distance in order to improve the energy efficiency of the algorithm by either reducing the time on air or the transmission power.

Table 12: Estimation of the maximum distance in meters per each (SF,TX)

SF \ TX	2	5	8	11	14
7	910	1225	1650	2220	2950
8	1220	1650	2200	2900	4000
9	1650	2200	2900	4000	5400
10	2200	2900	3900	5400	7300
11	2700	3600	4900	6600	8900
12	3300	4400	5900	8000	10800

According to LoRaWAN networks, node transmissions are regulated by defining a Duty Cycle Limit (DCL) for every channel per sub-band. DCL is the fraction of time per sub-band for which a node is allowed to transmit on channels of that sub-band. In Europe region [125], there are 2 sub-bands that are used for uplink transmissions of LoRa nodes, named g and g1 [126]. Each sub-band has a set of channel frequencies as listed in Table 13. According to Table 13, each node has a maximum of 1% DC for every sub-band, which corresponds to 36s of dwell time per hour. Dwell time is the time a node consumed for packet transmission. In other words, if a node consumes all

the DC on one channel of a sub-band, it cannot send any further packets on any other channel from the same sub-band [127]. In the proposed algorithm, we uniformly divide the DC of each sub-band on channels belonging to that sub-band, as described in the last column of Table 13. In other words, the time is divided into frames of 1 hour period. At the beginning of each frame, all channels will be resetting its DC to the maximum as shown in the last column of Table 13. By doing that, each channel has its own DC. Hence, whenever a node reaches its DC on a channel, it will not be blocked from using other channels of that band. This will enable the parallel transmissions on different channel frequencies. To the best of the authors knowledge, there is no research work that considers the duty-cycle per sub-band.

Table 13: Duty cycle per channels

Sub-band	Sub-band duty cycle	Channels (MHz)	Channel's duty cycle
g1	1%	868.1	0.33%
		868.3	0.33%
		868.5	0.33%
g	1%	867.1	0.20%
		867.3	0.20%
		867.5	0.20%
		867.7	0.20%
		867.9	0.20%

Regarding channels allocation, we assign at most two adjacent transmission power levels for a given channel to avoid the capture effect problem, which is a common challenge in LoRa networks due to the wide area coverage. In other words, all nodes that are within the same distance range and using the same channel will use either the same transmission power or at most two adjacent power levels such that the difference between the reception power of their transmitted packets on the same channel is less than 6 dBm. In other words, if two signals were received on the same channel at the gateway, the dominant one will be decoded, which is the signal with a received power greater than the received power of the other signal by at least 6 dB [128]. According to that, table 14 shows our distribution of the transmission power levels on different channels. By considering the channel frequencies CFs in the set of eligible transmission parameters, we almost double the number

of these configurations. However, the total number of eligible combinations of transmission parameters still very small compared to the actual research space. More importantly, the upper limit and the lower limit of the number of available combinations according to the node's distance is fixed regardless of node's distance to the gateway. This is because that each node will select the first pair of (SF and TX) from Table 12 such that the node distance to the gateway is less than the estimated distance in the table. By doing that, we minimize the Packet Error Rate (PER) as these combinations of parameters assure that the received power of the packets configured using one of these sets of parameters will be above the sensitivity level of the gateway. Hence, the worst-case scenario is when the node distance is around 2500m, as it will select the diagonal options, which are only six. Adding channels besides the pairs of (SF, TX) as demonstrated in Table 14 will double these options to be at maximum 12. This represents the upper limit of the eligible options for a node. For the best-case scenario, where node distance is less than the first entry on Table 12 ($d < 910m$) or greater than the last entry on the table ($d > 10800m$), there will be only 2 eligible options for them. Hence, the upper limit for these eligible options is 12 and the lower limit is 2. As a consequence, only very limited memory storage is required to store these combinations in each node.

Table 14: Distribution of the transmission power levels on different channels.

Channel ID	Channel Frequency (CF) MHz	Transmission power (TX) dBm
1	868.1	2, 5
2	868.3	8, 11
3	868.5	14
4	867.1	11
5	867.3	14
6	867.5	2
7	867.7	5
8	867.9	8

4.2.1 The Initialization Phase

In the proposed algorithm, we assume that each node can independently derive its Euclidean distance to the gateway by knowing its coordinates as well

as the gateway's coordinates. Once a node has determined its distance to the gateway, it can calculate the set of tuple transmission parameters (CF, TX, SF) that guarantee the successful reception of its packets at the gateway. According to our example, when the distance d equals 2500 m, for instance, the set of eligible transmission configuration will be as follows: (CF5, SF7, TX14), (CF3, SF7, TX14), (CF4, SF8, TX11), (CF2, SF8, TX11), (CF2, SF9, TX8), (CF8, SF9, TX8), (CF1, SF10, TX5), (CF7, SF10, TX5), (CF1, SF11, TX2), (CF6, SF11, TX2), (CF1, SF12, TX2), and (CF6, SF12, TX2). In fact, this example shows the upper limit of the number of available options that a node could have, which is 12. The lower limit of the number of available options is 2, which is the case when the distance is less than 910 m or greater than 10800 m. By doing that, we reduce the selection space of nodes from 240 options to a maximum of 12 options regardless of node's distance to the gateway. Once a node determines its transmission parameter options, it will sleep until it has a packet for transmission.

4.2.2 SAL Operational Modes

There are two operational modes in SAL that determine how a node selects a tuple of transmission parameters from the list that is created during the initialization phase, namely, the Round-Robin mode and the Random mode. In the Round-Robin mode, nodes initially select the first option in the list and keep using it until no more DCs are available on the channel of that option. In this case, it will select the second next option in the list, and continue until no more DCs are available on its channel, and so on. It is worth pointing out that the list is organized such that the options with the smallest SFs will be used first since the transmissions with small SFs will have less Time on Air (ToA). Hence, nodes will consume less of its DCs. In the Random mode, a node selects an option randomly from the list for each transmission regardless of whether more DCs are available. In other words, on each new transmission, a node selects a new option even if there are DCs available on the current selected channel. Consequently, we guarantee for each new transmission, nodes will select a new option even if the DC limit on the current selected channel is not yet reached. In both modes, if the DC limit on all the channels is reached, packets will be dropped until the beginning of the next frame (hour) where the DC is recharged and become at its maximum level. Algorithm 1 describes the Round-Robin approach of SAL algorithm.

Algorithm 1 Check DCL for current transmission options

```
1: Input: A new generated packet with  $\langle CF_i, TX_i, SF_i \rangle$ 
2:  $DC_{perCF} \leftarrow [DC_{CF1}, DC_{CF2}, DC_{CF3}, \dots, DC_{CF8}]$ 
3:  $ToA \leftarrow TimeonAir(PL, SF_i, CR_i)$ 
4: while  $ToA > DC_{perCF}[CF_i]$  do
5:    $i++$ 
6:   if  $i < TransOptions.size()$  then
7:      $CF_i \leftarrow TransOptions[i][0]$ 
8:      $TX_i \leftarrow TransOptions[i][1]$ 
9:      $SF_i \leftarrow TransOptions[i][2]$ 
10:  else
11:    break;
12:  end if
13: end while
14: if  $i < TransOptions.size()$  then
15:    $sendPacket()$ 
16:    $updateDCperCF()$ 
17: else
18:    $droppedPackets++$ 
19: end if
```

4.2.3 Data Transmission Phase

At the beginning of each frame (hour), the duty cycle of all channels is recharged. Once a node has a packet to transmit, it will select an option from the list it has created during the initialization phase based on the operational modes, that were discussed earlier, to configure the transmission parameters of the packet. Once a node finishes its transmission, it will enter DELAY1 period, similar to LoRaWAN, and update the DC of the used channel by subtracting the Time on Air (ToA) of the transmission from the DC of that channel.

4.3 Performance Evaluation

This section evaluates the two modes of SAL algorithm (Round-Robin and Random) in comparison with the LoRaWAN protocol using several metrics including Packet Error Rate (PER), Capture effect Probability, End-to-end delay, Packet delivery Ratio (PDR), Throughput, and Energy Consumption. SAL algorithm is implemented and evaluated in OMNET++ simulator [119] under FLoRa framework [120]. OMNET++ is an open-source,

extendable, component-based, C++ simulator that is primarily used for network simulations. OMNeT++ supports a model component architecture. Components (modules) are written in C++ and then integrated into larger components and models with the use of a high-level language named (NED). Model reusability is provided for free. OMNeT++ offers comprehensive GUI support, and the simulation kernel (and models) can be incorporated into a variety of applications thanks to its modular architecture. Furthermore, it is an open source and easy to modify as the network entities are organized in modules and submodules that are easy to modify by adding the needed parameters and/or methods. Moreover, the simulator can afford large-scale dense networks, where the number of active connected nodes could reach thousands without stop working. FloRa is an open source framework that implements LoRa network elements such as LoRa nodes, gateways, and network servers. Specifically, it implements LoRa physical layer and LoRaWAN MAC layer of LoRa nodes. It also includes a module to depict energy consumption of LoRa nodes. FLoRa framework provides flexible configurations for different physical transmission parameters such as SFs, transmission powers, coding rates, and bandwidths. Moreover, it supports bi-directional communication to exchange MAC commands between nodes and gateways and Duty cycle is implemented to regulate the access of nodes and gateways. Based on that, different research works used OMNET++ and FloRa framework for their simulation assessments [22] [129] [15] [26] [130] [67]. However, FloRa framework only supports the default sub-band of LoRaWAN protocol. In other words, since FLoRa framework uses only the default channels, we have modified the framework to support channels from all the available sub-bands and not just the default one. Specifically, our proposed protocols is mainly developed at the node's application layer of FLoRa framework. No required modifications were needed at the network server entity of FLoRa framework, as SAL algorithm, SBTS-LoRa protocol (Chapter 5), and ATS-LoRa protocol (chapter 6) are completely distributed. We suppose that simultaneous transmissions with different SFs are considered orthogonal, which is the same assumption implemented in FLoRa framework and used by different studies [120] [122] [123] [109].

Indeed, SAL algorithm is implemented in the application layer of LoRaWAN end nodes within the FloRa framework. A star network topology is assumed, which is the same network topology used for LoRaWAN protocol. Since SAL algorithm intended for large-scale networks, we evaluate the algorithm under a relatively large number of nodes (i.e., 1000 nodes) that are randomly distributed within a radius of 10 Km from the gateway and with each node generating a 20-byte packet with an exponential inter-arrival time. Table 15 Summarizes the used simulation parameters. Regarding the

procedure of collecting the results, I have modified a method in OMNET++ named *finish* that is automatically executed when the run elapsed such that all defined counters and variables that were defined to collect statistical data during the running of algorithms are written to external files. Examples of these parameters include the end-to-end delay, the number of received packets by the gateway, the number of send packets by nodes, the number of collided packets or packets that were received below the sensitivity level of the gateway, and the total energy consumed by nodes. After that, I have used MS Excel and Matlab to calculate the statistical metrics and draw the corresponding figures.

Table 15: Simulation parameters used to evaluate SAL algorithm.

Parameter	Value	Comments
CF	{868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, 867.9}	Carrier Frequencies (MHz)
SF	7 to 12	Spreading factors
TP	(2, 5, 8, 11, 14) dBm	Transmission powers
CR	4/5	Coding rate
BW	125kHz	Bandwidth
R	10 km	Field radius
N	1000	Number of nodes
$\overline{PL}(d_0)$	128.95	Mean path loss
d_0	1000 m	Base distance
n	2.32	Path loss exponent
X_σ	-2	Gaussian distributed random variable
Simulation time	5	Days

4.3.1 Packet Error Rate (PER)

The PER is the ratio of the total number of packets that is received under the gateway sensitivity to the total number of packets that is transmitted by end nodes. Fig. 9 shows the PER as function of the packet inter-arrival time. Both modes of the proposed algorithm achieve lower PER compared to LoRaWAN. Specifically, the average PER of SAL is only 8% compared

to that of LoRaWAN that stands at 83%. This can be attributed to the fact that SAL algorithm selects the combination of (CF, TX, SF) such that the estimated received power at the gateway will be above its sensitivity level. However, in LoRaWAN, nodes select randomly the combination of transmission parameters regardless of their distance to the gateway.

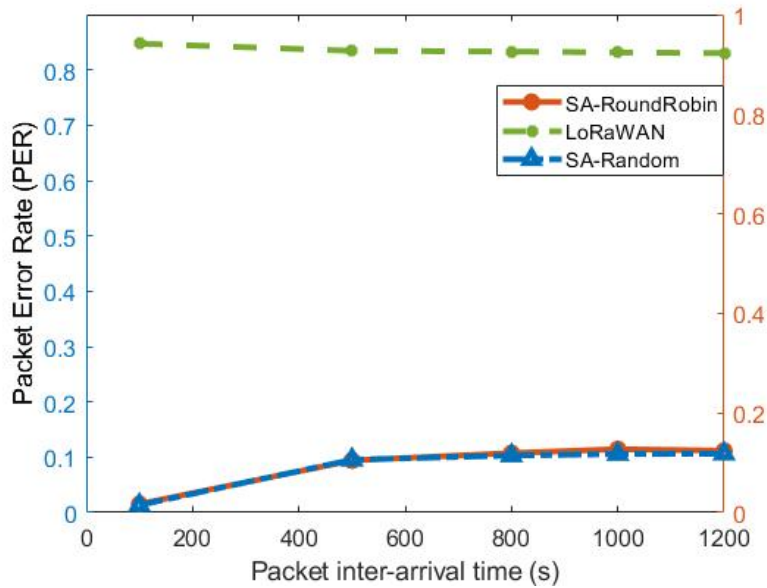


Figure 9: Packet Error Rate (PER) of SAL algorithm.

4.3.2 The Probability of the Capture Effect

Fig. 10 shows the probability of the capture effect as a function of the packet inter-arrival time. As shown in Fig. 10, SAL algorithm outperforms LoRaWAN, thanks to the wise distribution of transmission powers among the channels. In other words, nodes that use same channel will use similar transmission power levels as they are within the same distance from the gateway. Hence, their transmissions will be received nearly with the same power which will allow their successful decoding at the gateway. Unlike LoRaWAN, where a node can use a random transmission power on any channel regardless its distance to the gateway.

4.3.3 End-to-End Delay

Fig. 11 shows the end-to-end delay as a function of the packet inter-arrival time. As shown in Fig. 11, SAL-RoundRobin achieves the lowest delay since

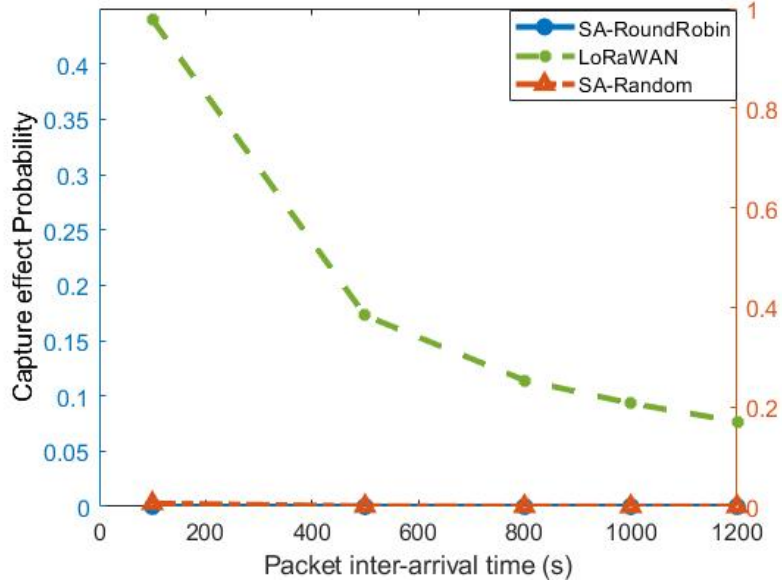


Figure 10: The probability of the capture effect of SAL algorithm.

it allows nodes to use firstly the smallest eligible SF in their options list and keep using it till no more DC on the associated frequency. In other words, nodes use firstly the smallest eligible SFs in their lists. On the other hand, LoRaWAN and SAL-Random have longer delay than the SAL-RoundRobin as they select their SFs randomly. As a result, they select randomly SFs for their transmissions and hence they may select more frequent larger SFs with higher ToA and hence higher delay. It is worth pointing out that, although SAL-Random achieves the highest end delay, especially with high packet generation rate, it achieves the highest Packet Delivery Rate (PDR) and throughput as demonstrated in the following sections.

4.3.4 Packet Delivery Ratio (PDR)

Packet Delivery Ratio (PDR) is the ratio of the number of successfully received packets at the network server to the total number of packets that is transmitted by end nodes. Fig. 12 shows the PDR as function of the packet inter-arrival time. Obviously, SAL algorithm shows a superior performance in terms of the PDR with 64% compared to that of LoRaWAN with a PDR of only 15%. Indeed, as shown in Fig. 10 and Fig. 9, LoRaWAN has higher packet error rate and higher capture effect ratio which explains its low PDR.

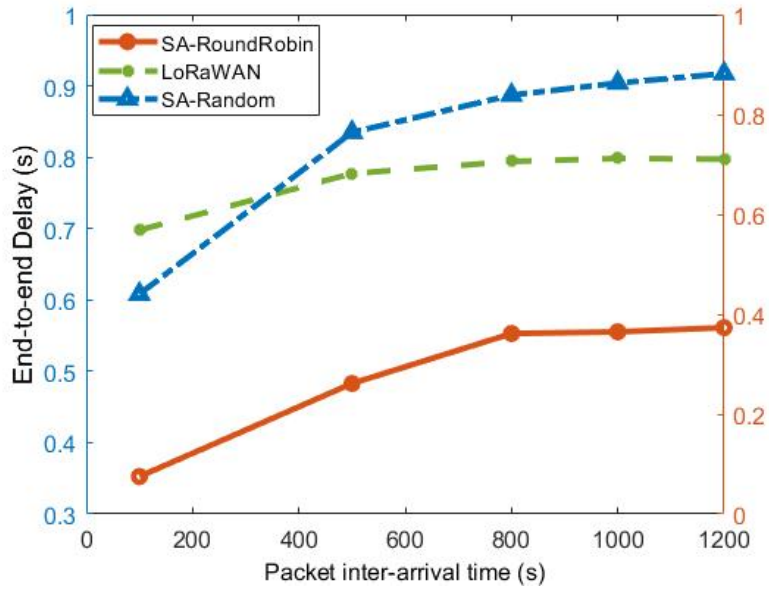


Figure 11: The end-to-end delay of SAL algorithm.

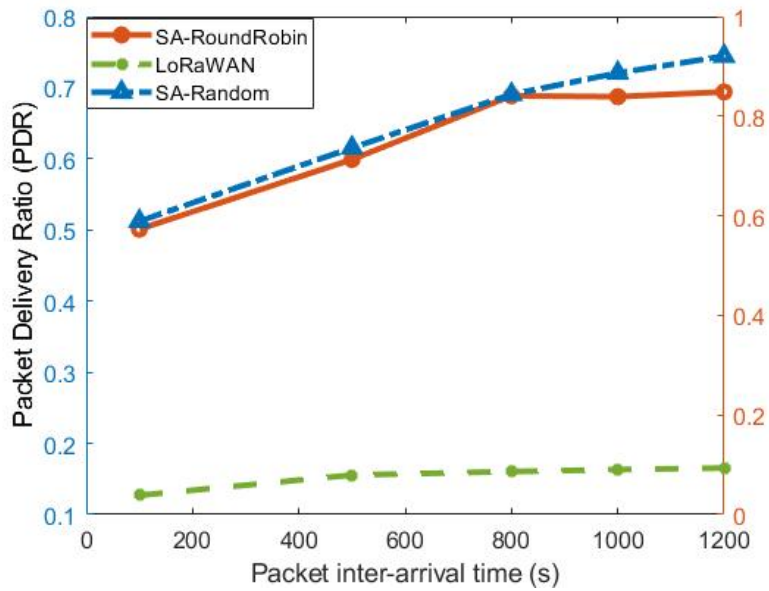


Figure 12: The Packet Delivery Ratio (PDR) of SAL algorithm.

4.3.5 Throughput

Fig. 13 shows the network throughput as function of the packet inter-arrival time. Firstly, it is worth pointing out that both operational modes of the SAL algorithm achieve higher throughput than LoRaWAN. More importantly, the throughput of the proposed algorithm is much higher than LoRaWAN especially when the network has a high packet generation rate thanks to the efficient distribution of channels, SFs, and transmission powers among nodes. In fact, with high packet generation rate, LoRaWAN performs the worse due to the high PER (Fig. 9) and high capture effect ratio(Fig. 10).

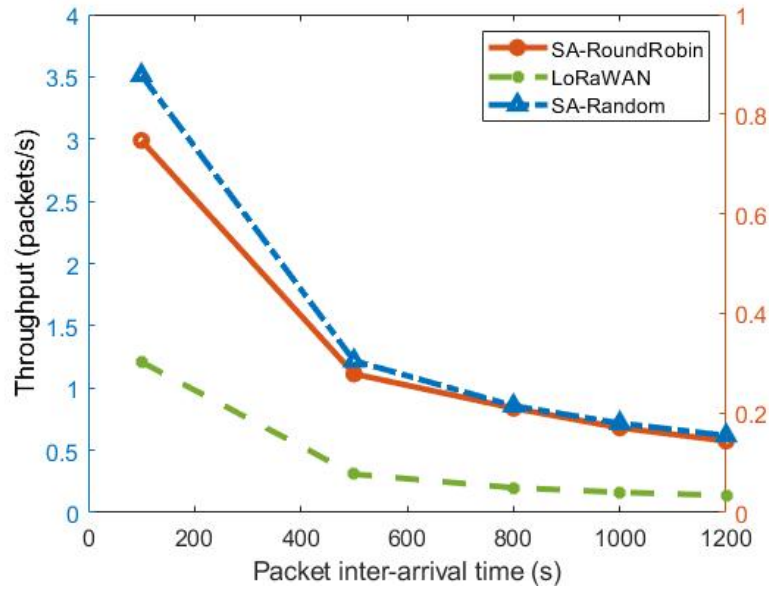


Figure 13: Network throughput of SAL algorithm.

4.3.6 Energy per Bit (EpB)

Fig. 14 shows the average energy consumed by nodes to successfully deliver one bit as a function of the average packet inter-arrival time. In general, LoRaWAN has the highest energy consumption since it has the highest PER and capture effect ratio. Specifically, when the network has high packet generation rate (i.e 1 per 100 seconds), LoRaWAN consumes more energy compared to the proposed algorithm.

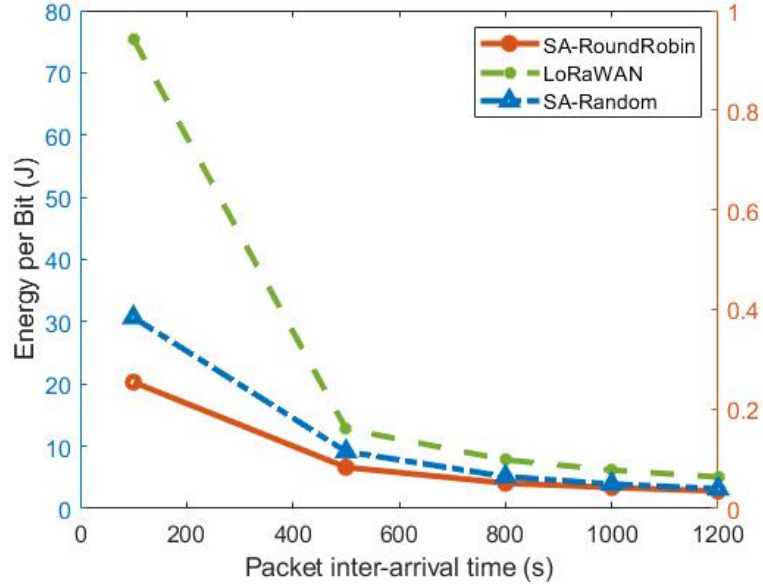


Figure 14: Energy consumed per bit in SAL algorithm.

4.4 Conclusion

This chapter proposed Sensitivity-Aware LoRa (SAL) algorithm that allows nodes to autonomously determine different combinations of transmission parameters such that the packet error rate (PER) is minimized. Furthermore, SAL algorithm exploits all channels, SFs and transmission power levels provided by LoRa physical layer to increase the number of parallel transmissions and hence increase the network throughput without violating the duty cycle. To fully taking advantage of the multichannel communication provided by LoRa, the duty cycle of sub-bands was distributed among the channels of each sub-band. To the best of the authors knowledge, no research work has been proposed that exploit the multichannel communication considering the duty cycle limitation. SAL algorithm provides a limited set of possible transmission parameter options regardless of node's context. Hence, only a small storage space is needed to store these options. The proposed algorithm was evaluated using extensive simulations that emulate the real environment. Simulation results show that the average PER was enhanced by an almost 90% compared to LoRaWAN. Hence, the average throughput using SAL algorithm was tripled compared to the average throughput of LoRaWAN. In order to address the ALOHA random channel access limitation besides finding appropriate configurations for nodes considering the limited duty cycle, the next chapter introduces a novel distributed Time Division Multiple

Access (TDMA) protocol named Sector-Based Time-Slotted (SBTS-LoRa) MAC protocol that addresses the main limitations of LoRa networks.

5 SBTS-LoRa: Sector-Based Time-Slotted MAC Protocol for LoRa Networks

In this chapter, Sector-Based Time-Slotted (SBTS-LoRa), a novel Time Division Multiple Access (TDMA) MAC protocol is proposed for LoRa networks. The aim is to address the main limitations of LoRa networks without compromising the energy efficiency. Besides the proposed protocol, a new comprehensive probability collision model is introduced that takes into account all possible events that could result in packet collisions at the receiver. The model is general and does not suppose a specific statistical distribution for the data generation rate or other factors. Most importantly, the model considers all supported transmission parameters of LoRa physical layer (Section 5.2). The second contribution of this chapter is the designing and deploying the SBTS-LoRa protocol as a novel MAC protocol to overcome key LoRa challenges such as the capture effect, the limited scalability, and the duty cycle. SBTS-LoRa replaces the ALOHA random access method with a TDMA access method. It allows each node to autonomously determine its transmission parameters by knowing only the gateway location and its own location. To do that, SBTS-LoRa firstly divides the network field into a set of annulus cells. Each cell will have a unique frequency, a set of eligible SFs, and a specific transmission power level. Nodes that are located on a specific cell boundary will use the transmission parameters assigned for that cell. Moreover, SBTS-LoRa divides each cell into a set of sectors. The sector ID to which a node belongs is the timeslot ID for which a node is allowed to transmit. The main novelty of SBTS-LoRa is the use of a decentralized approach that accurately and efficiently determine node's transmission parameters without burden the network with extensive control packets from the network server.

The third contribution is the conducting of extensive analysis and simulations using OMNET++ simulator under FLoRa framework to explore the performance of the SBTS-LoRa under different operating conditions [120]. In order to critically evaluate the proposed protocol, we implement EXPLoRa-AT [123] and [109] besides the Adaptive Data Rate (ADR) of LoRaWAN [8] protocol. We evaluate the performance of all protocols in large-scale extremely dense networks, where nodes are distributed in an area with a maximum distance of 14 km from the gateway and the number of connected nodes ranges from 1000-5000 nodes. To the best of our knowledge, no proposed protocol for LoRa networks is evaluated under these challenged conditions and considering all LoRa transmission parameters.

According to that, the chapter first introduces the main limitations of

LoRa networks addressed by the proposed protocol. Then, it describes in details the proposed SBTS-LoRa including the sectorization mechanism to autonomously determine node's timeslots. Finally, the chapter presents the results of the performance evaluation process of the proposed protocol.

5.1 Background and Problem Statement

As mentioned in chapter 1, LoRa physical layer provides variety of transmission parameters that highly affect the overall performance of the network. These parameters include the support of multi-channel communications, different Spreading Factors (SFs) that act as virtual channels, different transmission power levels (Tx), different channel bandwidth (BWs), and different Coding Rates (CRs). The supported number of channels, the transmission power levels, and the used bandwidths highly depend on the deployed region of the Industrial, Scientific and Medical (ISM) band. However, the number of supported Spreading factors (SFs) and the Coding Rates (CRs) are fixed for all regions. In fact, spreading factors determine the number of data bits that are modulated in each signal. For example, there are 10 data bits modulated in a signal that is transmitted with SF10. In fact, increasing the spreading factor will result in increasing the transmission time or the Time on Air (ToA) of a packet. Specifically, the data rate of a transmission with smaller SF is greater than the data rate of a transmission with larger SF. Besides controlling the data transmission rates, SFs control the sensitivity level of the receiver. Whenever a spreading factor of a signal increases, the receiver sensitivity of that signal is also increasing. As a result, the communication ranges are extended for signals with larger SFs. According to LoRa, there are six spreading factors ranges from SF7 to SF12. Signals with different SFs are considered orthogonal, which means that two or more packets transmitted on the same channel but with different spreading factors can be successfully decoded at the receiver. Consequently, we can think of spreading factors as if they are virtually partitioned each physical channel into six sub-channels with the same bandwidth. Obviously, this will expand the number of parallel successful transmissions.

In fact, exploiting the variety of transmission parameters by assigning different combinations of them to nodes will result in simultaneous collision-free transmissions, which will maximize the network throughput. Furthermore, by using diverse transmission parameters among the nodes, the scalability of the network is also maximized [8]. However, as explained previously, there are different data rates, and hence different ToAs, and different receiver sensitivity levels for each spreading factor, which will result in different transmission qualities. In other words, having just unique set of parameters for

each neighbor node is not the optimal distribution that will result in a high network throughput. The optimal distribution of transmission parameters is the distribution that maximizes the throughput without compromises the energy consumption. In order to achieve that, there are three main challenges of LoRa networks that need to be overcome as described in the following sections.

5.1.1 Scalability Issue

The main attractive feature that makes LoRa networks appropriate for IoT applications is the long-range coverage, where the gateway could receive packets from thousands of nodes that are far from it for up to 14 Km. However, long-range coverage is associated with the capture effect issue. Capture effect happens when two or more nodes located at different distances from the gateway transmit simultaneously on the same frequency. The gateway in this case will decode only the transmissions of the closer nodes, ignoring hence the transmissions from farther nodes. This is because the link budget of the closer nodes is much larger than the link budget of the farther ones. Hence, the transmissions of the closer nodes are dominant compared to the transmissions of the farther ones. In fact, the capture effect results from the use of uncontrolled transmission power levels. Thus, adopting some transmission power control technique as well as isolating far nodes from closer ones by assigning them different frequencies will inevitably mitigate such effect.

5.1.2 Collisions Due to ALOHA Access Method

According to LoRa physical layer, sensor nodes access the channel using ALOHA access method. In other words, once a node has a packet to transmit, it will wake-up and transmit it immediately without any carrier sensing or time regulations techniques. In fact, although there are multiple channels for transmissions in LoRa physical layer, nodes by default use the default channels only, which are three channels only in Europe band. Furthermore, as mentioned earlier, spreading factors are orthogonal, where transmissions with different SFs on the same channel might be successfully received at the gateway. However, inefficient distribution of SFs may result in having same SFs used by most of neighbor nodes. Consequently, with numerous connected nodes, collisions are not avoidable. Using some Time-Division Multiple Access (TDMA) techniques and having optimal distribution of SFs among the connected nodes are the key approaches in order to mitigate collisions.

5.1.3 The Duty Cycle Restrictions

Since LoRa networks use the unlicensed ISM band, duty cycle regulation is imposed on the unlicensed ISM band, which is considered a key limiting parameter [13]. Duty cycle is the fraction of time during which a node is allowed to transmit to the medium. For example, in Europe region, the maximum duty cycle of uplink transmissions of nodes is 1%, which means that nodes are allowed to transmit for only 36s per hour. In fact, this is highly affected by the used SF, as SFs control the Time on Air (ToA) of transmitted packets. For example, the ToA of a packet with 20 bytes payload that is transmitted in a channel with 125 kHz and a coding rate equals $4/5$ is ranging from 56.6ms to 1319 ms depending on the used SF [131]. In addition to that, duty cycle restrictions are also imposed on the downlink communication of the gateway. Accordingly, with large number of connected nodes, the gateway cannot acknowledge all nodes due to the duty cycle restrictions. Furthermore, in the previous section, we mentioned that replacing the ALOHA medium access method with TDMA method could reduce collisions and hence enhance the network throughput. However, the process of disseminating medium access schedules through downlink communications from the gateway may be disrupted because the gateway may reach its duty cycle limit before completing the schedule dissemination process.

As mentioned before, LoRa physical layer provides variant transmission parameters that if they are distributed and used in an efficient way, they could overcome most of LoRa network challenges. Hence, SBTS-LoRa is taking advantage of all transmission parameters to provide comprehensive protocol that eliminate collisions and hence increase the scalability of the network to the most possible extent. Accordingly, the research work in this chapter can be divided into two main stages: i) distributing all transmission parameters that affect the scalability of the network (section 5.3), ii) adding a TDMA layer on top of the proposed parameters distribution strategy, to allow collision-free transmissions for nodes with similar transmission parameters (section 5.4).

5.2 Collision Probability Model of LoRaWAN

In this section, we focus on deriving the probability of unsuccessful transmission attempt, $\beta(U, G)$, from a node U to the gateway G due to either channel error or collision in legacy LoRaWAN protocol. The main objective of this analysis is to mathematically assess the performance of LoRaWAN in order to define its weaknesses to help in proposing the right solution that aims at enhancing LoRaWAN performance. To the best of our knowledge,

there is no model that computes the probability of collisions using discrete events. Furthermore, the proposed model is more accurate as it considers all the transmission parameters of LoRa without simplifying the model with certain assumptions like assuming a specific distribution of nodes. Moreover, it considers all possible events that result in collision. To the best of our knowledge, no events have been neglected in order to simplify the model. The proposed model can be a starting point for future research work aiming at mathematically assessing the performance of LoRaWAN in order to optimally distribute the LoRa parameters in any random network topology.

According to LoRaWAN, nodes access the medium using ALOHA mechanism, where nodes randomly choose their spreading factors and their communication channel in order to proceed sending their messages at anytime without performing channel listening while respecting the duty cycle constraint. We calculate $\beta(U, G)$ by mimicking the real case scenario to the most possible extent. To do so, we determine the different events that arbitrate the LoRaWAN channel access. Accordingly, a transmission from node U is unsuccessfully received by the gateway G if one or more of the following events occur:

- A: a packet error occurs during the transmission on the wireless link (U, G)
- B: one or more sensor nodes within the transmission ranges of G transmit at the same time as node U in the same channel using the same spreading code.
- C: node U transmits while G is busy with a transmission from a neighbor node. For example, node U uses a given spreading factor SF_U . There is a node N within the transmission range of G . Node U will access during node N transmission using the same spreading factor as node N and the same channel as node N .
- D: the gateway G receives transmissions from another node while node U transmission is still in progress. Typically, if during the packet transfer from U to G in a given channel and using a given spreading factor, node N transmits to G using the same channel and the same spreading factor as U .

Fig. 15 shows a demonstrative example of the events.

Assuming the independence of the aforementioned events, the probability of successful transmission from node U to the gateway G can be expressed as follows

$$1 - \beta(U, G) = (1 - Pr\{A\}) (1 - Pr\{B\}) (1 - Pr\{C\}) (1 - Pr\{D\}) \quad (4)$$

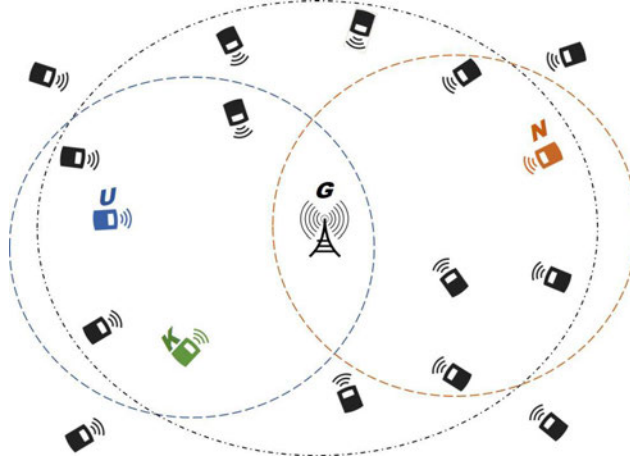


Figure 15: Probability collision model of LoRaWAN

where the $Pr\{A\} = l(U, G)$ is the packet error rate on link (U, G) . In what follows, we provide a detailed description on how to calculate the probabilities of events B , C , and D . To do so, let us discretize the time into so small slots of time.

5.2.1 Calculating $Pr\{B\}$

Theorem 1. *For every node U , the probability $Pr\{B\}$ that one or more nodes in $H(G) \setminus \{U\}$ transmit in the same slot as node U , in the same channel and using the same spreading factor is given by*

$$Pr\{B\} = 1 - \prod_{k \in H(G) \setminus \{U\}} \left(1 - \frac{1}{N_{ch}} \times \frac{1}{N_{sf}} \times \frac{N_s}{N_{st}} \times \rho_K\right) \quad (5)$$

To describe $Pr\{B\}$, we have defined the following events:

$$\begin{aligned} E &= \{ \text{no nodes in } H(G) \text{ accesses in} \\ &\quad \text{a given slot using a given channel and a given} \\ &\quad \text{spreading factor} \} \\ F &= \{ \text{node } U \text{ accesses the same} \\ &\quad \text{channel using the same SF in that slot} \} \end{aligned} \quad (6)$$

Then we have,

$$Pr\{B\} = 1 - Pr\{E|F\} \quad (7)$$

Let us consider sensor node $K \in H(G) \setminus \{U\}$. According to that, we can define the following events:

$$\begin{aligned}
X &= \{ \text{node } K \text{ accesses a given channel using a given } SF \\
&\quad \text{in a given slot} \} \\
Y &= \{ \text{node } U \text{ accesses that channel using} \\
&\quad \text{that } SF \text{ in that slot} \} \\
Z &= \{ K \text{ has a packet to transmit} \} \\
Q &= \{ \text{node } K \text{ is in active state} \}
\end{aligned} \tag{8}$$

Then we have,

$$\begin{aligned}
Pr\{ X | Y \} &= Pr\{ X, Z, Q | Y \} \\
&= Pr\{ X | Q \} \times Pr\{ Q | Z, Y \} \times Pr\{ Z \} \\
&= Pr\{ X | Q \} \times Pr\{ Q | Z \} \times Pr\{ Z \}
\end{aligned} \tag{9}$$

$$Pr\{ Z \} = \min(1, \lambda_K \times E[T_K]) = \rho_K \tag{10}$$

$E[T_K]$ can be defined as the time on air (ToA) of a packet from node K to G . In fact, computing the average value of $E[T_K]$ necessitates the estimation of the SFs distribution in the network according to LoRaWAN protocol. This distribution depends both on the network topology (nodes' positions) and mostly the network density. That being said, in dense network, regardless the nodes' positions, most of LoRaWAN nodes will end up choosing SF_{12} as they will not receive the acknowledgment from the gateway due to duty cycle limitations. For this reason, and since the time on air depends on the used SF, the authors opt for choosing SF_{12} to calculate the $E[T_K]$ as follows [3]:

$$E[T_K] = T_{oA_{12}} = \frac{2^{12}}{BW} \times \frac{PL}{12} \tag{11}$$

where $T_{oA_{12}}$ is the Time on Air (ToA) of a packet of payload length PL in bytes using the spreading factor SF_{12} . BW is the channel bandwidth. It is worth pointing out that $\frac{2^{12}}{BW}$ is the symbol duration T_{Sym} , which is the time taken to send 2^{12} chips at a chip rate equivalent to BW .

Recall that the total time on air for every sensor is limited by the duty cycle restrictions. Thus:

$$\lambda_K \times E[T_K] \leq \frac{N_s}{N_s t} < 1 \tag{12}$$

$$Pr\{Q|Z\} = \psi = \frac{N_s}{N_{st}} \quad (13)$$

$$Pr\{X|Q\} = \frac{1}{N_{ch}} \times \frac{1}{N_{sf}} \quad (14)$$

It is worth pointing out that X , Y , Z and Q are granular events that have been defined in order to meticulously calculate $Pr\{B\}$. Regarding X and Z , Z denotes the activity ratio of a node which is limited by the packet generation rate λ_K . Moreover, X can happen only if Z and Q occur. In other words, a node K can only access the channel using a given SF only if it has a packet to be transmitted and is allowed to access according to the duty cycle. In other words, if a node K has a packet to be transmitted but its duty cycle has been consumed; then, node K can not access the channel. Consequently, in order for node K to access the channel, two conditions have to be satisfied, node K has a packet in its queue and node K is active (its duty cycle has not been consumed yet).

5.2.2 Calculating $Pr\{C\}$

The probability $Pr\{C\}$ that the gateway G is already busy with a transmission on a given channel with a given spreading factor when it receives a transmission from U on the same channel using the same spreading factor can be expressed as follows

$$\begin{aligned} Pr\{C\} &= Pr\{G \text{ is already busy on a given channel} \\ &\quad \text{with a given SF} \mid \text{node } U \text{ accesses that channel} \\ &\quad \text{to transmit to } G \text{ using that SF}\} \\ &= Pr\{G \text{ is busy on a given channel only due} \\ &\quad \text{to nodes other than } U \text{ using a given SF}\} \\ &= Pr\{G \text{ is busy on a given channel } Ch_U \text{ due} \\ &\quad \text{to nodes other than } U \text{ that use a given } SF_U \\ &\quad \mid G \text{ is busy}\} \times Pr\{G \text{ is busy}\} \end{aligned} \quad (15)$$

The first element can be written as follows

$$\begin{aligned} &Pr\{G \text{ is busy on a given channel } Ch_U \text{ due to} \\ &\quad \text{nodes other than } U \text{ that use a given} \\ &\quad SF_U \mid G \text{ is busy}\} = \end{aligned}$$

$$\left[\frac{1 - \prod_{K \in H(G) \setminus \{U\}} (1 - \omega_K)}{1 - \prod_{K \in H(G)} (1 - \omega_K)} \right] \quad (16)$$

$$\begin{aligned} \omega_K &= Pr\{X|Z\} \times Pr\{Z\} \\ &= Pr\{X, Q|Z\} \times Pr\{Z\} \\ &= Pr\{X|Q, Z\} \times Pr\{Q|Z\} \times Pr\{Z\} \\ &= \frac{1}{N_{ch}} \times \frac{1}{N_{sf}} \times \psi \times \rho_K \end{aligned} \quad (17)$$

The second element can be expressed as follows

$$\begin{aligned} Pr\{G \text{ is busy}\} &= 1 - Pr\{G \text{ is not busy}\} \\ &= 1 - \prod_{K \in H(G)} \delta(G, K) \end{aligned} \quad (18)$$

$$\begin{aligned} \delta(G, K) &= Pr\{G \text{ is not occupied by a transmission} \\ &\quad \text{from } K\} \\ &= 1 - Pr\{G \text{ is occupied by a transmission} \\ &\quad \text{from } K\} \\ &= Pr\{G \text{ is occupied by a transmission from } K \\ &\quad | K \text{ is active}\} \times Pr\{K \text{ is active}\} \\ &= \psi \times (\lambda_K \times \bar{N}_C(K, G) + 1) \times ToA_{12} \\ &\quad + \lambda_K \times T_{ACK} \\ &= \psi \times \lambda_K [(\bar{N}_C(K, G) + 1) \times ToA_{12} + T_{ACK}] \\ \delta(G, K) &= 1 - \\ &\quad [\psi \times \lambda_K [(\bar{N}_C(K, G) + 1) \times ToA_{12} + T_{ACK}]] \end{aligned} \quad (19)$$

$$\bar{N}_C(K, G) = \frac{\beta(K, G)}{1 - \beta(K, G)} \quad (20)$$

$\bar{N}_C(K, G)$ can be derived as follows: $N_C(K, G)$ be a random variable representing the number of unsuccessful transmissions experienced by a given data message from K to G . If $\beta(K, G)$ is the probability that a transmission a tempt from K to G is failed. Then, $N_C(K, G)$ is a geometric random variable and thus we have

$$\bar{N}_C(K, G) = \frac{\beta(K, G)}{1 - \beta(K, G)}$$

where $\beta(K, G)$ is the probability of collision on link (K, G) . Finally, substituting Eq.16 and Eq.18 into Eq.15, we get expression of $Pr\{C\}$.

5.2.3 Calculating $Pr\{D\}$

We firstly define the vulnerability period T_v as follows:

$$T_v = \frac{ToA_{12} + D}{Slot} \quad (21)$$

T_v is the vulnerability period in terms of slots during which the transmission of a node within the range of G prevents the success of the in-progress transmission from U to G . Hence, the probability $Pr\{D\}$ that the gateway G receives transmission from other node than U while the transmission of node U is still in progress can be derived as follows:

$$\begin{aligned} Pr\{D\} &= 1 - Pr\{ \text{no node in } H(G) \setminus \{U\} \text{ transmits} \\ &\quad \text{during } T_v \text{ slots} \} \\ &= 1 - \prod_{K \in H(G) \setminus \{U\}} (1 - \omega_K)^{T_v} \end{aligned} \quad (22)$$

Finally, substituting Eq.5, Eq.15 and Eq.22 into Eq.4, we obtain the probability of collision $\beta(U, G)$. Table 16 represents the notations and description used in this section.

5.3 Annulus-Based Distribution of LoRa Transmission Parameters

In this section, an optimal distribution of all transmission parameters that affect the scalability of the network is described and outlined in Algorithm 2. The algorithm is executed during the initialization phase of the newly joined node. The algorithm takes the node coordinates (X_n, Y_n) and the gateway coordinates (X_G, Y_G) as input and returns the cell identifier $cell$, the selected channel frequency CF_{node} , the chosen transmission power level TX_{node} , and the selected spreading factor SF_{node} as outputs. It also defines three different vectors (line 3-5). The SFs vector holds all supported spreading factors, which are six, the CFs vector stores six different channel frequencies, and the TXs vector that holds six different power transmission levels. The values of both SFs and TXs vectors are listed in ascending order. According to LoRa networks, the network topology is one-hop star topology. We assume that each node knows its coordinates and the gateway's coordinates. As mentioned before, LoRa physical layer supports multiple transmission

Table 16: Notations and descriptions.

Notations	Description
$\beta(U, G)$	The probability of unsuccessful transmission from a node U to the gateway G .
$H(G)$	All the nodes within the transmission range of the gateway. G
N_{ch}	Number of available channels.
N_{sf}	Number of spreading factors.
N_s	Number of active slots per unit of time .
N_{st}	Total number of slots per unit of time.
ρ_K	The utilization of node K .
$E[T_K]$	The average transmission time from node K to the gateway G .
λ_K	the traffic rate from node K to the gateway G .
ToA_{SF}	The Time on Air (ToA) of a packet using the spreading factor SF .
BW	The channel bandwidth.
PL	The payload length.
ψ	Node duty cycle.
ω_K	The probability that node K transmits on the medium in a given slot in a given channel using a given SF .
$\delta(G, K)$	The probability that G is not occupied by a transmission from K .
$\bar{N}_C(K, G)$	The average number of unsuccessful transmissions from K to G before being successfully received.
D	The maximum propagation time.
T_v	The vulnerability period.

parameters that greatly affect the transmission quality. Among them, the Spreading Factor (SF) and the transmission power (Tx) are the most influencing transmission parameters that directly affect the transmission range and the bite rate. In addition, efficient utilization of the multi-channel communication that is provided by the physical layer will greatly enhance the throughput and the scalability of the network. In order to efficiently distribute these transmission parameters among the nodes, we partition our network into six equal sized annulus cells as shown in fig. 16 such that for a given $cell_i$, the radius is ir , where r is the field radius, R , divided by six (line 8). Note that our network partitioning into six cells is closely related to the total number of available SFs, which is six, as it will be explained later. The following equation defines the annulus width according to the field radius R .

$$r = \frac{R}{6} \quad (23)$$

According to the proposed algorithm, each cell is assigned one unique channel with a specific transmission power (line 11-13). By doing that, we split the nodes located far from the gateway and the node close to the gateway into different channels. Furthermore, all nodes that use the same channel will be relatively close to each other and hence they may use the same transmission power. By doing that, we will mitigate collisions resulting from the capture effect [128], which is considered as one of the main concerns of LoRa networks. Regarding the distribution of SFs, each cell is assigned a set of eligible SFs. The eligible SFs of a node is a set of SFs that can be used by a node in a transmission such that it can be successfully received and decoded by the gateway. As mentioned earlier, LoRa networks support six different SFs from SF7 to SF12. Smaller SFs have smaller transmission ranges, so it can be only used with close nodes to the gateway. In other words, the number of eligible SFs per cell depends on how close the cell is to the gateway. For example, $cell_1$, the closest cell, supports all SFs while $cell_6$, the farthest cell, supports only SF12, as the transmissions of nodes on $cell_6$ will not be successfully received if they use smaller SFs because it will be received below the sensitivity threshold of the gateway. Table 17 shows the list of eligible SFs for every cell.

In order for a node to specify its cell and since we assume that each node knows its coordinates and the gateway's ones, each node first calculates the euclidean distance d_{NG} between the node n and the gateway G (line 6). Once done, it will calculate the radius of each $cell_i$ according to the annulus width r as follows:

$$cellRadius_i = i \times r \quad (24)$$

where i refers to the cell identifier.

Algorithm 2 SBTS-LoRa MAC Protocol

- 1: **Input:** node coordinates (X_n, Y_n), Gateway coordinates (X_G, Y_G), node density d , and p
- 2: **Output:** $cell, CF_{node}, TX_{node}, SF_{node}$, and slotID S_n
- 3: $SFs \leftarrow [7, 8, 9, 10, 11, 12]$
- 4: $CFs \leftarrow [cf_1, cf_2, \dots, cf_6]$
- 5: $TXs \leftarrow [Tx_1, Tx_2, \dots, Tx_6]$
- 6: $D_{NG} \leftarrow calculateEuclideanDistance(X_n, Y_n, X_G, Y_G)$
- 7: # r is the annulus width, R is the field radius
- 8: $r \leftarrow R/6$
- 9: **for** $i \leftarrow 1$ to 6 **do**
- 10: **if** ($D_{NG} > (i - 1) \cdot r$) & ($D_{NG} \leq (i \cdot r)$) **then**
- 11: $cell \leftarrow i$
- 12: $CF_{node} \leftarrow CFs[i - 1]$
- 13: $TX_{node} \leftarrow TXs[i - 1]$
- 14: $ns \leftarrow 6 - (cell - 1)$
- 15: $sr \leftarrow r/ns$
- 16: **for** $k \leftarrow 1$ to ns **do**
- 17: $min \leftarrow ((k - 1) \cdot sr) + ((cell - 1) \cdot r)$
- 18: $max \leftarrow ((k) \cdot sr) + ((cell - 1) \cdot r)$
- 19: **if** ($D_{NG} > min$) & ($D_{NG} \leq max$) **then**
- 20: $subCell \leftarrow k$
- 21: $SF_{node} \leftarrow 7 + (cell - 1) + (k - 1)$
- 22: **end if**
- 23: **end for**
- 24: **end if**
- 25: **end for**
- 26: $temp \leftarrow cell \cdot r - (r/(6 - (cell - 1)))$
- 27: $\alpha_{cell_i} \leftarrow 2 \cdot p/d \cdot ((cell \cdot r)^2 - temp^2)$ ▷ Eq.27
- 28: $m_{cell_i} \leftarrow 2 \cdot \pi/\alpha_{cell_i}$
- 29: $\theta_n \leftarrow \arctan(Y_n - Y_G/X_n - X_G)$
- 30: $S_n \leftarrow \theta_n/\alpha_{cell_i}$

Table 17: Transmission parameters distribution among cells.

Cell ID	Channel ID	Tx Power	Eligible SFs
Cell 1	Channel 1	TX1	SF7 - SF12
Cell 2	Channel 2	TX2	SF8 - SF12
Cell 3	Channel 3	TX3	SF9 - SF12
Cell 4	Channel 4	TX4	SF10 - SF12
Cell 5	Channel 5	TX5	SF11 - SF12
Cell 6	Channel 6	TX6	SF12

After defining the boundaries of each cell according to the field radius R , each node iteratively calculates its cell identifier (line 9-10). In other words, a node will be considered on $cell_1$ if its distance to the gateway is less than the radius of $cell_1$. In general, a node will be considered on $cell_i$ if the following condition is satisfied

$$(d_{N_G} > ((i - 1) \times r)) \ \& \ (d_{N_G} \leq (i \times r)) \quad (25)$$

where d_{N_G} refers to the distance between the node and the gateway and r is the cell width. Once a node determines its cell, it will be assigned the unique channel and the transmission power level that is defined for that cell (line 11-13).

Regarding the distribution of SFs among the cells, each cell is assigned a set of SFs and in order to distribute the set of eligible SFs among the nodes on that cell, we further divide each cell into a number of sub-cells ns that is equal to the number of eligible SFs on each cell, as shown in fig. 16. For example, $cell_1$ supports all six SFs, so it will be divided into six equal-width sub-cells and each sub-cell is assigned one SF. On the other hand, $cell_2$ supports only five SFs (from SF8 to SF12), so it will be divided into five sub-cells by assigning one SF for each sub-cell. In other words, closer cells to the gateway will have smaller cell identifier, and hence it will have larger number of eligible SFs as explained in Algorithm 2 line (14). We named this approach as the Cell-based distribution. In order for a node to determine its sub-cell and hence its SF, it will first iteratively determine the boundaries of sub-cell of the cell that is located on (line 16-18). Once done, based on its distance to the gateway it will determine its sub-cell and hence its spreading factor (line 19-21).

According to LoRa specification, spreading factors are orthogonal. Indeed, if two nodes n_1 and n_2 as shown in Fig. 16 on the same cell, $cell_2$, but

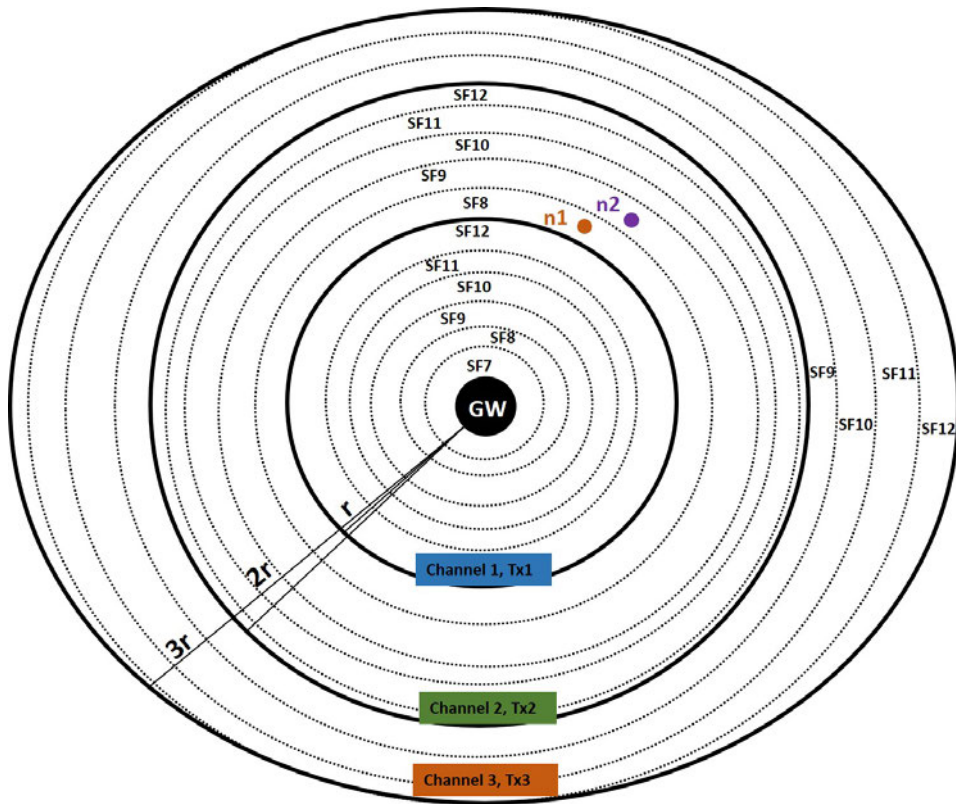


Figure 16: Partitioning network area into cells and sub-cells.

on different sub-cells, which means on the same channel but using different SFs, transmit simultaneously, there will be no collisions at the receiver due to the orthogonality of the SFs. By splitting the area around the gateway into multiple cells and sub-cells with different transmission parameters, we mitigate collisions resulting from the use of the same transmission parameters and hence the scalability will be improved.

Fig. 17 shows the difference between the SF distribution of our proposed protocol and the Adaptive Data Rate (ADR) algorithm of LoRaWAN. As shown in the figure, 4000 nodes are distributed in a radio range of 14 km around a single gateway. In the ADR algorithm, the majority of nodes have the same SF, which is SF12 in this example. This is mainly because of the large number of nodes that is supported by the gateway. According to ADR algorithm, the gateway transmits periodic downlink messages to nodes to update their SF and transmission power but due to the duty cycle restrictions of the ISM band, the gateway cannot transmit these packets to all the nodes in the network. Consequently, nodes that are not receiving these downlink packets from the gateway in a specific period of time will suppose

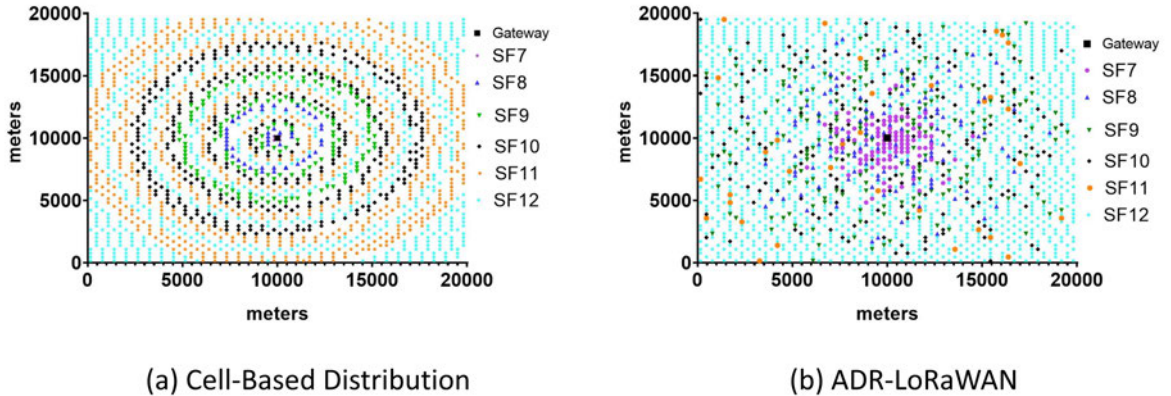


Figure 17: Allocation of SF for 4000 nodes: (a) Cell-based and (b) ADR-LoRaWAN.

that their transmissions were not successfully received by the gateway and hence they end up increasing their SF. Consequently, majority of nodes will end up using SF12 as shown in Fig17.b. This will increase the probability of collisions as well as decreasing the network throughput as explained in section 5.5. However, in our proposed algorithm, we maintain the diversity of SFs among the nodes even with challenging large-scale dense deployment area thanks to the autonomy of the proposed algorithm, where nodes specify their transmission parameters without any required intervention from the server.

5.4 Sector-Based Time-Slotted LoRa (SBTS-LoRa) Protocol Description

In order to avoid collisions between nodes on the same sub-cell that are sharing the same channel, the same transmission power and the same spreading factor, we regulate their transmissions by assigning a unique timeslot for each node. To do so, we propose SBTS-LoRa protocol that is described in details in the following sections.

5.4.1 Overview

Many researches have emphasized on the importance of changing the ALOHA access method of LoRa networks and using instead Time-Division Multiple access (TDMA) [13][132]. For instance, authors in [57] found that the maximum LoRaWAN throughput with slotted Aloha S-ALOHA is the double of

LoRaWAN throughput with pure-Aloha P-ALOHA. These findings confirm the efficiency of adopting TDMA approaches on top of LoRa physical layer to mitigate collisions and hence increase its scalability. According to that, there are some researches that have proposed TDMA medium access algorithms for LoRa networks [117][15][18] [69] [16]. However, all these algorithms use centralized approaches to schedule transmissions between nodes. In other words, both the newly-joined and the already-joined nodes need to receive periodic transmission schedules from the gateway using extensive downlink transmissions. As a result, the scalability will be compromised because of the duty cycle restrictions of the downlink transmissions that are imposed by the ISM band. However, in our proposed algorithm, the schedules are determined by the nodes autonomously without any need of downlink transmissions from the gateway. Specifically, based on node's location to the gateway, each node independently determines its slot number and hence its transmission time. Since LoRa networks have a star network topology with a gateway centered in the middle and nodes are distributed around the gateway, we get inspired from the geometry of circles and sectors of circles to specify independently the transmission schedule of each node. The following sections describe in details the initialization and data transmission phases of the proposed SBTS-LoRa protocol.

5.4.2 The Initialization Phase

In order to avoid collisions between nodes that are on the same sub-cell, which they use the same SF on the same channel, we further divide cells into sectors to which we assign numbers, as shown in Fig. 18. The number assigned to each sector is simply the slot number for every node belonging to that sector. By doing so, the same slot number can be reused by nodes that are in different subcells without any collision since they are using different orthogonal SFs. As a result, the frame size can be reduced such that only a limited number of timeslots can safely serve large number of nodes. As for nodes that are in the same sub-cell, they will get different slot numbers as they reside in different sectors.

We assume that a node n knowing its geographical coordinates (X_n, Y_n) and the ones of the gateway (X_G, Y_G) will automatically and autonomously know to which sector it belongs and hence its slot number for transmission. To do that, each node located on $cell_i$ calculates an angle α_i according to which $cell_i$ will be partitioned into sectors of angle size α_i with the gateway as sector origin. Fig.18 shows an example of sectors division of $cell_3$. As shown in Fig.18, $cell_3$ has 4 sub-cells and 8 different sectors each with an angle α_3 . The sectors will be assigned a number according to their relative position

from the gateway as it will be detailed later. The intersection between a sub-cell and a sector is called *Annulus-Sector*. Given that, each sub-cell is assigned a different SF, all nodes located on a given sector use that sector number as their slot number. In other words, the same slot number can be used by nodes on different sub-cells without any collision since they use different SFs. Similarly, in order to avoid collisions between nodes on the same sub-cell, we aim at guaranteeing the existence of a unique node per *Annulus-Sector*. In other words, to guarantee that only one node exists in every *Annulus-Sector*, the following equation should be satisfied:

$$\left[\frac{\alpha_i}{2} \times \left(ir^2 - \left(ir - \frac{r}{6 - (i - 1)} \right)^2 \right) \right] \times d = 1 \quad (26)$$

where i is the cell identifier that ranges from $i = 1$ (the closet cell) to $i = 6$ (the farthest cell), d is the node density, α_i is the sector angle of $cell_i$, and r is the annulus width, which is the maximum distance from the gateway, R divided by 6.

The first term on the left hand-side of equation (26) simply denotes the area of the *Annulus-Sector* of the last sub-cell of $cell_i$, which is multiplied by the node density d to estimate the number of sensor nodes in the last *Annulus-Sector*. According to equation (26), the number of nodes in the last *Annulus-Sector* must be equal to 1 in order to have just a unique node uses a given slot number and a given SF. Any other node on the same cell has either a different SF or a different slot number. Indeed, if we guarantee that the largest *Annulus-Sector* in a given sector contains only one node, we can deduce that for all previous *Annulus-Sector*, there is at maximum one node since previous *Annulus-Sector* have smaller areas than the last one. By doing that, we can achieve collision-free transmissions without the need for extensive down-link transmissions. To demonstrate that, let us consider the example that is shown in Fig.18. The last and largest *Annulus-Sector* here is *Annulus-Sector 4*, if we guarantee that there is only one node in that sector, we can guarantee different slot numbers for nodes that use same SF on the same channel. Furthermore, by guaranteeing uniqueness timeslots for nodes located on the largest *Annulus-Sector*, we can deduce that nodes that are located on smaller *Annulus-Sector*, which are in our example the *Annulus-Sectors* ranging from 1 to 3, have unique slot numbers.

From the previous equation, we can derive the angle of $cell_i$ as follows:

$$\alpha_i = \frac{2}{d \times \left(ir^2 - \left(ir - \frac{r}{6 - (i - 1)} \right)^2 \right)} \quad (27)$$

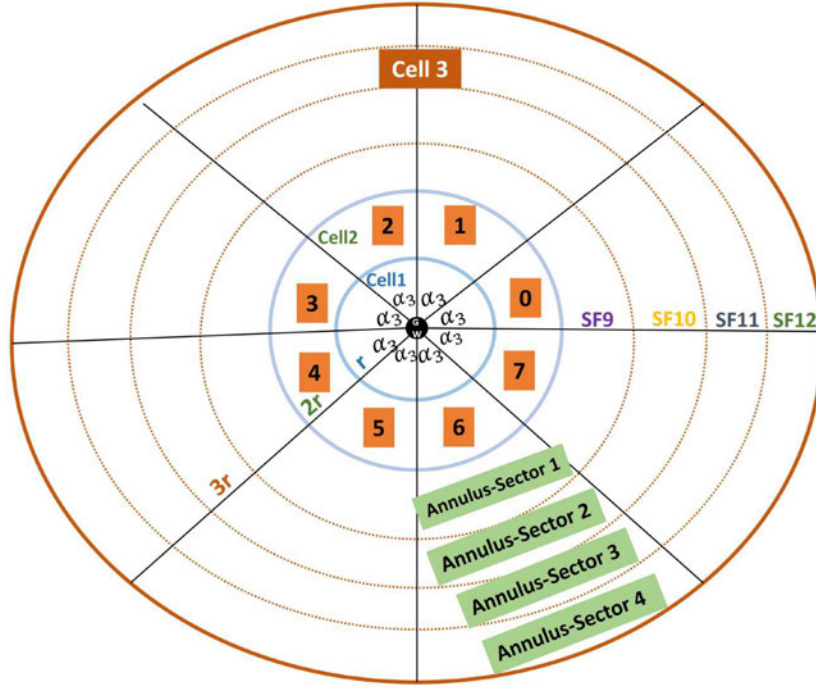


Figure 18: Dividing cells into sectors.

In fact, the size of a sector angle α is affected by two main factors: i) the node density of the network, d and ii) the radius of $cell_i$ that equals ir . Furthermore, the size of α has a direct impact on the frame size. The frame size in SBTS-LoRa is the total number of available time slots per $cell_i$. According to SBTS-LoRa, each cell has its own frame size that depends on the radius of that cell. In other words, closer cells have frame size smaller than the farther cells. This means that farther cells with less number of eligible SFs have more time slots. This will have the advantage of mitigating collisions on farther cells due to the reduced number of eligible SFs on these cells. For example, since nodes on $cell_6$ are so far from the gateway, they can use only SF12. Hence, if there are two or more nodes on $cell_6$ transmitting simultaneously, there will be collisions between them. By having large frame size for far cells, we allow for more collision-free communication for those nodes.

After a node calculates the angle α_i of its $cell_i$, it can easily calculate the frame size m_i of its cell by the following formula:

$$m_i = \frac{2\pi}{\alpha_i} \quad (28)$$

Note that, according to SBTS-LoRa protocol, there is a fixed frame size for

each cell. However, the slot duration is varying depending on the given sub-cell that reflects the used SF. This is because the frame size depends on the angle of α_i , where the α_i depends on the cell radius. The cell radius is doubled when cells get farther from the gateway. For example, as shown in Fig. 18, there are 8 timeslots for $cell_3$ regardless of the used SF. However, the slot duration for nodes located on the first sub-cell is different from the slot duration of nodes located on the last sub-cell since they use different SFs in their transmissions. In fact, the slot duration here acts as a third dimension besides the coordinates of the nodes. Fig. 19 shows a demonstrative example of the frame structure of $cell_3$. As shown in the figure, the frame consists of 5 timeslots regardless of the used SF. However, the duration of the slots for each frame is varying depends on the used SF.

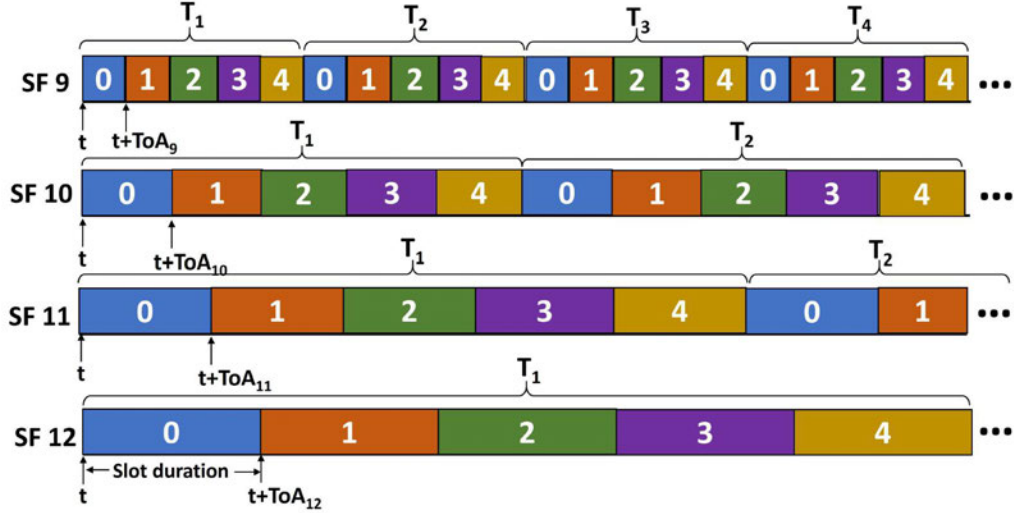


Figure 19: SBTS-LoRa frame structure of a cell partitioned into four sub-cells.

After that, a node n calculates its polar angle θ_n based on the following formula:

$$\theta_n = \tan^{-1} \left(\frac{Y_n - Y_G}{X_n - X_G} \right) \quad (29)$$

where (X_n, Y_n) are the coordinates of node n and (X_G, Y_G) are the coordinates of the gateway. Fig. 20 shows the polar coordinates of node n . Finally, each node n finds its slot number s_n by applying the following equation:

$$S_n = \frac{\theta_n}{\alpha_i} \quad (30)$$

Fig. 21 shows a Flowchart diagram of the proposed SBTS-LoRa protocol.

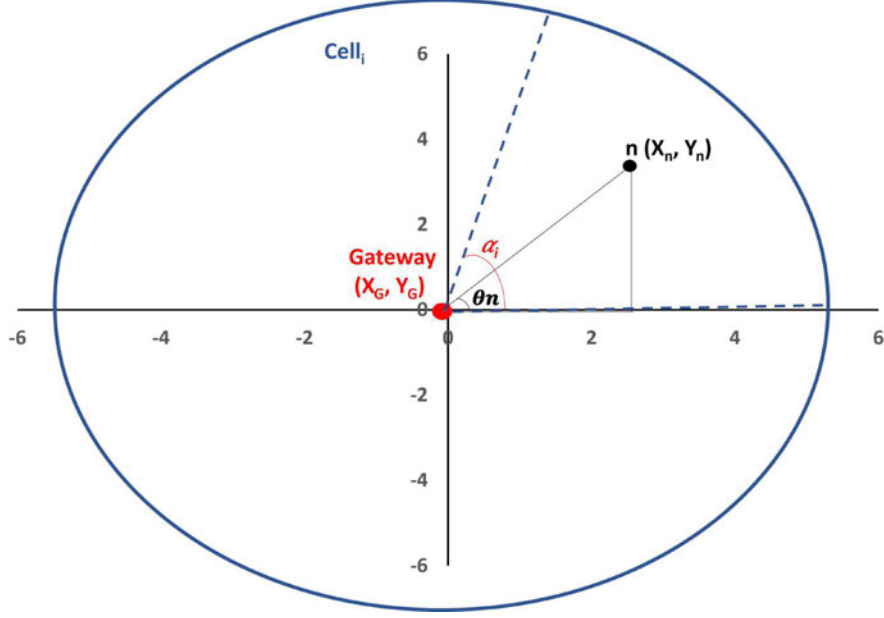


Figure 20: Polar coordinates of node n .

Furthermore, Algorithm 2 from line (26-30) illustrates how a node determine independently its timeslot ID. Once the slot number is determined, every node sleeps until its timeslot begins to start its data transmission.

5.4.3 Data Transmission Phase

During the data transmission phase, each node wakes up at the beginning of its timeslot to transmit its packet. According to SBTS-LoRa protocol, each node transmits one packet per frame. As mentioned earlier, there is a given frame size for each cell, which depends on the angle size of the sectors of a given cell. To make sure that nodes are transmitting once per frame without violating their duty cycle, the following condition must be satisfied:

$$t \times m_i \times ToA_{SF} \geq \psi_n \quad (31)$$

where t refers to an integer that denotes the time frame ID, ToA_{SF} is the Time on Air of a packet that uses a given SF, m_i is the number of timeslots per $cell_i$, and ψ_n is the duty cycle of node n , which is the time a node must wait before it can transmit again.

In other words, once a node succeeds its transmission of a packet, it will schedule the time for the next packet such that it will not violate its duty

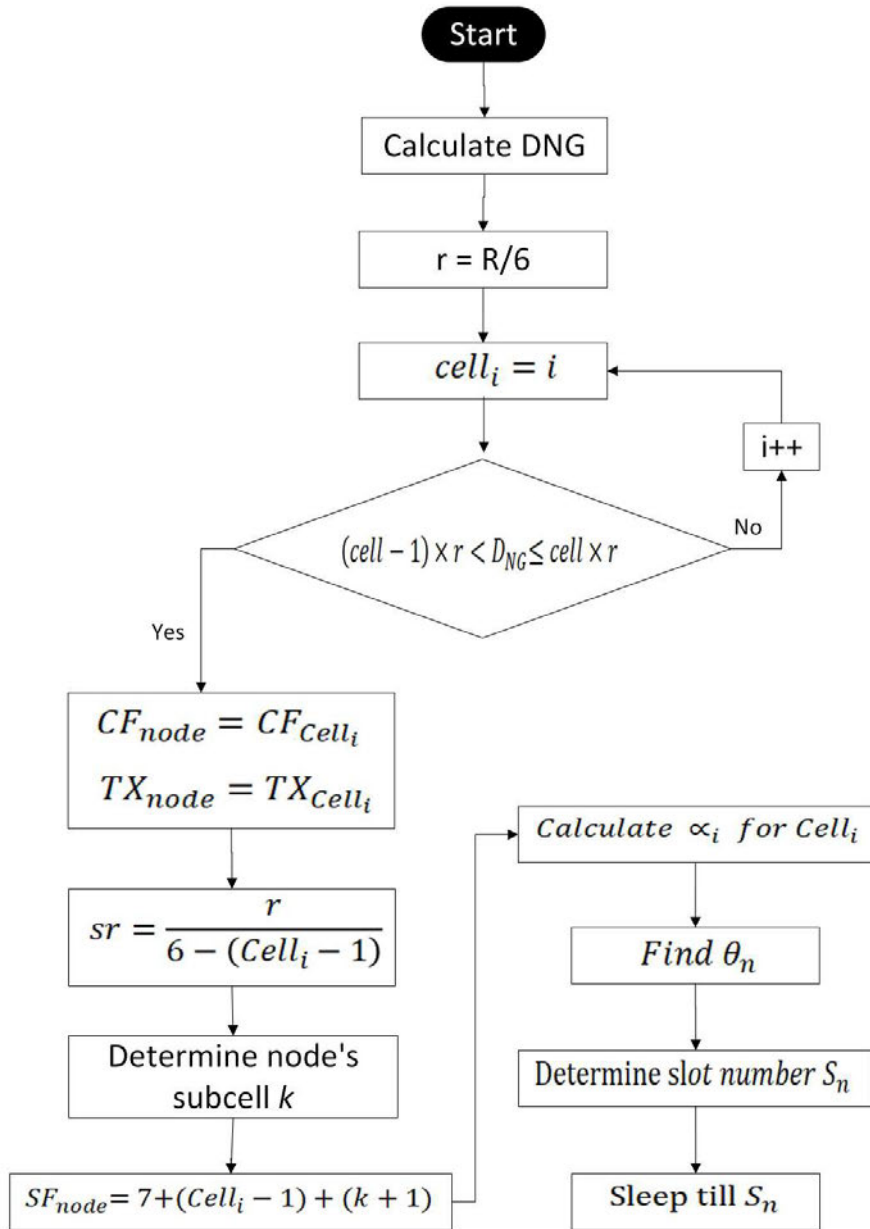


Figure 21: Flowchart diagram of SBTS-LoRa protocol

cycle ψ_n . To do that, a node will calculate the left-hand side of equation 31 by multiplying the frame ID t (start from 1) by the frame length m_i and ToA_{SF} of the selected SF. If the resulting waiting time for the node was less than the duty cycle ψ_n , which means that the node cannot send a packet in the next upcoming frame, it will postpone the transmission to the next frame by increasing the frame ID t by one. This process is repeated until the node finds the frame ID t for the next packet transmission that complies with its duty cycle ψ_n . By doing that, nodes respect their duty cycles as well as their schedule of transmissions.

5.4.4 Finding the Optimal Frame Size

According to SBTS-LoRa protocol, the frame size of cells that are closer to the gateway is less than the frame size of cells that are farther from the gateway. This is mainly because the angle of closer cells is larger than the angle of farther cells. This has an advantage of having larger frame sizes for nodes on farther cells where there is a limited number of SFs that could be used. However, extremely large frame size is undesirable as it will increase the waiting time for a node to transmit again especially for farther nodes, where large SFs with high ToA are used and thus the network throughput may be badly affected. Hence, a trade off between the frame size m_i and the number of nodes per *Annulus-Sector* of a given cell, p , must be performed.

As mentioned in the previous section, the frame size m_i (Eq. 28) is mainly affected by α_i which in turn depends on the number of nodes per *Annulus-Sector* of a cell, called hereafter as p . Accordingly, Equation 27 can be generalized as follows:

$$\alpha_i = \frac{2 \times p}{d \times \left(ir^2 - \left(ir - \frac{r}{6 - (i - 1)} \right)^2 \right)} \quad (32)$$

Note that in Equation 27, p is set to be equal 1. Indeed, setting p equals 1 means that we impose the presence of a unique sensor in the last *Annulus-Sector* of a given cell. Imposing a unique sensor in the last *Annulus-Sector* will result in at most one sensor in the other *Annulus-Sectors* of the same cell as the last *Annulus-Sector* has the largest area. Obviously, if we increase the p value, more nodes will be sharing the same slot which may result in more collisions but at the same time increasing p will increase the α_i of a given $cell_i$ and hence the frame size will be reduced and thus the end-to-end delay will be reduced. Clearly, an optimal p value will maximize the throughput since there is a tradeoff between the collisions and the end-to-end delay.

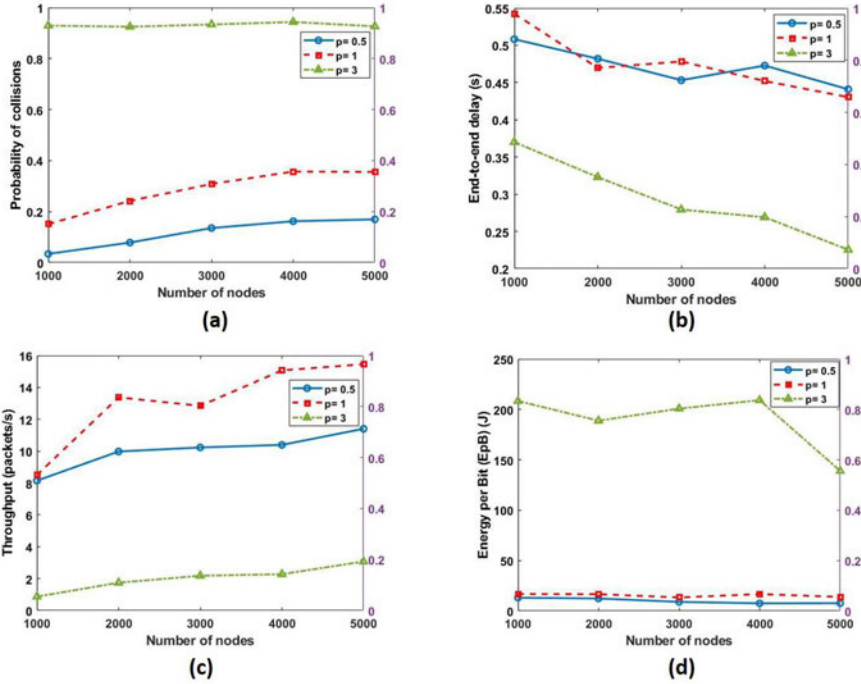


Figure 22: Finding optimal p value

Hence, we have run SBTS-LoRa with different node densities d and p values to find an optimal p that maximize the network throughput. Fig. 22 illustrates the result of the simulation analysis, where p equals 0.5, 1, or 3 and with a number of nodes ranging from 1000 to 5000 nodes. We evaluate the impact of p in terms of the probability of collision (Fig 22.a), the end-to-end delay (Fig 22.b), the throughput (Fig 22.c), and the total energy consumed per a successful reception of a bit (Fig 22.d). As shown in Fig 22.c, the maximum network throughput was achieved when $p=1$. Furthermore, although the probability of collision and the end-to-end delay is not the best with $p=1$, this is not compromising the energy consumption as it is also optimized with $p=1$. Hence, the optimal p for our simulations is 1. Thus, in section 5.5, we evaluate SBTS-LoRa protocol with $p=1$.

5.5 Performance Evaluation

This section presents a simulation-based assessment that is performed on SBTS-LoRa protocol using OMNET++ [119] simulator under FLoRa framework [120]. As mentioned earlier, SBTS-LoRa is intended for large scale environments, where the number of connected nodes is extremely large. Ac-

Table 18: Simulation parameters for evaluating SBTS-LoRa protocol

Parameter	Value	Comments
SF	7 to 12	Spreading Factors.
TP	(2, 5,8,11,14) dBm	Transmission power.
CR	4/8	Coding Rate.
CF	{868.1, 868.3, 868.5, 867.1, 867.3, 867.5}	Carrier Frequencies (MHz).
BW	125 kHz	Bandwidth.
PL	20 Bytes	Payload length.
R	14 Km	Field radius.
N	1000-5000	Number of nodes.
Simulation Time	10	Days.

According to that, the number of connected nodes in the performance evaluation ranges from 1000 to 5000 nodes that are randomly distributed within a radius of 14 Km from the gateway. Regarding the packet size, each node generates a packet of 20 bytes length with an inter-arrival time between packets following the exponential distribution with 1000s mean. Similar to [120], we used the European regional parameters for the LoRa physical layer with 1% duty cycle for both the LoRa nodes and the gateway. Table 18 Summarizes the simulation parameters of the simulations. We compare the performance of the SBTS-LoRa protocol to EXPLoRa-AT [123], BitRateRatio (BRR) [109], and the Adaptive Data Rate (ADR) algorithm of LoRaWAN [8]. To do that, both EXPLoRa-AT [123] and BRR [109] algorithms have been implemented using OMNETT++ and FloRa framework. In order to make this section self-contained, the following sections provide an overview about EXPLoRa-AT and BRR algorithms that were used as benchmark along with the ADR-LoRaWAN algorithm to evaluate the proposed SBTS-LoRa protocol.

5.5.1 EXPLoRa-AT Algorithm

EXPLoRa-AT uses LoRa default channel and a fixed transmission power (14 dBm). It distributes spreading factors among nodes by dividing nodes into six groups, which corresponds to the number of supported SFs in LoRa physical layer, such that all groups eventually will have similar Time-on-Air (ToA). Similar to ADR algorithm of LoRaWAN, EXPLoRa-AT runs at the server and distributes the recommended SF for each node in a centralized way. In other words, the server has to update individually each node with

the recommended SF. Specifically, once the server receives a packet from a node, it will send the updated SF to that node in its receiving windows. Therefore, duty cycle limitation imposed on gateway's downlink communication could be the bottleneck of the system. According to EXPLoRa-AT algorithm, all nodes start with SF12 to make sure that their joining requests are not received below the sensitivity at the server. Then, after the joining requests transmission phase, the server will run the algorithm, determine the recommended SF for each node according to the join packet's RSSI, and then transmit that SF on nodes' upcoming receiving windows. Since all nodes are sending only on the default channel using SF12, the collisions between the joining requests may prohibit some nodes from accessing the gateway. In fact, according to our simulations, the number of nodes that could reach the server is decreasing with the increase of node density.

In fact, with extremely large number of connected nodes, the server cannot receive packets from all nodes with the same SF and on the same channel due to collisions. According to that, only a small portion of nodes will succeed delivering their messages to the gateway, and hence their SFs will be appropriately adapted. Fig. 23 shows the number of nodes successfully initialized at the server, i.e. the number of nodes the server receives packets from and hence records their RSSIs in order to adapt their SFs compared to the actual number of nodes in the network for 12 days simulation time.

Furthermore, in order for the server to run EXPLoRa-AT algorithm, it has to wait for the network to stabilize where all nodes have sent their join requests so the server can have node's RSSIs in order to be able to estimate the appropriate SF according to the RSSI of node's join request packets. This condition could be hard to satisfy especially with large number of connected nodes (2000 nodes or more), or with mobile nodes. Algorithm 3 shows the main steps to implement the EXPLoRa-AT algorithm. The algorithm takes as input a vector with an initial number of nodes per SF based on the RSSIs of the joining request packets. The w vector is the time-on air weights for each SF. The algorithm creates firstly a channel congestion index vector P by multiplying the number of nodes per SF n_{vec} by the inverse of the time-on air weights vector w (line 8-9). Then, it collects the indexes of the unbalanced values in P vector (line 11-17). After that, the algorithm iteratively updates the P vector such that all the values of P will have a non-decreasing order (line 19 - 32).

5.5.2 The Bit Rate Ratio (BRR) Algorithm

the Bit Rate Ratio (BRR) algorithm, exploits the fact that the transmissions with smaller SFs have higher bit rates, and hence lower Time on Air (ToA),

Algorithm 3 EXPLoRa-AT Algorithm

```
1: Input: initial number of nodes per SF  $n_{vec}$ 
2: Output: the updated number of nodes per SF after balancing the ToA
    $n_{vec}$ 
3:  $SF \leftarrow [7, 8, 9, 10, 11, 12]$ 
4:  $w \leftarrow [1, 1.83, 3.33, 6.67, 13.34, 24.04]$ 
5:  $q \leftarrow w^{-1}$ 
6:  $x \leftarrow 0$ 
7:  $p_{vec} \leftarrow -1$ 
8: for  $i \leftarrow 1$  to 6 do
9:    $P[i] \leftarrow n_{vec} \times w[i]$ 
10: end for
11: for  $i \leftarrow 1$  to 6 do
12:   if  $P[i] > P[i - 1]$  then
13:      $pidx.add(i - 1)$ 
14:      $idx.add(x)$ 
15:      $x ++$ 
16:   end if
17: end for
18:  $old_p \leftarrow idx$ 
19: while  $old_p \neq p_{vec}$  do
20:    $start \leftarrow 0$ 
21:   for  $pidx.begin()$  to  $pidx.end()$  do
22:      $i \leftarrow pidx.begin()$ 
23:     for  $j \leftarrow start$  to  $i$  do
24:        $P_{num+} = P[j] \times q[j]$ 
25:        $P_{dem+} = q[j]$ 
26:     end for
27:     for  $j \leftarrow start$  to  $i$  do
28:        $P[j] \leftarrow P_{num+} / P_{dem+}$ 
29:     end for
30:      $P_{num} \leftarrow 0$ 
31:      $P_{dem} \leftarrow 0$ 
32:      $start \leftarrow i$ 
33:   end for
34: end while
```

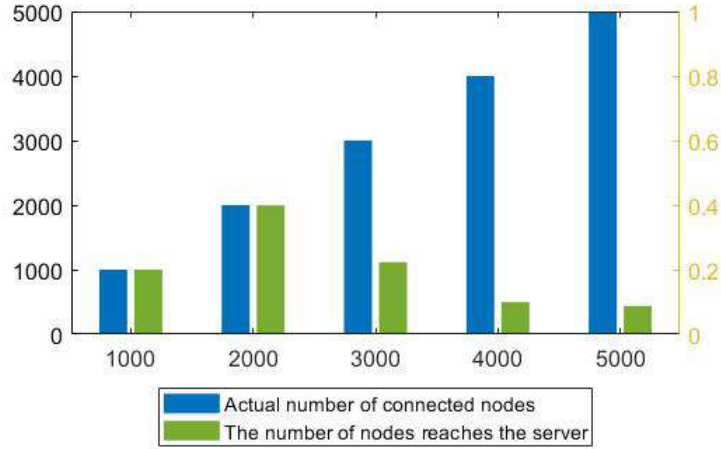


Figure 23: Scalability limitation of EXPLoRa-AT algorithm.

than the transmissions with larger SFs. Therefore, the BRR algorithm distributes the spreading factors among nodes such that large portions of nodes use small SFs and only small portions of nodes use large SFs. By doing that, collisions resulting from long transmissions could be mitigated. To do that, the portion of nodes P_{s_i} that use a given SF_i is calculated according to the following equation [109]:

$$P_s = \frac{\frac{SF}{2^{12}}}{\sum_{i=7}^i \frac{1}{2^i}} \quad (33)$$

Similar to EXPLoRa-AT [123] and ADR [8] algorithms, the Bit Rate Ratio [109] algorithm is implemented in a centralized approach, where it runs at the server that is responsible of distributing SFs using downlink communications. However, we have implemented the algorithm in a distributed manner as shown in Algorithm 4, so our proposed protocol can be evaluated with respect to both centralized and distributed algorithms. According to our implementation of BRR [109] algorithm, each node firstly determines the set of eligible SFs that can select from according to its path loss to the gateway. To do so, we use the cell-based distribution of SFs as explained in Section 5.3. Then, each node autonomously calculates the P_{s_i} (equation 33) for each SF. After that, each node will generate a vector of aggregates P_{s_i} , called AP_{s_i} , for each SF (line 8-10). Fig. 24 shows an example of aggregated ratios where the set of eligible SFs accommodate all SFs (SF7-SF12). After that, each

node will generate a random number x between $(0,1]$ and check at which SF interval it exists. Then, the node will select the corresponding SF for that period (line 11-14). For example, as shown in Fig. 24, when the generated random number $x=0.8$, the node will select SF9 for its transmissions. As illustrated in Fig. 24, smaller SFs have larger intervals of random numbers than larger SFs. Note that, in order to determine the set of eligible SFs for each node, we have been inspired by the Cell-based distribution by dividing the area around the gateway into cells such that each cell has different set of eligible SFs that depends on the distance to the gateway. However, note that BRR [109] uses only the default channels unlike the proposed SBTS-LoRa protocol.



Figure 24: Example of the implemented Bit Rate Ratio Algorithm.

Algorithm 4 The Bit Rate Ratio Algorithm

- 1: **Input:** the started SF of the cell range $start$
 - 2: **Output:** The assigned SF for a node $node_{SF}$
 - 3: $sum \leftarrow 0$
 - 4: **for** $i \leftarrow start$ to 12 **do**
 - 5: $P_s \leftarrow (i/2^i)/sum(i/2^i)$ ▷ Eq.33
 - 6: **end for**
 - 7: $APs_i \leftarrow P_s[0]$
 - 8: **for** $i \leftarrow 1$ to 6 **do**
 - 9: $APs_i \leftarrow APs_i[i - 1] + APs_i[i]$
 - 10: **end for**
 - 11: $x \leftarrow rand(0, 1]$
 - 12: **for** $i \leftarrow 1$ to 6 **do**
 - 13: **if** $x \geq APs_i[i - 1] \ \& \ x < APs_i[i]$ **then**
 - 14: $node_{SF} \leftarrow start + i$
 - 15: **end if**
 - 16: **end for**
-

The following sections demonstrate the performance evaluation of the proposed protocol with respect to the end-to-end delay, network throughput,

probability of collisions, Packet Error Rate (PER), and energy consumption metrics.

5.5.3 The Probability of Collision

Fig. 25 shows the probability of collision as a function of the number of nodes. First, the probability of collision of all protocols is increasing with the increase of the number of nodes as the traffic rate is getting higher. However, SBTS-LoRa is increasing slowly compared to other protocols. Most importantly, SBTS-LoRa protocol is achieving the lowest probability of collision compared to other protocols thanks to the efficient distribution of transmission parameters and timeslots among the nodes that aim at mitigating collisions to the most possible extent. Our implementation of the BRR algorithm has the second lowest probability of collisions since it is distributed, where each node autonomously determines its transmission parameters.

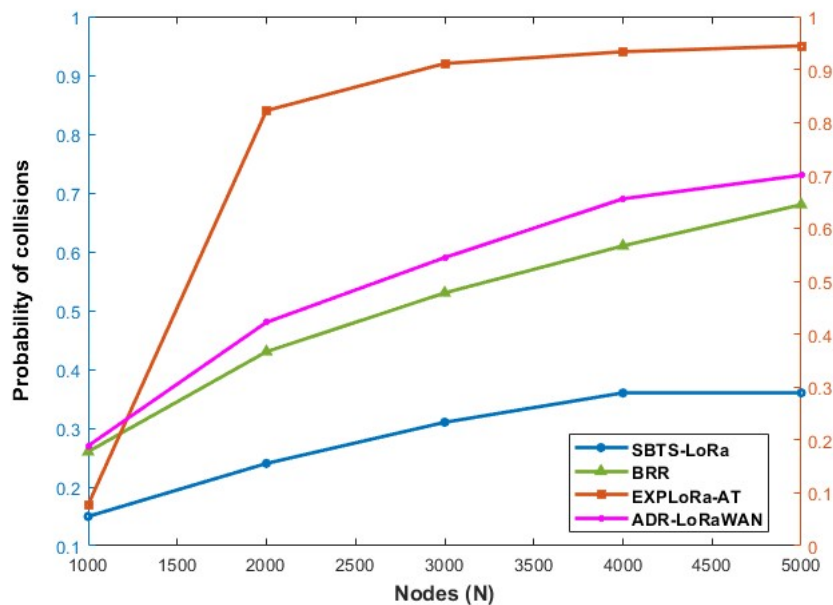


Figure 25: Probability of collision.

On the other hand, ADR-LoRaWAN and EXPLoRa-AT have the highest probability of collisions because they are centralized protocols, where the selection and the distribution of the transmission parameters are performed at the server. Furthermore, LoRaWAN and EXPLoRa-AT use only the default channels without exploiting the multi-channel feature of LoRa physical layer and thus all the nodes tend to communicate on the same channel which will

further increase the number of collisions. Specifically, since these protocols are centralized protocols, they have both uplink and downlink communications that are performed only on the default channels, which will further increase the probability of collisions. EXPLoRa-AT has the highest probability of collisions, especially with extremely high number of nodes (more than 2000 nodes) because all nodes, according to EXPLoRa-AT algorithm, start their transmissions with SF12, to insure the successful reception of packets regardless of node's distance to the gateway. However, with an increasing number of nodes that send packets with the same highest SF12, the probability of collision is getting worse.

5.5.4 The End-to-End Delay

Fig. 26 shows the end-to-end delay as a function of the number of nodes. The end-to-end delay is greatly affected by the used SF. Indeed, the usage of smaller SFs results in lower end-to-end delay as the packets that are transmitted by small SFs will have less Time on Air (ToA). According to SBTS-LoRa, SFs are assigned based on node's locations. This explains the minor decrease of SBTS-LoRa curve at $N=2000$ nodes compared to others. Furthermore, the delay is only considered for the successfully received packets by the gateway. This explains why the end-to-end delay of BRR, EXPLoRa-AT, and ADR-LoRaWAN are generally decreasing with the increased traffic, as collisions for these protocols are increasing with the increased traffic (Fig.25), and hence the number of successfully received packets, for which the end-to-end delay is counted for decreases. Moreover, EXPLoRa-AT has the lowest end-to-end delay compared to other protocols, especially when $N=2000$ or more, as the number of nodes that successfully reaches the server is getting too small as shown in Fig. 23. With high number of connected nodes, only the traffic of a small portion of nodes reaches the server, which explains the small end-to-end delays achieved by EXPLoRa-AT.

The BRR algorithm, on the other hand has the highest end-to-end delay because the majority of nodes (about 75%) have large SFs (SF10-SF12). Obviously, this distribution is not compliant with the main objective of BRR, where the majority of nodes is supposed to use small SFs. In our simulations, nodes are distributed in large-scale environment and the area around the gateway is divided into annulus-cells such that the cells that are located farther from the gateway have higher area and hence accommodate higher number of nodes. Consequently, nodes located farther from the gateway are eligible to select only large SFs, which will result in higher delays.

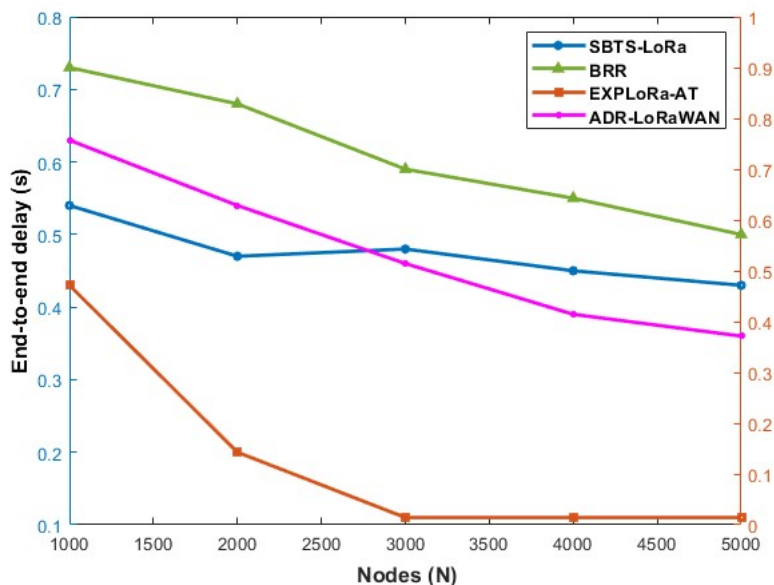


Figure 26: The end-to-end delay.

5.5.5 The Throughput

The throughput is mainly affected by the end-to-end delay and the probability of collisions. Fig. 27 demonstrates the network throughput as a function of the number of nodes. First, it is worth pointing out that SBTS-LoRa is by far achieving the highest throughput compared to other protocols with almost 15 packets/second that can be successfully received by the gateway when $N=5000$ nodes. Indeed, SBTS-LoRa is achieving in average 1362 % larger throughput than the second best protocol BRR. Indeed, BRR achieves only in average 1.18 packets/second that could be successfully received by the gateway on the same network density. In other words, the network throughput with SBTS-LoRa protocol is at least 8 times greater than the achieved throughput with other protocols. In fact, this emphasizes how changing the access method from ALOHA to TDMA is vital in terms of network throughput. In fact, having a timeslot for each node for transmission instead of the random access to the medium will inevitably decrease collisions, as demonstrated in Fig.25. Specifically, with large number of connected nodes that are scattered in a wide-coverage area, which is the case of LPWA networks, nodes will not just encounter destructive collisions, where all collided packets are destroyed, but also encounter the capture effect, where strong signals are dominant on weak ones. This in fact limits the scalability of the network, which is considered as a key challenge in LoRa networks as mentioned in

Section 5.1.1. Furthermore, TDMA methods are best suited for high density networks, where timeslots are efficiently used by nodes, and hence the end-to-end delay is minimized. Consequently, using TDMA method will increase the network throughput and hence, the network scalability will be enhanced.

Indeed, despite the fact that EXPLoRa-AT and LoRaWAN are achieving lower end-to-end delays than our protocol (Fig. 26), they end up with lower throughput as the impact of collision is much important (Fig. 25). Specifically, the throughput of EXPLoRa-AT is decreasing with the increase of the network density due to the increase of the probability of collisions (Fig.25). On the other hand, the throughput of LoRaWAN and BRR is almost stabilized even with the increase of the network density due to the stabilization of collisions as the network is saturated (see Fig. 25). However, it is worth pointing out that the throughput of SBTS-LoRa protocol depicts a clear increase with the increase of the number of nodes. Thus, our protocol is much more scalable compared to other protocols.

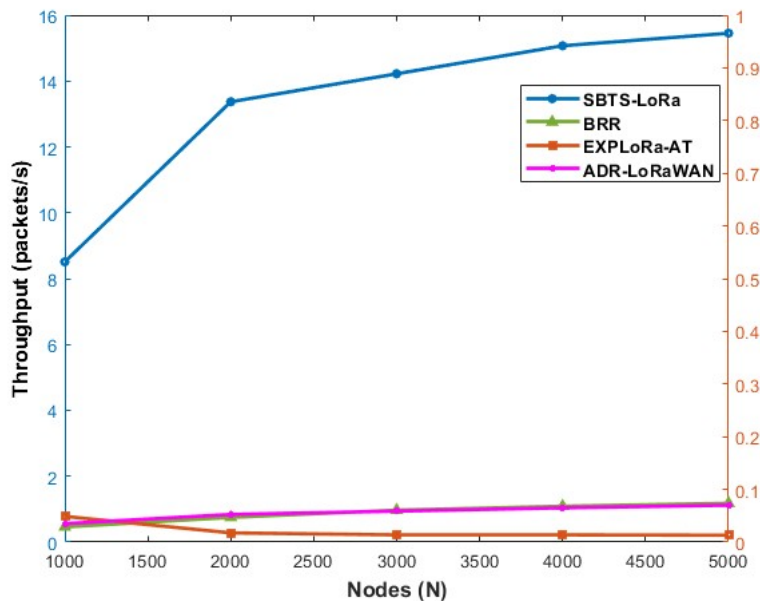


Figure 27: The network throughput.

5.5.6 The Packet Error Rate (PER)

The PER is the ratio of the total number of packets that are received under the gateway sensitivity. Since EXPLoRa-AT and LoRaWAN use centralized approaches in how they assign SFs to nodes, they have the lowest PER

compared to the distributed ones. In fact, the server in centralized network has better knowledge about the network. Consequently, the distribution of SFs and other transmission parameters is more accurate. This is true for ADR-LoRaWAN algorithm as shown in Fig. 28 . However, unlike the ADR-LoRaWAN, the PER of EXPLoRa-AT is almost zero as all nodes start their traffic using SF12 and then change their SFs if they received downlink communication from the server that contains the new selected SF. Note that the server in this algorithm only receives traffic from a very small portion of nodes (Fig. 23), that are close to the gateway. As a result, only those nodes use smaller SFs than SF12 while all the remaining ones are using SF12. Hence, if no collision happens, the delivery of EXPLORA-AT is guaranteed as nodes mostly use high SFs with the maximum transmission power. On the other hand, BRR is achieving the highest PER as nodes randomly choose the transmission power level and the SFs selection is based on a bounded random variable x , as explained previously. The PER of SBTS-LoRa increases with extreme high number of nodes (more than 4000 nodes), as more nodes could inappropriately select SFs and/or transmission power level. Although the PER of our protocol increases with the increase of the number of connected nodes, this is not affecting the overall network throughput (Fig. 27).

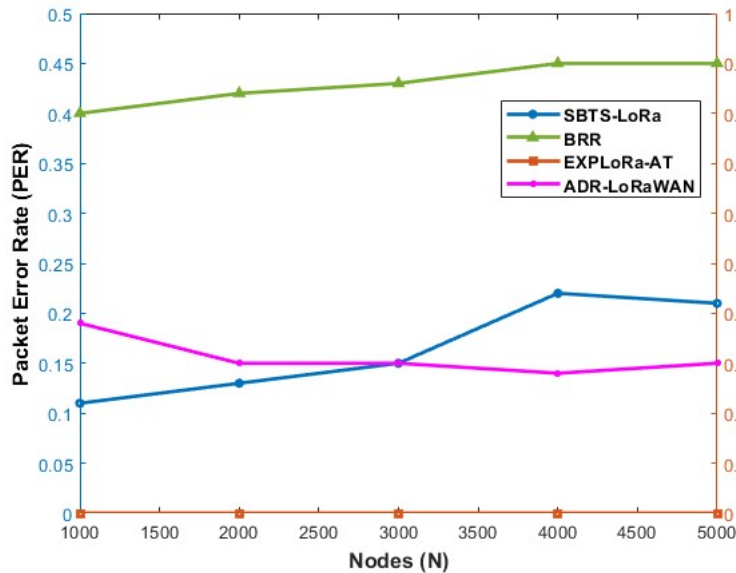


Figure 28: The packet Error Rate (PER).

Table 19: A summary of performance metrics.

	Collisions	End-to end delay (s)	Throughput (pack- ets/s)	PER	Energy per (J) bit
SBTS- LoRa	28%	0.47	13.05	16%	15.49
BRR	50%	0.61	0.89	43%	61.30
EXPLoRa- AT	76%	0.19	0.34	0%	257.08
ADR- LoRaWAN	55%	0.48	0.90	16%	54.85

5.5.7 The Energy Consumption (Energy per Bit)

Fig.29 shows the total energy consumed by nodes to successfully deliver one bit as a function of the number of nodes. Obviously EXPLoRa-AT has the highest energy consumption since it has the highest collisions (Fig.25). On the other hand, the energy consumption of both ADR-LoRaWAN and BRR algorithms is increasing with the increase of the network density, which is mainly due to their high probability of collisions (Fig.25), the high packet error rate (Fig.28), and the high end delays (Fig.26). However, SBTS-LoRa protocol achieves the lowest energy consumption since it has the lowest probability of collisions. Furthermore, the energy consumption level of nodes is stable even with the increase of the network density thanks to the efficient use of LoRa transmission parameters and the judicious allocation of time-slots that result in lower collisions. To sum up, table 19 shows the average of collisions, end delay, throughput, PER, and energy per bit for all evaluated protocols.

5.6 Conclusion

This chapter develops SBTS-LoRa, a novel MAC protocol that is targeting large-scale networks. SBTS-LoRa improves the scalability of the network by mitigating collisions resulting from nodes ALOHA-access method and avoiding downlink communication to control the transmission parameters distribution among nodes. SBTS-LoRa firstly divides the network into six annulus virtual cells such that all nodes on the same cell use the same channel and transmission power level. This will mitigate common issues such as the capture effect. Then, each cell is further divided into a number of

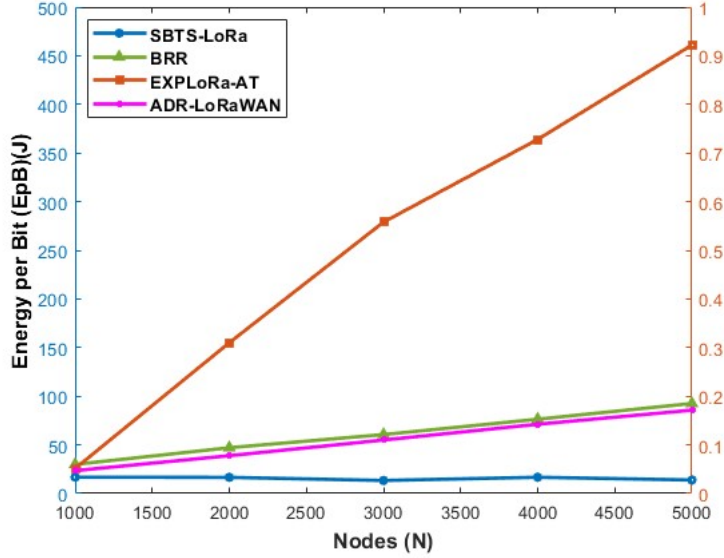


Figure 29: The energy consumption per bit (EpB)

annulus sub-cells that match the number of eligible SFs for a given cell. On top of that, each cell is further divided into sectors where the sector ID represents the timeslot ID for which a node transmits. Accordingly, nodes autonomously determine their transmission parameters and timeslot by only recognizing gateway and its coordinates. Simulation results show tremendous enhancements in collision rate, network throughput, and energy consumption. Specifically, for large-scale dense networks, the average throughput of SBTS-LoRa protocol is about 13 times better than the ADR-LoRawan.

Going forward, we will further enhance the network performance by considering a TDMA dynamic frame size instead of the static one. In fact, assuming static features for IoT networks may not be realistic for some scenarios. By adopting different node densities for cells with different areas, we can have dynamic frame length for each cell. In this case, we avoid having very long frame length for cells with smaller node density. This will further enhance the end-to-end delay and hence the network throughput is improved. According to that, the following chapter introduce a novel TDMA MAC prtocol with dynamic frame size.

6 Autonomous Adaptive Frame Size for Time-Slotted LoRa MAC Protocol

Striving for optimal configuration and distribution of transmission parameters, this chapter proposes an Autonomous Time-Slotted MAC protocol (ATS-LoRa) that allows LoRa nodes to individually determine their transmission parameters without extensive downlink transmissions from the gateway. In our novel approach, the nodes autonomously calculate their most efficient transmission parameters as a function of the location information of nodes and gateways. The autonomous selection of these parameters without the involvement of the gateway greatly enhances the network performance in terms of collision rate, throughput, end-to-end delay, and energy efficiency, as described in Section 6.2.

The first contribution of this chapter is the conception of a TDMA based MAC protocol that allows nodes to independently determine the appropriate transmission parameters without requiring any downlink traffic from the gateway. In addition, the proposed protocol instructs nodes to utilize the minimum SF whenever eligible. This is a very important feature in enhancing the transmission time, end to end delay, and energy consumption. In this context, eligibility means using the minimum SF to guarantee that the transmitted packet will not be received below the GW's sensitivity level. As long as all nodes use the appropriate SF to guarantee the delivery of their packets, the proposed protocol has a Packet Error Rate (PER) of almost zero. Moreover, in ATS-LoRa protocol, any channel has six different frame sizes corresponding to the available six SFs. The frame size has lower and upper boundaries taking into account nodes duty cycles. Furthermore, and as a main contribution, the frame size is dynamic, and it mainly depends on the node density on a given channel with a given SF. Simulation results show that ATS-LoRa protocol uses a frame size of only 150 slots with a very large number of connected nodes (4000 nodes). This is a very important feature that highly reduces the end-to-end delay and hence the network throughput is enhanced. Indeed, in the previous proposed protocol SBTS-LoRa in Chapter 5, although it considerably reduces the collision rate compared to LoRaWAN ADR algorithm, the large frame size in dense networks can be further investigated. In other words, with a larger number of connected nodes (5000 nodes) the frame size in SBTS-LoRa was quite large (up to 2400 slots). In this enhanced version, we aim at considerably reducing the frame sizes. In fact, one of the main limitations found in the literature is the fixed or large frame sizes of the proposed TDMA approaches.

The second contribution of this chapter is conducting a large-scale sim-

ulations using OMNET++ [119] simulator under FLoRa framework [120]. The goal of the simulation is to investigate the capacity of the network with a large number of connected nodes.

The chapter is organized as follows. Section 6.1 describes in details the proposed protocol with its three different modes. Section 6.2 presents the results of the evaluation process. Section 6.3 concludes the chapter with a summary of the main results and contributions.

6.1 Protocol Description

In this work, a star network topology is assumed with a gateway at the field's center and nodes distributed randomly in a field with a radius R . Communications between the nodes and the gateway are 1-hop. Although the 1-hop transmissions are considered simple, scheduling transmissions between nodes to achieve low collision rates is challenging especially with a large number of connected nodes, which is the case with long-range communications. Furthermore, in order to regulate the access to the unlicensed shared medium, LoRa is imposed to respect harsh duty cycles. The duty cycle can be defined as the amount of time a node or a gateway can transmit during a given time period. For example, in Europe band, which is the considered band in this study, the duty cycle for uplink transmissions from nodes to the gateway should not exceed 1%. Hence, each node has a maximum of 36 seconds per hour for transmissions. In other words, to maintain its duty cycle, a node should wait for at least $ToA_{SF} \times 99$ before transmitting again after transmitting a packet with a duration of ToA_{SF} , which is the packet's Time on Air using spreading factor SF . Downlink transmissions from the gateway to nodes have the same duty cycle fraction namely, the 1% duty cycle, except for one channel that has a 10% duty cycle. As a result, the gateway, with its limited duty cycle, cannot easily regulate node's uplink transmissions and their parameters through downlink packets. In other words, adopting a centralized approach where gateways control node's transmissions through downlink packets that are transmitted to individual nodes, which is the case of the Adaptive Data Rate ADR algorithm in LoRaWAN, might not be optimal. The gateway will reach its duty cycle limit before updating all nodes with their appropriate transmission parameters, as explained in the the previous chapter. Hence longer convergence time is needed in order to configure all the nodes. Note that this convergence time is not only long but also increases with the number of nodes in the network which will compromise the network scalability. Thus, duty cycle regulation is considered as one of the main challenges in LoRa networks [13]. To overcome such an issue, nodes need to set up their proper transmission parameters without extensive

downlink transmissions from the gateway. By allowing nodes to be aware of some aspects of their environment, they will be more intelligent and able to choose the most suitable transmission parameters. In particular, allowing each node to acquire only its own location and the gateway's location will assist in determining its transmission parameters, such as the appropriate SF, frequency, and time-slot. According to LoRa, the gateway sensitivity of each SF is different, with higher SFs, the level of sensitivity rises. As a result, the transmission ranges of higher SFs are greater than those of smaller ones. ToA, on the other hand, is longer for higher SFs than for smaller ones. As a result, shorter ToA leads to lower collision rates, and shorter end-to-end delays. Thus, using smaller SFs whenever possible is recommended as it may increase the network throughput. Considering that, in this work, we propose dividing the network field into six coronas, which equals the number of supported SFs in LoRa as demonstrated in Fig.30. As shown in the figure, each corona C_i has its own SF. The zone of each corona C_i is actually the eligible range of the SF that is assigned to that corona. The eligible range of a specific SF can be defined as the maximum distance from the gateway at which nodes can use a given SF safely so that their packet transmissions will not be received below the gateway's sensitivity level. In other words, the estimated maximum distance from the gateway at which a node can use a particular SF is represented by the radius R_i of each corona. Therefore, packet transmissions from nodes on a particular corona will not be received below the gateway's sensitivity level. As a result, determining the appropriate SF for each node necessitates accurately predicting SF ranges. The distribution of SFs among nodes is explained in the following section.

6.1.1 Determining the SF Transmission Parameter

This section explains the distribution of SFs among nodes. Because LoRa's SFs are orthogonal, concurrent signals on the same frequency encoded with different SFs can be safely received at the gateway as long as the difference in their received power does not exceed a certain threshold. Accordingly, we partition the network field into six coronas, which is matching the number of supported SFs in LoRa. A specific SF is allocated to each corona C_i . The smallest SF, or SF7, is given to the closest corona to the gateway. Coronas are assigned higher SFs as we move away from the gateway. The longest eligible distance for a node to use a particular SF such that its transmission is not received below the sensitivity threshold of the gateway is defined as the radius R of a given corona C_i . Hence, Packet Errors are mitigated. A node will compare its distance with the radius R_i of each corona C_i starting with the smallest corona to determine to which corona it belongs and thus

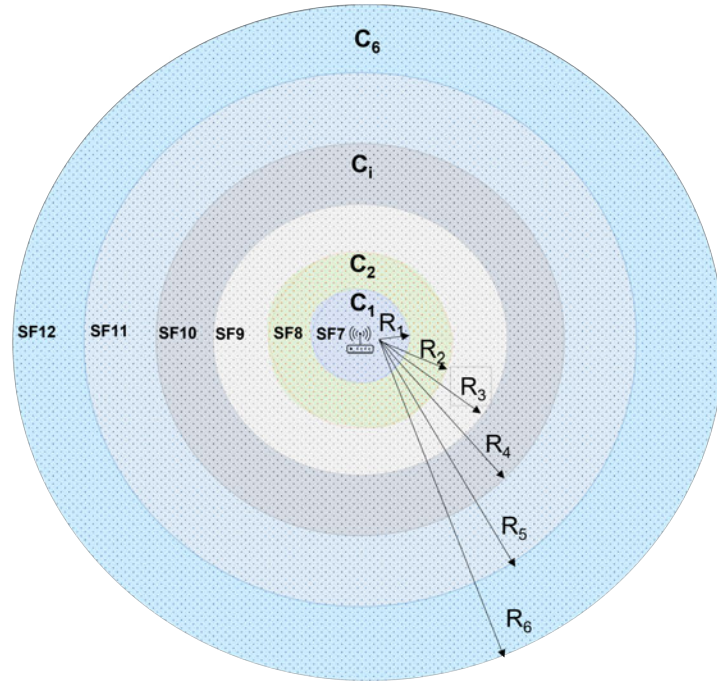


Figure 30: Dividing the field into overlapping coronas with variable SFs.

its appropriate SF for its transmissions. A node n considers itself belongs to corona C_i if the following condition is met: $R_{i-1} < dist_n \leq R_i$, where $dist_n$ is the distance between node n and the gateway. particularly, a node n will select the SF of corona C_i with a R_i that is greater than its distance $dist_n$. By doing so, receiving packets below the sensitivity level of the gateway is completely avoided. We assume that each node knows its coordinates as well as the coordinates of the associated gateway in order to calculate its distance to the gateway. In addition, each node keeps a vector of R_i values, with a total length of six, for comparison with its distance. It is worth noting that the process of determining the corresponding corona for nodes, and thus the appropriate SF, is done only once for static nodes during their network joining phase, which is the case in most of IoT applications. Furthermore, SF ranges and hence corona radiuses are not fixed values and vary greatly depending on the environment. For example, rural environments with possible Line of Sight (LoS) have longer SF ranges than urban environments. Once a node has determined its SF, it will use one of the following three approaches to determine its channel frequency CF and time-slot SID .

6.1.2 Equal Radius-Based Corona (ERBC) Mode

In this mode, we further divide each corona C_i into K equal-width sub-coronas. Then, each sub-corona is labeled with an identifier F_j and assigned a unique channel CF_j , as shown in Fig. 31. According to ERBC mode, a node n that is located on corona C_i and sub-corona F_j will use the SF_i and the frequency CF_j that corresponds to the SF assigned to corona C_i and the channel assigned to sub-corona F_j , respectively. The goal of this partitioning is to mitigate collisions among nodes that are located on a given corona, by spatially distributing frequencies among them. To do so, we firstly find the width L_i of sub-coronas of a given corona C_i as follows:

$$L_i = \frac{R_i - R_{i-1}}{K} \quad (34)$$

where R_i and R_{i-1} are the radii of corona C_i and C_{i-1} respectively and K is the number of the available frequencies. In order for a node n to determine its F_j , and hence its appropriate frequency CF_j , it will use the following formula

$$F_j = \frac{dist_n - R_{i-1}}{L_i} \mod K \quad (35)$$

where $dist_n$ is the distance of node n from the gateway. We assume that each channel is labeled with an ID F_j and once a node determine the channel ID F_j , it can determine the corresponding channel frequency CF_j . For example, in Fig. 31, node n will use the frequency CF_j that is assigned to the sub-corona F_j because it is located in the range of that sub-corona.

Note that nodes located in the same corona and the same sub-corona are using the same SF and the same channel which may cause collisions among simultaneous transmissions. To avoid this, we aim at assigning these nodes different time slots in order to enhance the network throughput. Thus, in order to determine the slot-ID SID for a node, we separate communications among nodes on the same sub-corona F_j using slots with a slot duration compatible with the used SF in corona C_i . In fact, partitioning nodes on the same sub-coronas into different time slots emulates dividing them into sectors with specific angle α_j . The number of needed slots m_j on every sub-corona F_j depends on the node density d of that sub-corona. Furthermore, the frame size, which is $m_j \times T_{oA_{SF}}$, must respect the duty cycle of nodes, which is 1%. To do that, m_j must has a minimum value of 100 as follows

$$m_j = \begin{cases} 100, & \text{if } m_j < 100 \\ A_{F_j} \times d, & \text{if } m_j \geq 100 \end{cases} \quad (36)$$

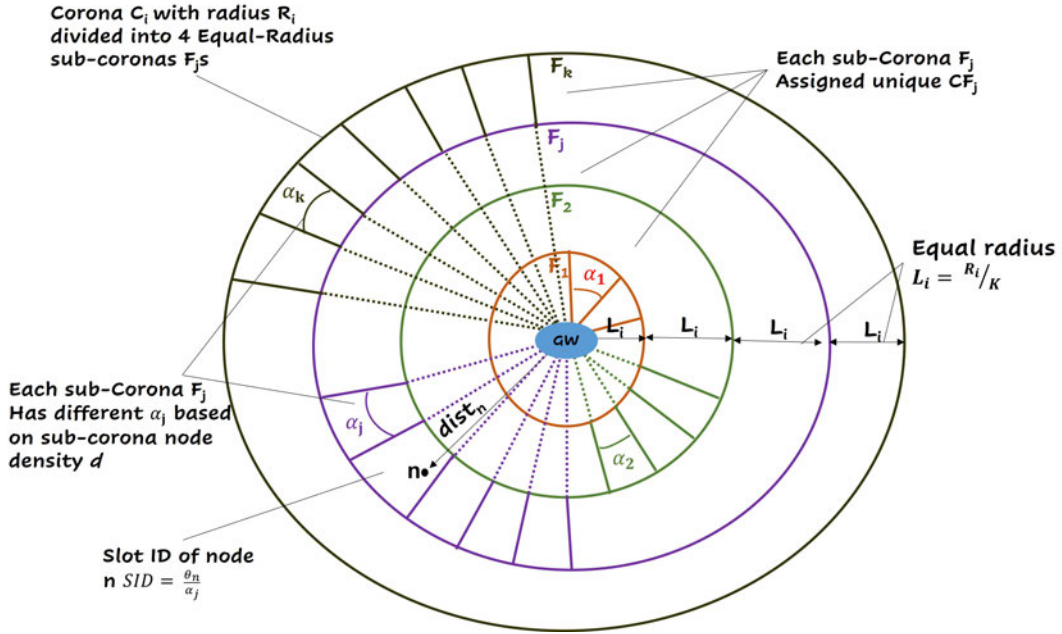


Figure 31: Equal Radius-Based Corona (ERBC) mode example of Corona C_i with radius R_i .

where A_{F_j} is the area of sub-corona F_j and d is the network density. After a node has determined the appropriate m_j , it will simply find the sector's angle $\alpha_j = 2\pi/m_j$. Finally, a node determine its slot-ID SID as follows

$$SID = \frac{\theta_n}{\alpha_j} \quad (37)$$

where θ_n is the node angle relative to the associated gateway. For example, node n in Fig. 31 uses $SID = 3$ based on its θ_n to the gateway.

As shown in Fig. 31, each sub-corona F_j has different area and hence different node density. According to that, the number of needed slots m_j for each sub-corona F_j is different. In other words, inner sub-coronas have smaller area and hence smaller m . However, outer sub-coronas have larger areas and hence larger m .

6.1.3 Equal Area-Based Corona (EABC) Mode

In this mode, each corona C_i is divided into K sub-coronas such that all sub-coronas of a given corona C_i has the same area and hence the same node density. Then, and similar to ERBC mode, each sub-corona F_j is assigned a unique channel CF_j such that all nodes that are located on the

Algorithm 5 ATS-LoRa MAC protocol

- 1: **Input:** node coordinates (X_n, Y_n) , Gateway coordinates (X_G, Y_G) , number of channels K , and node density d
- 2: **Output:** The Spreading Factor SF_i , the Channel CF_j , and the timeslot SID for a given node n
- 3: $CFs \leftarrow [CF_1, CF_2, \dots, CF_K]$
- 4: $R_i \leftarrow [R_1, R_2, \dots, R_i]$
- 5: $dist_n \leftarrow euclideanDistance(X_n, Y_n, X_G, Y_G)$
- 6: # Determine node's SF SF_i
- 7: **for** $i \leftarrow 1$ to 7 **do**
- 8: **if** $dist_n \leq R[i]$ **then**
- 9: $SF_i \leftarrow (i - 1) + 7$
- 10: $C_i \leftarrow i$
- 11: **break**
- 12: **end if**
- 13: **end for**
- 14: **if** ERBC mode **then**
- 15: $L_i \leftarrow (R_i - R_{i-1})/K$
- 16: # Find sub-corona ID F_j
- 17: $F_j \leftarrow \frac{dist_n - R_{i-1}}{L_i} \bmod K$
- 18: # Find area of sub-corona F_j
- 19: $A_{F_j} \leftarrow \pi((R_{i-1} + F_j \times L_i)^2 - (R_{i-1} + F_{j-1} \times L_i)^2)$
- 20: **else**
- 21: # EABC mode
- 22: $F_j \leftarrow FindChannelID(dist_n, C_i, 1, K)$
- 23: $r_{F_j} \leftarrow \sqrt{\frac{F_j(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2}$
- 24: $r_{F_{j-1}} \leftarrow \sqrt{\frac{F_{j-1}(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2}$
- 25: $A_{F_j} \leftarrow \pi(r_{F_j}^2 - r_{F_{j-1}}^2)$
- 26: **end if**
- 27: $CF_j \leftarrow CFs[F_j]$
- 28: $m_j \leftarrow A_{F_j} \times d$
- 29: **if** $m_j < 100$ **then**
- 30: $m_j \leftarrow 100$
- 31: **end if**
- 32: $\alpha_j \leftarrow 2\pi/m_j$
- 33: # Calculate node's theta to the gateway θ_n
- 34: $\theta_n \leftarrow \arctan(Y_n - Y_G/X_n - X_G)$
- 35: $SID \leftarrow \theta_n/\alpha_j$

Algorithm 6 Find Channel ID in EABC mode

```
1: Input: node's distance to the gateway  $dist_n$ , node's Corona ID  $C_i$ , first
   sub-corona ID  $first$ , and last sub-corona ID  $last$ 
2: Output: Node's sub-corona ID  $F_j$ 
3: if  $first == last$  then
4:   return  $first$ 
5: end if
6:  $mid \leftarrow (first + last - 1)/2$ 
7:  $r_{F_j} \leftarrow \sqrt{\frac{mid(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2}$ 
8: if  $dist_n \geq r_{F_j}$  then
9:    $first \leftarrow mid + 1$ 
10:   $FindChannelID(dist_n, C_i, first, last)$ 
11: else
12:   $last \leftarrow mid$ 
13:   $FindChannelID(dist_n, C_i, first, last)$ 
14: end if
```

range of sub-corona F_j will use channel CF_j . By doing that, we can achieve balanced traffic load of a given SF in all channels. In order to have equal area sub-coronas, each sub-corona will have different radius r_j . A node n is considered in the range of sub-corona F_j if it satisfies the following condition: $r_{j-1} < dist_n \leq r_j$, where r_j and r_{j-1} are the radii of sub-corona F_j and F_{j-1} , respectively. r_j is calculated as follows

$$r_j = \sqrt{\frac{F_j(R_i^2 - R_{i-1}^2)}{K} + R_{i-1}^2} \quad (38)$$

As shown in Fig. 32, all sub-coronas have the same area and hence the same node density. According to that, all sub-coronas F_j of a given corona C_i needs the same number of slots m . Hence, as depicted in Fig. 32, all sub-coronas has the same number of slots m . Unlike ERBC mode (Fig. 31), where each sub-corona has different area, and hence different density leading to a different number of slots m . After finding the value of r_j , each node can simply find the area of the sub-corona located in A_{F_j} . Then, similar to ERBC mode, Eq.36 and Eq.37 are used to calculate m_j and SID , respectively. Algorithm 5 demonstrates the main steps of the ATS-LoRa protocol. Furthermore, Algorithm 6 shows the procedure of determining the sub-corona ID F_j in EABC mode.

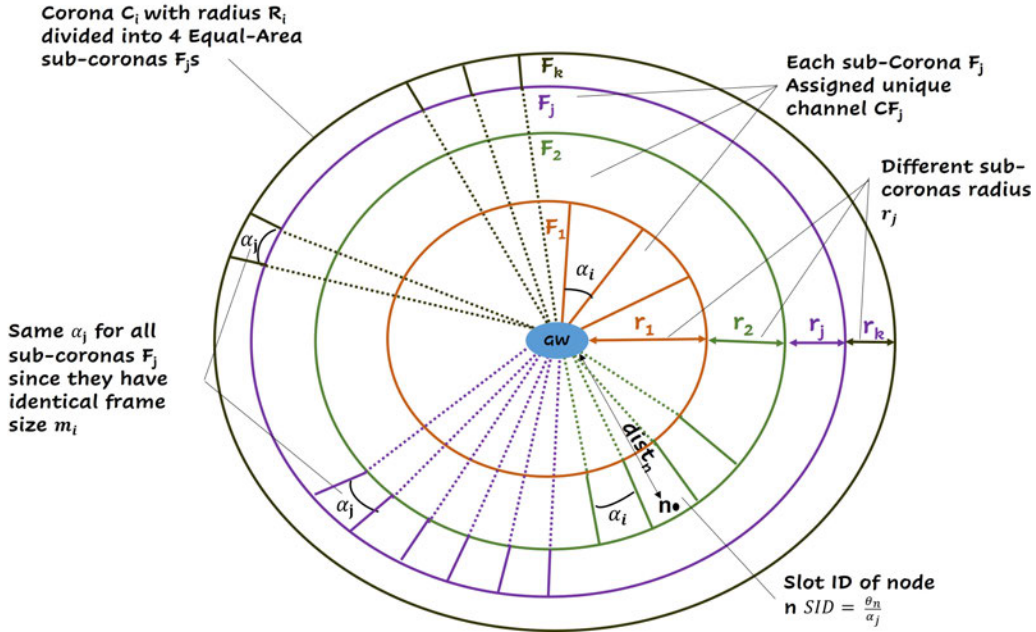


Figure 32: Equal Area-Based Corona (EABC) mode example of Corona C_i with radius R_i .

6.1.4 Equal Area-Based Sector (EABS) Mode

In this mode, instead of dividing coronas into sub-coronas, either with the same width (ERBC mode) or with the same area (EABC mode), we divide the network field into K sectors with angle $\beta = 2\pi/K$ and assign a channel to each sector. Fig.33 shows an example of EABS mode. All nodes located in sector Sec_j will use channel CF_j that is assigned for that sector. In other words, based on node's angle θ_n to the sector's angle β , nodes find their sector IDs Sec_j as follows

$$Sec_j = \frac{\theta_n}{\beta} \quad (39)$$

where $\beta = 2\pi/K$, θ_n is the node's angle to the gateway, and K is the number of the available CFs.

Similar to EABC mode, EABS mode divides the network field into sectors with identical areas. Hence, the traffic load is balanced among the channels. As shown in Fig. 33, $Corona - Sector_{ij}$ refers to the intersection between corona C_i and sector Sec_j . All nodes in the same $Corona - Sector_{ij}$ will use the same SF_i assigned to corona C_i and the same channel CF_j assigned to sector Sec_j . To control channel access between them, we divide each $Corona - Sector_{ij}$ into a grid of $q_{ij} = \sqrt{m_{ij}}$ rows and columns. We assume

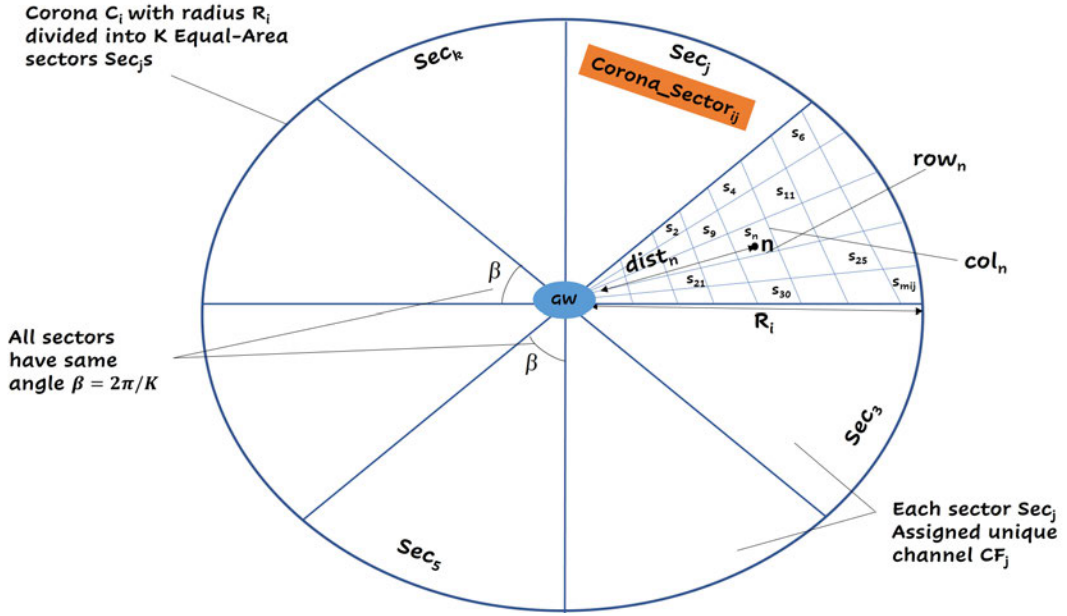


Figure 33: Equal Area-Based Sector (EABS) mode example of Corona C_i with radius R_i .

that each square, which is the intersection of a row and a column, has an identifier ranging from 1 to m_{ij} , which defines the time-slot ID SID . In this mode, m_{ij} is calculated as follows

$$m_{ij} = \begin{cases} 100, & \text{if } m_{ij} < 100 \\ \frac{\beta}{2}(R_i^2 - R_{i-1}^2) \times d, & \text{if } m_{ij} \geq 100 \end{cases} \quad (40)$$

where R_i and R_{i-1} are the radii of corona C_i and C_{i-1} respectively and d is the network density. As shown in Eq.40, the number of needed timeslots per frame or the frame size m_{ij} mainly depends on the node density of a given *Corona – Sector* $_{ij}$. Hence, frame sizes are dynamic and associated to the network density of a given area with common transmission parameters. However, when the network density d of a given *Corona – Sector* $_{ij}$ is not large, short frame sizes m are resulted. If this is the case, we may get frame sizes that is shorter than the DC limit between consecutive transmissions. To overcome that, frame sizes m_{ij} has a lower limit, which is 100 slots per frame, to respect the duty cycle of nodes, which is 1%. By doing that, we take advantage of the waiting time between the successive transmissions by dividing it into timeslots and assign them to nodes that belong to a specific *Corona – Sector* $_{ij}$.

To determine node's SID , a node must firstly determine which square it

is located in. Hence each node computes its row_n and col_n as follows

$$\begin{aligned} row_n &= \frac{q_{ij}(dist_n - R_{i-1})}{R_i - R_{i-1}} \\ col_n &= \theta_n \quad \text{mod} \quad \frac{\beta}{q_{ij}} \end{aligned} \tag{41}$$

where q_{ij} is the number of rows and columns in $Corona - Sector_{ij}$, $dist_n$ is the distance between the node n and the associated gateway, R_i and R_{i-1} are the radii of corona C_i and C_{i-1} respectively. Then, time-slot SID can be retrieved as follows

$$SID = (row_n - 1) \times q_{ij} + col_n \tag{42}$$

According to the proposed algorithms, each channel has six frames, corresponding to the number of the available SFs as shown in Fig. 34. The frame size of a given SF on a given channel depends on the node density of the area that use that channel with that SF. In other words, the number of needed slots m_j differ among channels for a given SF depending on the node density of the area that is assigned a given channel CF_j . For example, as shown in Fig. 34, the number of needed slots m_j for SF_7 is different from channel to another. Furthermore, the slot duration is different from frame to another depending on the used SF. Hence, the proposed protocol is completely dynamic as it has dynamic frame sizes and dynamic timeslot duration.

6.1.5 Frame Sizes Comparison

This section emphasizes on the main feature of the proposed algorithm which is the small frame sizes compared to the number of connected nodes N . As mentioned earlier, Time-Slotted MAC protocols generally relate frame sizes to the total number of connected nodes. Hence, with some approaches the increase of the number of nodes will inevitably proportionally increase the frame size which may result in extremely large values as LoRa networks are designed to support large number of nodes. However, in ATS-LoRa protocol with the three operational modes, the increase in the frame size is not related to the increase of the total number of connected nodes. Alternatively, the increase of the frame size is related to the increase of the number of nodes that use common transmission parameters, such as common SF and channel CF, so to mitigate collisions among them. Fig.35 shows a comparison between the frame size of SBTS-LoRa protocol proposed in chapter 5 and the ATS-LoRa protocol proposed in this chapter with its three modes. To better clarify the

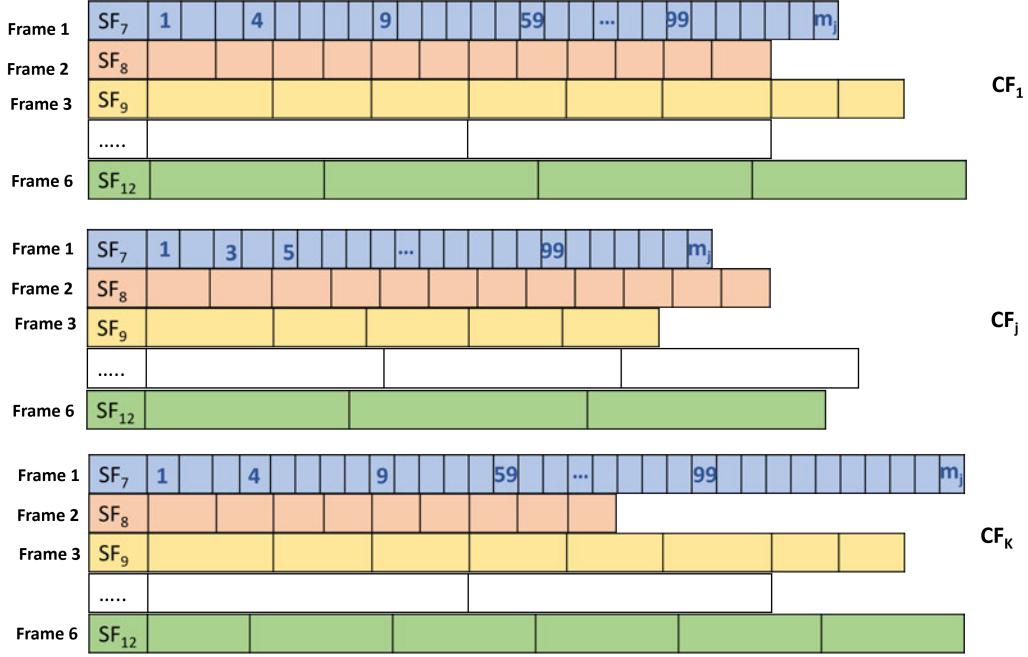


Figure 34: Time frames per channel CF.

frame sizes, Fig.35.a shows the frame sizes of ATS-LoRa in ERBC, EABC, and EABS modes as function of the number of connected nodes. Fig.35.b shows the frame sizes of SBTS-LoRa protocol as function of the number of connected nodes. The number of nodes is ranging from 1000 to 10000 nodes. The field radius is around 9 Km. Fig.35 shows the frame sizes of the last and largest corona that has $SF = 12$. As shown in Fig. 35.a, the frame size of all three modes is identical for small networks (< 2000). However, for larger number of connected nodes, ATS-LoRa with EABC or EABS modes has lower frame sizes than the ATS-LoRa with ERBC mode. This is because the former modes divide the coronas into equal-area sub-coronas or sectors, respectively. Hence, the number of nodes that use common SFs and channels and thus require timeslots separation, is more balanced. In this simulation, we consider the last corona because it is the most challenging one as it has the highest SF with the longest end to end delay and highest energy consumption. Hence, with a payload size equals $50B$ and a number of connected nodes $N = 10000$, the waiting time between two successive transmissions is only 15.7 minutes in EABC mode and 17 minutes in ERBC and EABS modes. The waiting time is calculated as follows:

$$T_{Wait} = (m - 1) \times T_{Slot} \quad (43)$$

where m is the frame size and T_{Slot} is the slot duration, which is in this example equals ToA_{SF12} , since we consider it for only the last corona with $SF = 12$.

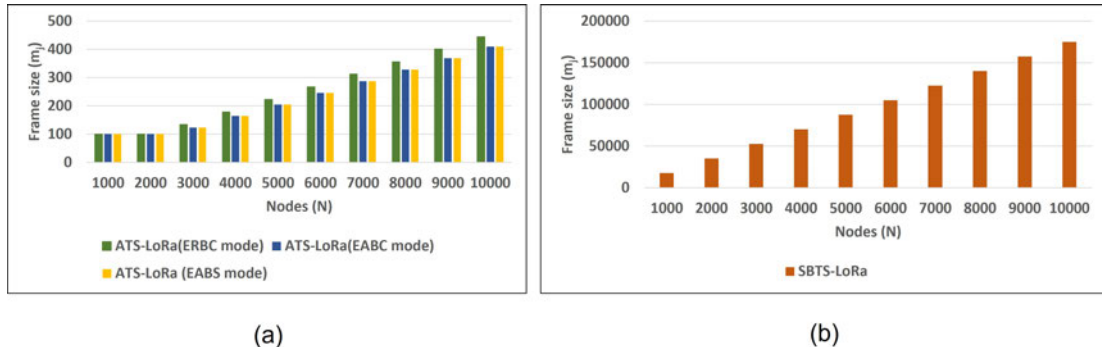


Figure 35: Comparison between the frame sizes of ATS-LoRa protocol in ERBC, EABC, and EABS modes and SBTS-LoRa protocol [4].

6.2 Performance Evaluation

In this section, the proposed MAC protocol with its three modes is evaluated using OMNET++ simulator [119] and FLoRa framework [120]. A star network topology is assumed where the gateway is located at the center of the network field. The nodes are distributed at random around the gateway, with a maximum distance corresponding to the field radius R . The radius of the field and the radius of SF coronas R_i are field-dependent, which means that they vary depending on the deployed environment. Accordingly, this simulation assumes a suburban environment and employs the well-known log-distance path loss model with shadowing [133] as expressed in the following equation

$$PL(d) = \overline{PL}(d_0) + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma \quad (44)$$

where $\overline{PL}(d_0)$ is the mean path loss for distance d_0 , n is the path loss exponent, and X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ . The values used in the simulations for these parameters are shown in Table 20. ATS-LoRa protocol assumes a single packet transmission per frame with a packet length of 20 bytes. The protocol imposes a minimum frame size of $100 \times ToA_{SF}$ as explained in section 6.1. However, there is no limit for the maximum frame size, as the frame size remain reasonable even with large number of connected nodes, as demon-

strated in section 6.1.5, thanks to the possible parallel frames per channel and SF.

Table 20: Simulation parameters used to evaluate the ATS-LoRa.

Parameter	Value	Comments
$\overline{PL}(d_0)$	128.95	Mean path loss for distance $d_0 = 1000m$
n	1.5	Path loss exponent
σ	0	
CF	{868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, 867.9}	Carrier Frequencies (MHz)
SF	7 to 12	Spreading factors
TP	14 dBm	Transmission powers
CR	4/5	Coding rate
BW	125kHz	Bandwidth
R	8921m	Field radius
N	1000 - 4000	Number of nodes
Simulation time	11	Days

According to the proposed protocol, the network field is divided into six coronas that are assigned, each, a given SF. Since the smallest SF, SF7, has the lowest transmission range, the algorithm starts assigning the smallest SF to the closest corona to the gateway and so on. Each corona's radius corresponds to the maximum eligible distance for nodes to use the assigned SF. The goal of this partitioning is to avoid assigning a node a SF that will prevent the successful delivery to the gateway. As a result, nodes will avoid consuming their energy for hardly delivered packets. We conduct a simulation using the OMNET++ simulator to determine the maximum eligible distance for a given SF such that the corresponding transmissions are not received below the sensitivity level of the gateway. Specifically, the simulation consists of one node and one gateway. Firstly, a node is assigned a given SF value starting from the smallest one, SF7, to the largest one, SF12. Then, on each run, the node is moving farther from the gateway while sending packets to the gateway. The maximum eligible distance for a given SF is the maximum distance for the node in which the gateway can successfully receive packets above the gateway sensitivity level. The simulation is repeated for all the SF

values. Table 21 exhibits the maximum allowable distance for each SF with a transmission power equals 14 dBm and a path loss model parameters as indicated in Table 20. Furthermore, Table 21 shows the sensitivity levels that is configured at the gateway as specified in [1]. The following sections evaluate the performance of the proposed protocol with its three modes compared to The SBTS-LoRa protocol and LoRaWAN-ADR in terms of the probability of collision, the end-to-end delay, the network throughput, and the energy consumption.

6.2.1 The Probability of Collisions

Fig.36 shows the probability of collision as function of the total number of connected nodes. As shown in this figure, ERBC and EABC modes achieves the lowest collision rates. On the other hand, EABS mode achieves the highest collision rate that close to the collision rate of LoRaWAN. This is due to the slots assignment procedure among nodes. In fact, in all modes, the total number of the available time-slots per channel m_{ij} is related to the estimated node density of a given sub-corona or *Corona – Sector_{ij}* (see Eq.36 and Eq.40). However, in ERBC, EABC and SBTS-LoRa, the assignment of time-slots corresponds also to node’s relative position angle ”theta” to the gateway. Hence, there is a low probability to have two or more nodes on the same sub-corona and with the same ”theta”. As shown in Fig.36, the collision rate is zero in ERBC and EABC modes with $N = 1000$, which is a quite large number of nodes. Furthermore, in ERBC and EABC modes, the collision rates are slowly increasing with the increase of the number of connected nodes compared to EABS mode. In other words, although each *Corona – Sector_{ij}* in EABS mode have a frame size that corresponds to its need, similar to ERBC and EABC modes, two or more nodes can have the same time-slot ID *SID* if they are located on the same square (see Eq.42).

Table 21: MAX eligible distance for each spreading factor SF [1].

SF	MAX eligible distance (m)	Sensitivity (S)
7	2450	-124
8	3306	-127
9	4450	-130
10	5998	-133
11	7316	-135
12	8921	-137

In other words, the assignment of time-slot IDs does not depend on nodes' theta like other modes. In EABS mode, the time-slot assignment depends on which square a node is located. In fact, the probability of having more nodes with the same SID is increasing with the increase of the number of nodes. Furthermore, nodes start their transmissions usually at the beginning of their time-slots without any random back off. Hence, destructive collisions are happened with nodes located on the same $Corona - Sector_{ij}$ and use the same SID . This explains the highly increasing collision rate of EABS mode.

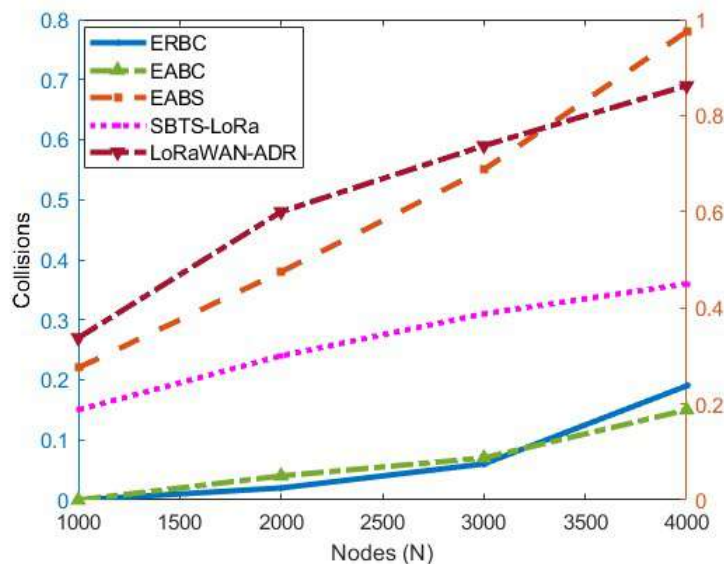


Figure 36: Probability of collisions of ATS-LoRa compared to SBTS-LoRa and ADR-LoRaWAN.

6.2.2 The End-to-End Delay

Fig.37 shows the end-to-end delay as function of the number of connected nodes. As shown in Fig.37, all modes of the ATS-LoRa protocol achieve lower end-to-end delay than SBTS-LoRa and LoRaWAN protocols. In fact, the used SF of each node has a significant impact on the end-to-end delay of their transmissions. Smaller SFs have shorter ToA and thus less delay. As a result, from a delay standpoint, it is preferable to use lower SFs whenever they are allowable. To better explain the end delay, Fig.38 shows the distribution of SFs among nodes in both ATS-LoRa and SBTS-LoRa protocols. As shown in the figure, SBTS-LoRa algorithm use larger SFs more than the ATS-LoRa.

This explains the longer delay of SBTS-LoRa algorithm. Furthermore, the end-to-end delay is only considered for the successfully received packets by the gateway. That's why the end-to-end delay decreases with large number of nodes as the number of successfully received packets will be reduced due to high collision rate. More precisely, the closest nodes to the gateway will be the ones that will succeed to deliver their packets to the gateway as they have the shortest time on air which will explain the reduced end-to-end delay. Hence, EABS mode achieves the lowest end to end delay as it has the highest collision rate.

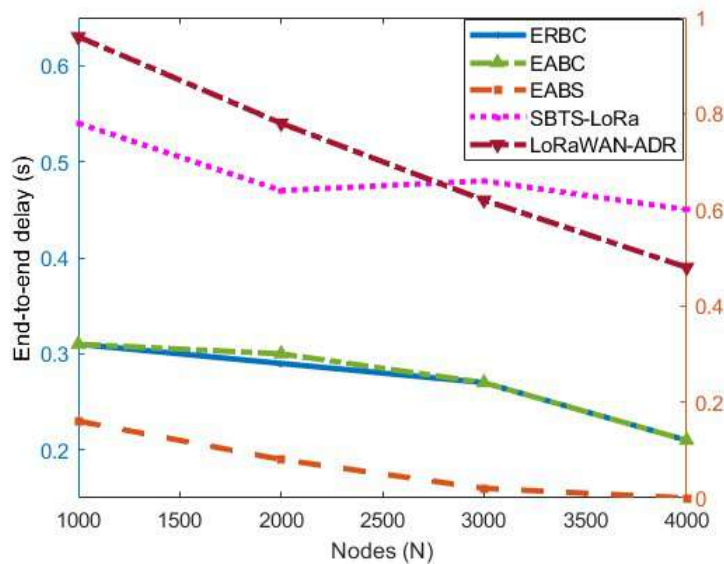


Figure 37: The end-to-end delay of ATS-LoRa protocol.

6.2.3 Network Throughput

Fig.39 shows the network throughput as function of the number of connected nodes. The results show that the end-to-end delay and collision rate have a significant impact on network throughput. Indeed, as more traffic is generated in the network, the network throughput increases in proportion to the number of connected nodes until it reaches a maximum. The network throughput then begins to deteriorate as the probability of collision takes over the successfully received packets, resulting in a decrease in network throughput. As shown in Fig.39, The ATS-LoRa protocol outperforms the SBTS-LoRa and LoRaWAN protocols in terms of throughput. Specifically, ERBC and EABC modes achieve the highest network throughput since they

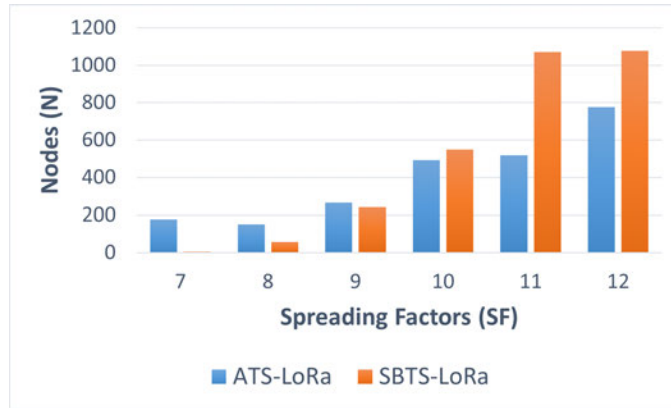


Figure 38: Comparison between SF distribution in SBTS-LoRa and ATS-LoRa protocols.

have the lowest collision rate. In ERBC and EABC modes, the network throughput reaches 54 packets/second when the number of connected nodes equals 3000 nodes. The optimal throughput results can be attributed to the efficient distribution of the transmission parameters that resulted in the lowest collision rate and end-to-end delay. On the other hand, although the EABS mode has the worst collision rate among other ATS-LoRa modes, it still has better network throughput than SBTS-LoRa and LoRaWAN protocols due to the efficient distribution of SFs compared to them as ATS-LoRa protocol uses the minimum SF whenever possible. Hence, the Time on Air (ToA) is reduced. Furthermore, ATS-LoRa has smaller frame sizes compared to SBTS-LoRa protocol, as shown in Fig.35. Hence, the waiting time between successive data transmissions is reduced.

6.2.4 The Energy Consumption

Fig.40 depicts the energy required to successfully transmit one bit as a function of the network size. Obviously, the ATS-LoRa protocol with EABS mode consumes the highest energy since it has the highest collision rate. ERBC and EABC modes of ATS-LoRa protocol achieve similar energy consumption levels, which is expected, as they have similar collision rates, end-to-end delay, and network throughput. It is, however, notable that ERBC and EABC modes are showing almost stable energy consumption with the increase in the number of connected nodes. This behaviour is critical as it demonstrates the scalability and robustness of these modes. The results also show that ATS-LoRa protocol consumes more energy compared to the SBTS-LoRa protocol. However, this is justified as the number of successfully delivered packets is

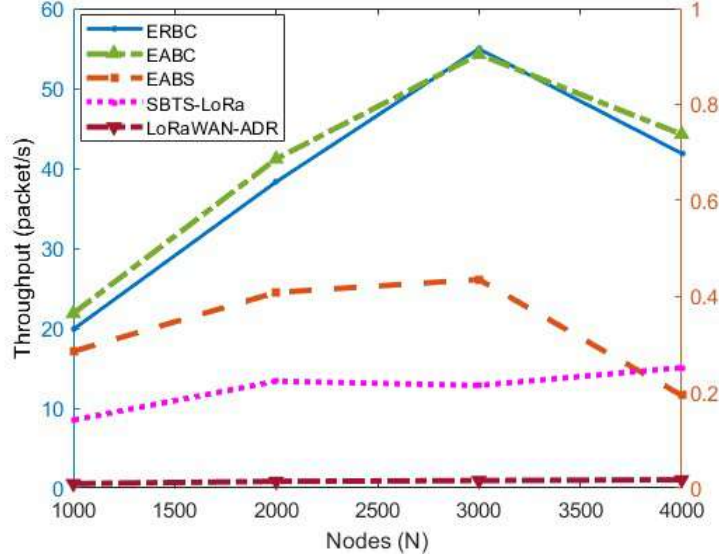


Figure 39: Network throughput of ATS-LoRa protocol.

higher in the ATS-LoRa protocol, as shown in Fig.39. We assume one packet transmission per frame in this simulation. Unlike SBTS-LoRa, the frame size in ATS-LoRa is determined by the node density of a given sub-corona and is not fixed. As a result, the ATS-LoRa protocol has a small frame size without compromising the node's duty cycle.

6.3 Conclusion

This chapter presents a novel LoRaWAN TDMA MAC protocol that autonomously configures nodes with the appropriate transmission parameters (e.g., SF, Channel, times lot) utilizing the node's relative position to the gateway. To achieve this, the proposed protocol divides the network field into sections of corona and sectors and assigns specific parameters for each one of them. A node located in a specific range of corona or sector will use the parameters that are virtually assigned to that section. Specifically, the network field is divided into six coronas that corresponds to the supported six SFs by the LoRa physical layer. Each corona is assigned a specific SF starting with the smallest SF for the closest corona to the gateway. A node located on a given corona will use the corresponding SF for that corona. The corona radius is dynamic and reflects the maximum allowable distance for a node to use a given SF such that its transmission is received above the sensitivity level of the gateway. According to that, the Packet Error Rate (PER)

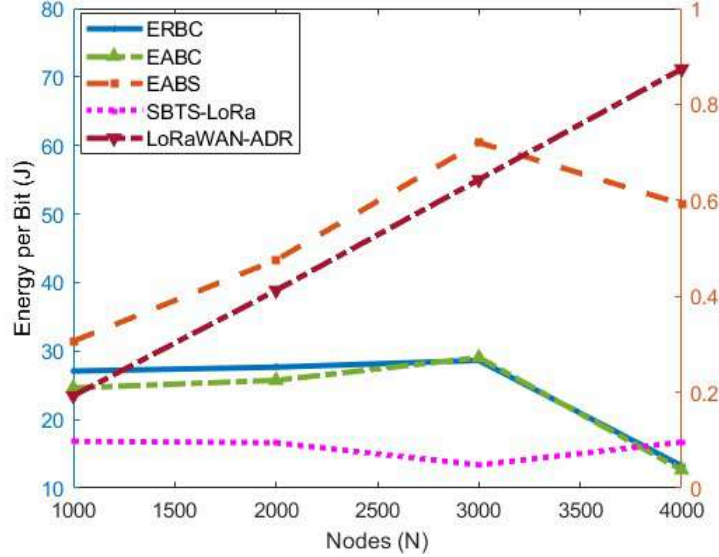


Figure 40: Energy consumption of ATS-LoRa protocol.

in the proposed protocol is almost zero. The assignment of channels and timeslots is done using three variant approaches, namely, Equal Radius-based Corona (ERBC), Equal Area-based Corona (EABC), and Equal Area-based Sector (EABS) showing outperforming results in terms of end-to-end delay and throughput compared to SBTS-LoRa and ADR LoRaWAN. Hence, the proposed protocol with all its variants is more scalable than the SBTS-LoRa and the LoRaWAN.

For future work, further improvements could be applied to enhance the approach in determining the optimal SF ranges. In other words, we aim at leveraging some AI techniques to determine the optimal SF for each node without compromising the end-to-end delay, the collision rate, and the energy consumption. Moreover, the proposed algorithms can be extended to be adapted for mobile nodes with multiple gateways. In this regard, further enhancements are needed during the initialization of nodes and the assignment of the appropriate transmission parameters for them. In other words, further techniques could be implemented to determine when a node need to change their assigned transmission parameters based on their updated locations.

7 Conclusion

This chapter highlights the main findings in this thesis, briefly reviewing the main made contributions and the associated results as well as proposing some potential future research directions.

7.1 Concluding Remarks

LoRa networks offer appealing characteristics such as the long transmission range, the low cost, the low energy consumption, and the low data rates. These characteristics make them ideal for a wide range of IoT applications that require huge connection across large transmission ranges such as smart cities, smart agriculture, and smart metering. According to LoRa alliance, LoRa networks can connect hundreds of thousands of nodes with low traffic rates over large transmission ranges. However, real-life deployments and modeling investigations have revealed that the stated performance in large scale networks with a high number of connected nodes is far from being fulfilled. This is mainly due to the significant interference caused by the usage of the shared free unlicensed band, as well as the high collision rates caused by the use of the ALOHA channel access. Furthermore, LoRa networks are bound by the duty cycle aspect of the open band, which is crucial for down-link traffic. Moreover, LoRa physical layer provides different transmission parameters such as different orthogonal Spreading Factors (SFs), different channel frequencies (CFs), and different transmission power (Tx) levels. In fact, simultaneous transmissions with different transmission parameters would result in successful reception of all of them. According to that, configuring nodes with the appropriate transmission parameters would decrease the collision rates and hence the network throughput and scalability are improved. Based on that, the ultimate aim of this thesis was to address the main challenges of LoRa networks and conceive comprehensive solutions considering all these challenges such that the network scalability and the energy efficiency are maximized. To achieve that aim, the following objectives were defined:

The first objective was to provide a thorough review of the technical aspects of LoRa networks, as explained in Chapter 2. In Chapter 2, a review about the main characteristics of LoRa physical layer, MAC, and Data Link layers was provided. Furthermore, the chapter introduced an overview about other LPWA technologies and compare their technical features with LoRa. Moreover, the chapter summarized some of the large-scale real-deployed LoRa projects in some areas such as in smart cities, smart

agriculture, and smart metering. The goal of this review is to provide a clear sight about the possibilities and the limits of LoRa networks in real deployment scenarios.

The second objective of this thesis was to provide a thorough review of the related work to identify the main research gaps. According to that, Chapter 3 reviewed the research on enhancing the capacity and scalability of LoRa networks, focusing on the MAC (Medium Access Control) and data link layers. Unlike other surveys, this one focused on these layers since they are critical in regulating collision rates, which have a considerable influence on network scalability. The chapter provided a thorough assessment of the literature, organized around significant constraints that might impede the network's ability to reach its performance goals, represented in the random access of the medium, the limited duty cycle of downlink traffic from the server to nodes, and configuring nodes with appropriate transmission parameters.

Enlightened by the review conducted in Chapter 2 and Chapter 3, the following research gaps were identified:

1. Conceiving a light distribution algorithm for the transmission parameters of LoRa nodes such that the network scalability is maximized without violating the energy efficiency and the limited duty cycle.
2. Designing a decentralized Time-Division Multiple Access (TDMA) protocol that works as an alternative to the random ALOHA channel access such that it does not require extensive downlink traffic to disseminate the schedule among nodes or synchronize them with the server.
3. Proposing a TDMA MAC protocol with dynamic frame sizes and time-slot duration, which conform to the dynamic nature of LoRa networks as they support different transmission times and ranges.

According to the first identified gap, the third objective was to conceive a distribution algorithm of LoRa transmission parameters that maximize the network scalability considering the main limitations of LoRa networks, as detailed in Chapter 4. In this chapter, we introduced Sensitivity-Aware LoRa Configuration (SAL), a new algorithm for efficient autonomous and distributed selection of LoRa physical layer transmission parameters. The aim was to address the limitations of the currently adopted MAC algorithm in LoRaWAN networks (i.e., Adaptive Data Rate - ADR). The selection of the transmission parameters in ADR is done randomly by the gateway, an approach that may result in the gateway reaching its Duty Cycle Limit, consequently hindering it from sending the configuration information to the end points under large-scale networks which will negatively affect the network performance. Unlike ADR, SAL used a decentralized approach to select

node's transmission parameters without any need for gateway's control packets and it only considers a combination of parameters that is guaranteed to be received successfully by the gateway. Specifically, SAL used all the supported channels and SFs to maximize the potential successful simultaneous transmissions considering the different DC of each band. The performance of the proposed algorithm is validated through extensive simulations under different scenarios and operation conditions showing promising results. According to the simulation results, the average enhancements in the PDR of SAL algorithm compared to the ADR-LoRaWAN was 320%.

The fourth objective, inline with the second identified gap, was to design a medium access mechanism that maximize the network scalability while taking into account the limited duty cycle of LoRa nodes, as described in Chapter 5. The chapter presented the Sector-Based Time-Slotted (SBTS-LoRa) protocol, a novel Time Division Multiple Access (TDMA) MAC protocol for LoRa networks. The aim was to address the main limitations of LoRa networks without compromising the energy efficiency. The first contribution of this chapter was the designing and deploying of the SBTS-LoRa protocol as a novel MAC protocol to overcome key LoRa challenges such as the capture effect, the limited scalability, and the duty cycle. SBTS-LoRa replaced the ALOHA random access method with a TDMA access method. It allowed each node to autonomously determine its transmission parameters by knowing only the gateway location and its own location. To do that, SBTS-LoRa firstly divided the network field into a set of annulus cells. Each cell will have a unique frequency, a set of eligible SFs, and a specific transmission power level. Nodes that are located on a specific cell boundary will use the transmission parameters assigned for that cell. Moreover, SBTS-LoRa divided each cell into a set of sectors. The sector ID to which a node belongs is the timeslot ID for which a node is allowed to transmit. The main novelty of SBTS-LoRa was the use of a decentralized approach that accurately and efficiently determine node's transmission parameters without burden the network with extensive control packets from the network server.

The second contribution of Chapter 5 was the conducting of extensive analysis and simulations using OMNET++ simulator under FLoRa framework to explore the performance of the SBTS-LoRa under different operating conditions [120]. In order to critically evaluate the proposed protocol, we have implemented EXPLoRa-AT [123] and [109] besides the Adaptive Data Rate (ADR) of LoRaWAN [8] protocol. We evaluated the performance of all protocols in large-scale extremely dense networks, where nodes are distributed in an area with a maximum distance of 14 km from the gateway and the number of connected nodes ranges from 1000-5000 nodes. To the best of our knowledge, no proposed protocol for LoRa networks is evaluated under

these challenged conditions and considering all LoRa transmission parameters. The simulation results showed the average throughput of the proposed protocol equals 13 packets/s, compared to the average throughput of the ADR-LoRaWAN of only 0.9 packets/s, with an average enhancement that equals 1344%. Regarding the energy efficiency, the average consumed energy per bit in SBTS-LoRa was 15.49 J, where the average energy consumption per bit in ADR-LoRaWAN was 54.85 J, resulting in an average energy efficiency of 71%.

The last objective, in line with the third identified research gap, was to design a Time-Division Multiple Access (TDMA) protocol with dynamic frame sizes, as explained in Chapter 6. In this chapter, we proposed an Autonomous Time-Slotted MAC protocol (ATS-LoRa) that allows LoRa nodes to individually determine their transmission parameters without extensive downlink transmissions from the gateway. In our novel approach, the nodes autonomously calculate their most efficient transmission parameters as a function of the location information of nodes and gateways. The autonomous selection of these parameters without the involvement of the gateway greatly enhances the network performance in terms of collision rate, throughput, end-to-end delay, and energy efficiency. The first contribution of this chapter was the proposing of a TDMA based MAC protocol that allows nodes to independently determine the appropriate transmission parameters without requiring any downlink traffic from the gateway. In addition, the proposed protocol instructed nodes to utilize the minimum SF whenever eligible. This is a very important feature in enhancing the transmission time, end to end delay, and energy consumption. In this context, eligibility means using the minimum SF to guarantee that the transmitted packet will not be received below the GW's sensitivity level. As long as all nodes use the appropriate SF to guarantee the delivery of their packets, the proposed protocol has a Packet Error Rate (PER) of almost zero. Moreover, in ATS-LoRa protocol, any channel has six different frame sizes corresponding to the available six SFs. The frame size has lower and upper boundaries taking into account nodes duty cycles. Furthermore, the frame size is dynamic, and it mainly depends on the node density on a given channel with a given SF. Simulation results show that the ATS-LoRa protocol achieves a frame size that is only 150 slots with a very large number of connected nodes (4000 nodes). This is a very important feature that highly reduces the end-to-end delay and hence the network throughput is enhanced. Indeed, in the previous proposed protocol SBTS-LoRa in Chapter 5, although it considerably reduces the collision rate compared to LoRaWAN ADR algorithm, the large frame size in dense networks can be further investigated. In other words, with a larger number of connected nodes (5000 nodes) the frame size in SBTS-LoRa was quite

large (up to 2400 slots). In this enhanced version, we aimed at considerably reducing the frame sizes. In fact, one of the main limitations found in the literature is the fixed or large frame sizes of the proposed TDMA approaches.

The second contribution of this chapter was the conducting of a large-scale simulations using OMNET++ [119] simulator under FLoRa framework [120]. The goal of the simulations was to investigate the capacity of the network with large number of connected nodes. The average network throughput of ATS-LoRa protocol with Equal Area-Based Corona mode (EABC) was 40 packets/s compared to the previous proposed protocol SBTS-LoRa of 13 packets/s, which an average enhancement of 207%.

7.2 Future Work Perspectives

Based on our previous work on enhancing LoRaWAN , we are now in a position to answer the research questions of this thesis. In this regard, we articulate that our attempts to address LoRa limitations have only partially been successful and there are still a need for further research efforts as summarized in the following points:

7.2.1 ALOHA Medium Access Mechanism Alternatives

First, it is worth pointing out that the ALOHA medium access issue is well investigated in the literature, as described in Section 3.1. Moreover, it is undeniable that the Aloha-based access is challenging in large-scale LoRa networks as it may cause high collision rates. However, we should emphasize also that the ALOHA channel access is not always a challenge or a limitation as it has a simple, energy-efficient implementation that makes it the most suitable way to access the medium for sensors with limited processing and power resources. Hence, the ALOHA channel access is perfectly fitted for low-density networks with low traffic rates. However, adopting some techniques to dynamically switch from the ALOHA channel access to other access techniques, like slotted ALOHA or TDMA, when the network is getting congested may greatly enhance the network performance. In other words, allowing the network itself to decide its channel access method based on the current observed network traffic will result in smart networks that change their channel access on demand based on the observed performance.

7.2.2 Addressing the Limited Duty Cycle

Similar to the ALOHA channel access, the limited duty cycle for downlink traffic is well investigated as explained in section 3.2. Nevertheless, the re-

search concerning the compression techniques employed by the gateway to reduce the size of downlink packets remains partially unresolved. We emphasize that focus of future efforts should revolve around proposing lightweight decompression techniques at the end nodes, given their restricted processing capabilities. Indeed, advocating compression/decompression methods that require substantial processing power and longer processing times may not be viable for networks of this nature.

Another method to deal with the limited DC for downlink transmissions' is gateway densification. We believe that this research area needs more investigation as it will greatly help in increasing the network capacity. However, inefficient solutions may result in an unnecessary extensive budget or disappointing scalability results. Hence, smart decisions about the optimal number of gateways with their optimal locations for a given network based on its needs will greatly enhance network scalability.

7.2.3 Configuration of Node's Parameters

As discussed in Section 3.3, different techniques have been proposed to approach the optimal distribution of all the transmission parameters namely; SFs, CF, TX, BW and CR. However, there is still room for further improvements as the main goal is to reach the optimal configuration while being energy efficient with minimum computational overhead. Consequently, the use of distributed AI techniques that can be implemented at the node side is recommended. Indeed, adopting for instance multi-agent distributed algorithms will mitigate the burden at the gateway side and most importantly mitigate the needs of extensive downlink traffic. However, careful investigations are required as most of the distributed AI techniques proposed in the literature are addressing the distribution of one or few transmission parameters without providing comprehensive solutions that deal with all available transmission parameters. It is true that by doing so low computational overhead is guaranteed at the node side but at the same time suboptimal configuration is achieved. Therefore, it is highly recommended to propose lightweight distributed solutions that deal with the optimal configuration of all the transmission parameters. Network slicing can also be a good candidate to achieve such a goal especially in heterogeneous LoRa networks that accommodate different applications with different quality of services.

Network slicing is how the network is virtually divided, sliced, so nodes with different Quality of Service (QoS) requirements are isolated from each other. This virtual slicing is helpful in sharing the common network physical resources efficiently between different virtual network slices that have different QoS requirements. Network slicing can help in creating efficient

heterogeneous networks that accommodate different applications with different QoS requirements [134]. Hence, physical network resources can be available on demand or as a service for nodes. In LoRa networks, physical resources can be seen as the transmission parameters since every combination of transmission parameters leads to the use of a given channel with a given time on air and hence a given packet delivery ratio and energy consumption. Thus, the careful adaptation of the network slicing methods for LoRa should be performed due to the limited resources of such networks [135].

Last but not least, we recommend considering realistic network deployment if any solution from the aforementioned research directions is investigated. Indeed, we observed that most of the already proposed works in the literature assume ideal environments in terms of the interference and the propagation model. This will result in unpractical solutions that do not guarantee the claimed promised performance level. Furthermore, most of the proposed methods assume that all LoRa nodes in the network are identical in terms of the data traffic periodicity, priority, and tasks. However, in real scenarios, we may have a network that accommodates a variety of sensors that work for different applications and hence have different characteristics in terms of the data periodicity and the priority of their traffic. Moreover, most of the related work assumes a single-gateway network for simplicity. However, in real situations, multiple gateways are commonly needed to enhance the scalability of the network. Hence, considering more realistic environments, either through the use of real testbeds or by improving the current simulators is still unresolved and needs more effort.

References

- [1] Semtech. LoRa SX1272/73 Datasheet. URL <https://shorturl.at/B0T01>.
- [2] J.E. Shuda, A.J. Rix, and M.J. Booyesen. Towards Module-Level Performance and Health Monitoring of Solar PV Plants Using LoRa Wireless Sensor Networks. In *2018 IEEE PES/IAS PowerAfrica*, pages 172–177, June 2018. doi: 10.1109/PowerAfrica.2018.8521179.
- [3] Semtech. SX1272/3/6/7/8: LoRa Modem Designer’s Guide, 2013. URL <https://shorturl.at/dG0P4>.
- [4] Hanan Alahmadi, Fatma Bouabdallah, and Ahmed Al-Dubai. A novel time-slotted LoRa MAC protocol for scalable IoT networks. *Future Generation Computer Systems*, 134:287–302, September 2022. ISSN 0167-739X. doi: 10.1016/j.future.2022.04.003. URL <https://www.sciencedirect.com/science/article/pii/S0167739X22001261>.
- [5] J. Lee, Y. Kim, Y. Kwak, J. Zhang, A. Papasakellariou, T. Novlan, C. Sun, and Y. Li. LTE-advanced in 3GPP Rel -13/14: an evolution toward 5G. *IEEE Communications Magazine*, 54(3):36–42, March 2016. ISSN 1558-1896. doi: 10.1109/MCOM.2016.7432169.
- [6] Almudena Díaz Zayas and Pedro Merino. The 3GPP NB-IoT system architecture for the Internet of Things. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 277–282, May 2017. doi: 10.1109/ICCW.2017.7962670. ISSN: 2474-9133.
- [7] Sigfox networking guide, . URL <https://shorturl.at/ftuzF>.
- [8] LoRa Alliance. What is LoRaWAN. URL <https://shorturl.at/dFJPW>.
- [9] 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, 2020. ISSN 1553-877X. doi: 10.1109/COMST.2019.2949598.
- [10] Zehua Sun, Huanqi Yang, Kai Liu, Zhimeng Yin, Zhenjiang Li, and Weitao Xu. Recent Advances in LoRa: A Comprehensive Survey. *ACM Transactions on Sensor Networks*, June 2022. ISSN 1550-4859. doi: 10.1145/3543856. URL <https://doi.org/10.1145/3543856>.

-
- [11] About LoRaWAN® | LoRa Alliance®, . URL <https://loro-alliance.org/about-lorawan>.
- [12] Tommaso Polonelli, Davide Brunelli, and Luca Benini. Slotted ALOHA Overlay on LoRaWAN - A Distributed Synchronization Approach. In *2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 129–132, Bucharest, October 2018. IEEE. ISBN 978-1-5386-8296-8. doi: 10.1109/EUC.2018.00026. URL <https://ieeexplore.ieee.org/document/8588859/>.
- [13] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, 55(9):34–40, 2017. ISSN 0163-6804. doi: 10.1109/MCOM.2017.1600613. URL <http://ieeexplore.ieee.org/document/8030482/>.
- [14] Understanding ADR | DEVELOPER PORTAL, . URL <https://loro-developers.semtech.com/documentation/tech-papers-and-guides/understanding-adr/>.
- [15] 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, December 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2019.2942776.
- [16] Junhee Lee, Young Seog Yoon, Hyun Woo Oh, and Kwang Roh Park. DG-LoRa: Deterministic Group Acknowledgment Transmissions in LoRa Networks for Industrial IoT Applications. *Sensors*, 21(4):1444, January 2021. doi: 10.3390/s21041444. URL <https://www.mdpi.com/1424-8220/21/4/1444>. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
- [17] Quy Lam Hoang and Hoon Oh. A Real-Time LoRa Protocol Using Logical Frame Partitioning for Periodic and Aperiodic Data Transmission. *IEEE Internet of Things Journal*, 9(16):15401–15412, August 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2022.3162019.
- [18] Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. Low Overhead Scheduling of LoRa Transmissions for Improved Scalability. *IEEE Internet of Things Journal*, 6(2):3097–3109, April 2019. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2018.2878942. URL <https://ieeexplore.ieee.org/document/8516298/>.
-

-
- [19] Dimitrios Zorbas, Christelle Caillouet, Khaled Abdelfadeel Hassan, and Dirk Pesch. Optimal Data Collection Time in LoRa Networks—A Time-Slotted Approach. *Sensors*, 21(4):1193, January 2021. ISSN 1424-8220. doi: 10.3390/s21041193. URL <https://www.mdpi.com/1424-8220/21/4/1193>. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
- [20] Celia Garrido-Hidalgo, Luis Roda-Sanchez, F. Javier Ramírez, Antonio Fernández-Caballero, and Teresa Olivares. Efficient online resource allocation in large-scale LoRaWAN networks: A multi-agent approach. *Computer Networks*, 221:109525, February 2023. ISSN 1389-1286. doi: 10.1016/j.comnet.2022.109525. URL <https://www.sciencedirect.com/science/article/pii/S138912862200559X>.
- [21] Natalia Chinchilla-Romero, Jorge Navarro-Ortiz, Pablo Muñoz, and Pablo Ameigeiras. Collision Avoidance Resource Allocation for LoRaWAN. *Sensors*, 21(4):1218, January 2021. doi: 10.3390/s21041218. URL <https://www.mdpi.com/1424-8220/21/4/1218>. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
- [22] Anna Triantafyllou, Panagiotis Sarigiannidis, Thomas Lagkas, Ioannis D. Moscholios, and Antonios Sarigiannidis. Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance. *Computer Networks*, 186:107735, February 2021. ISSN 1389-1286. doi: 10.1016/j.comnet.2020.107735. URL <https://www.sciencedirect.com/science/article/pii/S1389128620313232>.
- [23] Dimitrios Zorbas, Khaled Abdelfadeel, Panayiotis Kotzanikolaou, and Dirk Pesch. TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things. *Computer Communications*, 153:1–10, March 2020. ISSN 01403664. doi: 10.1016/j.comcom.2020.01.056. URL <https://linkinghub.elsevier.com/retrieve/pii/S0140366419314677>.
- [24] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris. Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations. In *2018 25th International Conference on Telecommunications (ICT)*, pages 516–520, June 2018. doi: 10.1109/ICT.2018.8464901.
- [25] Ruben M. Sandoval, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach. *IEEE Transactions on Network and Service Management*, 16(3):884–895, September 2019. ISSN 1932-4537,
-

-
- 2373-7379. doi: 10.1109/TNSM.2019.2927759. URL <https://ieeexplore.ieee.org/document/8758936/>.
- [26] Lluís Casals, Carles Gomez, and Rafael Vidal. The SF12 Well in LoRaWAN: Problem and End-Device-Based Solutions. *Sensors*, 21(19): 6478, January 2021. ISSN 1424-8220. doi: 10.3390/s21196478. URL <https://www.mdpi.com/1424-8220/21/19/6478>. Number: 19 Publisher: Multidisciplinary Digital Publishing Institute.
- [27] Yinghui Li, Jing Yang, and Jiliang Wang. DyLoRa: Towards Energy Efficient Dynamic LoRa Transmission Control. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 2312–2320, Toronto, ON, Canada, July 2020. IEEE. ISBN 978-1-72816-412-0. doi: 10.1109/INFOCOM41043.2020.9155407. URL <https://ieeexplore.ieee.org/document/9155407/>.
- [28] Fei Gu, Jianwei Niu, Landu Jiang, Xue Liu, and Mohammed Atiquzaman. Survey of the low power wide area network technologies. *Journal of Network and Computer Applications*, 149:102459, January 2020. ISSN 1084-8045. doi: 10.1016/j.jnca.2019.102459. URL <https://www.sciencedirect.com/science/article/pii/S1084804519303194>.
- [29] Ingenu. *How RPMA works the making of RPMA*. Ingenu Inc., 2016.
- [30] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, 2017. ISSN 1553-877X. doi: 10.1109/COMST.2017.2652320.
- [31] Riccardo Marini, Konstantin Mikhaylov, Gianni Pasolini, and Chiara Buratti. Low-Power Wide-Area Networks: Comparison of LoRaWAN and NB-IoT Performance. *IEEE Internet of Things Journal*, 9(21): 21051–21063, November 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2022.3176394.
- [32] Augustine Ikpehai, Bamidele Adebisi, Khaled M. Rabie, Kelvin Anoh, Ruth E. Ande, Mohammad Hammoudeh, Haris Gacanin, and Uche M. Mbanaso. Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review. *IEEE Internet of Things Journal*, 6(2):2225–2240, April 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2018.2883728.
- [33] Mohammed Jouhari, Nasir Saeed, Mohamed-Slim Alouini, and El Mehdi Amhoud. A Survey on Scalable LoRaWAN for Massive
-

-
- IoT: Recent Advances, Potentials, and Challenges. *IEEE Communications Surveys & Tutorials*, 25(3):1841–1876, 2023. ISSN 1553-877X, 2373-745X. doi: 10.1109/COMST.2023.3274934. URL <http://arxiv.org/abs/2202.11082>.
- [34] LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL, . URL <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>.
- [35] Semtech. AN1200.22 LoRa Modulation Basics, 2015. URL <https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>.
- [36] Junhyun Park, Kunho Park, Hyeongho Bae, and Chong-Kwon Kim. *EARN* : Enhanced ADR With Coding Rate Adaptation in LoRaWAN. *IEEE Internet of Things Journal*, 7(12):11873–11883, December 2020. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2020.3005881. URL <https://ieeexplore.ieee.org/document/9129785/>.
- [37] Semtech. LoRaWAN 1.1 Regional Parameters, 2017.
- [38] gateway conf, . URL <https://shorturl.at/erDT9>.
- [39] Blind ADR DEVELOPER PORTAL, . URL <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/blind-adr/>.
- [40] Asset Tracking and Workforce Safety in Saudi Arabia for the Red Sea Giga Project, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/actility-red-sea-solution-that-scales>.
- [41] The Urbana IoT Platform to manage the public lighting system, over LoRaWAN® network, for municipalities in Sicily, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/urbana-iot-solutions-that-scale>.
- [42] The Things Industries and Connexin to Deploy LoRaWAN® Smart City Projects Across Coventry and Warwickshire, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/sts-ttn-and-connexin>.
- [43] Cellnex Deploys LoRaWAN Network with Global Omnium, . URL <https://resources.lora-alliance.org/lorawan-solutions-tha>
-

-
- t-scale/cellnex-deploys-lorawan-network-solutions-that-scale.
- [44] MultiTech Solving Parking Problems with LoRaWAN, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/multitech-nobel-parking-pni-helium-lorawan-solutions-that-scale>.
- [45] CRA and TEKTELIC Build LoRaWAN® Network Coverage Across Czech Republic, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/cra-tektelic-lorawan-solutions-that-scale>.
- [46] Unlocking Innovation in Tampa, Florida with Expansive Senet Public LoRaWAN Network Coverage, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/senet-tampa-usecase-solutions-that-scale>.
- [47] Improve Productivity, Reduce Waste, Save Time: Real-Time Irrigation Control, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/solutions-that-scale-iot-ventures-irrigation-control-solution>.
- [48] Talkpool Sensors Utilize LoRaWAN® Connectivity through Netmore to Provide Building Damage Prevention Solutions, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/talkpool-and-netmore-lorawan-solutions-that-scale>.
- [49] Milesight Implements Smart Building Solution in Canada to Optimize Energy Efficiency and Indoor Air Quality, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/milesight-implements-smart-building-solution-in-canada-to-optimize-energy-efficiency-and-indoor-air-quality>.
- [50] 47,000 IAQ Sensors Create a Healthier Learning Environment in Canadian Schools, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/case-study-milesight-assek-solutions-that-scale>.
- [51] EverSmart Clean: A Cleaning Revolution from Microshare, . URL <https://resources.lora-alliance.org/lorawan-solutions-that-scale/microshare-solutions-that-scale>.

-
- [52] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study. *ACM Transactions on Sensor Networks*, 15(2):16:1–16:35, February 2019. ISSN 1550-4859. doi: 10.1145/3293534. URL <https://doi.org/10.1145/3293534>.
- [53] Ramon Sanchez-Iborra, Jesus Sanchez-Gomez, Juan Ballesta-Viñas, Maria-Dolores Cano, and Antonio F. Skarmeta. Performance Evaluation of LoRa Considering Scenario Conditions. *Sensors*, 18(3):772, March 2018. ISSN 1424-8220. doi: 10.3390/s18030772. URL <https://www.mdpi.com/1424-8220/18/3/772>. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [54] Panagiotis Gkotsiopoulos, Dimitrios Zorbas, and Christos Douligeris. Performance Determinants in LoRa Networks: A Literature Review. *IEEE Communications Surveys & Tutorials*, 23(3):1721–1758, 2021. ISSN 1553-877X. doi: 10.1109/COMST.2021.3090409.
- [55] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. Charm: Exploiting Geographical Diversity through Coherent Combining in Low-Power Wide-Area Networks. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 60–71, April 2018. doi: 10.1109/IPSN.2018.00013.
- [56] Forward Error Correction and Code Rate, . URL <https://www.thethingsnetwork.org/docs/lorawan/fec-and-code-rate/>.
- [57] Tommaso Polonelli, Davide Brunelli, Achille Marzocchi, and Luca Benini. Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment. *Sensors*, 19(4):838, February 2019. ISSN 1424-8220. doi: 10.3390/s19040838. URL <http://www.mdpi.com/1424-8220/19/4/838>.
- [58] Frank Loh, Noah Mehling, Stefan Geißler, and Tobias Hofffeld. Simulative Performance Study of Slotted Aloha for LoRaWAN Channel Access. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, April 2022. doi: 10.1109/NOMS54207.2022.9789898. ISSN: 2374-9709.
- [59] Laurent Chasserat, Nicola Accettura, and Pascal Berthou. Short: Achieving Energy Efficiency in Dense LoRaWANs through TDMA. In *2020 IEEE 21st International Symposium on "A World of Wireless,*

Mobile and Multimedia Networks” (WoWMoM), pages 26–29, August 2020. doi: 10.1109/WoWMoM49955.2020.00019.

- [60] Luca Beltramelli, Aamir Mahmood, Patrik Österberg, Mikael Gidlund, Paolo Ferrari, and Emiliano Sisinni. Energy Efficiency of Slotted LoRaWAN Communication With Out-of-Band Synchronization. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021. ISSN 1557-9662. doi: 10.1109/TIM.2021.3051238.
- [61] Luca Beltramelli, Aamir Mahmood, Patrik Osterberg, and Mikael Gidlund. LoRa Beyond ALOHA: An Investigation of Alternative Random Access Protocols. *IEEE Transactions on Industrial Informatics*, 17(5):3544–3554, May 2021. ISSN 1551-3203, 1941-0050. doi: 10.1109/TII.2020.2977046. URL <https://ieeexplore.ieee.org/document/9018210/>.
- [62] Anastasios Valkanis, Georgia A. Beletsioti, Konstantinos Kantelis, Petros Nicopolitidis, and Georgios Papadimitriou. A Reinforcement Learning assisted Backoff Algorithm for LoRa networks. In *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6, November 2021. doi: 10.1109/CITS52676.2021.9618456.
- [63] Athanasios Tsakmakis, Anastasios Valkanis, Georgia Beletsioti, Konstantinos Kantelis, Petros Nicopolitidis, and Georgios Papadimitriou. An Adaptive LoRaWAN MAC Protocol for Event Detection Applications. *Sensors*, 22(9):3538, January 2022. ISSN 1424-8220. doi: 10.3390/s22093538. URL <https://www.mdpi.com/1424-8220/22/9/3538>. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.
- [64] Gokcer Yapar, Tuna Tugcu, and Orhan Ermis. Time-Slotted ALOHA-based LoRaWAN Scheduling with Aggregated Acknowledgement Approach. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 383–390, November 2019. doi: 10.23919/FRUCT48121.2019.8981533. ISSN: 2305-7254.
- [65] Luca Leonardi, Filippo Battaglia, Gaetano Patti, and Lucia Lo Bello. Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 4141–4146, October 2018. doi: 10.1109/IECON.2018.8591568. ISSN: 2577-1647.

-
- [66] 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, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2977659.
- [67] Anna Triantafyllou, Dimitrios Zorbas, and Panagiotis Sarigiannidis. Time-slotted LoRa MAC with variable payload support. *Computer Communications*, 193:146–154, September 2022. ISSN 0140-3664. doi: 10.1016/j.comcom.2022.06.043. URL <https://www.sciencedirect.com/science/article/pii/S0140366422002444>.
- [68] Dimitrios Zorbas, Khaled Q. Abdelfadeel, Victor Cionca, Dirk Pesch, and Brendan O’Flynn. Offline Scheduling Algorithms for Time-Slotted LoRa-based Bulk Data Transmission. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 949–954, Limerick, Ireland, April 2019. IEEE. ISBN 978-1-5386-4980-0. doi: 10.1109/WF-IoT.2019.8767277. URL <https://ieeexplore.ieee.org/document/8767277/>.
- [69] 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, January 2020. ISSN 2327-4662. doi: 10.1109/JIOT.2019.2949918.
- [70] Zhuqing Xu, Junzhou Luo, Zhimeng Yin, Tian He, and Fang Dong. S-MAC: Achieving High Scalability via Adaptive Scheduling in LPWAN. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 506–515, July 2020. doi: 10.1109/INFOCOM41043.2020.9155474. ISSN: 2641-9874.
- [71] Rahim Haiahem, Pascale Minet, Selma Boumerdassi, and Leila Azouz Saidane. Collision-Free Transmissions in an IoT Monitoring Application Based on LoRaWAN. *Sensors*, 20(14):4053, January 2020. ISSN 1424-8220. doi: 10.3390/s20144053. URL <https://www.mdpi.com/1424-8220/20/14/4053>. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.
- [72] Jakub Pullmann and Dominik Macko. A New Planning-Based Collision-Prevention Mechanism in Long-Range IoT Networks. *IEEE Internet of Things Journal*, 6(6):9439–9446, December 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2019.2940994.

-
- [73] Luca Leonardi, Lucia Lo Bello, and Gaetano Patti. MRT-LoRa: A multi-hop real-time communication protocol for industrial IoT applications over LoRa networks. *Computer Communications*, 199:72–86, February 2023. ISSN 0140-3664. doi: 10.1016/j.comcom.2022.12.013. URL <https://www.sciencedirect.com/science/article/pii/S0140366422004637>.
- [74] Amalinda Gamage, Jansen Liando, Chaojie Gu, Rui Tan, Mo Li, and Olivier Seller. LMAC: Efficient Carrier-Sense Multiple Access for LoRa. *ACM Transactions on Sensor Networks*, 19(2):44:1–44:27, February 2023. ISSN 1550-4859. doi: 10.1145/3564530. URL <https://doi.org/10.1145/3564530>.
- [75] Thanh-Hai To and Andrzej Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018. doi: 10.1109/ICC.2018.8422800. ISSN: 1938-1883.
- [76] Nikos Kouvelas, Vijay Rao, and R. R. Venkatesha Prasad. Employing p-CSMA on a LoRa Network Simulator, May 2018. URL <http://arxiv.org/abs/1805.12263>. arXiv:1805.12263 [cs].
- [77] Edward M. Rochester, Asif M. Yousuf, Behnam Ousat, and Majid Ghaderi. Lightweight Carrier Sensing in LoRa: Implementation and Performance Evaluation. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, June 2020. doi: 10.1109/ICC40277.2020.9149103. ISSN: 1938-1883.
- [78] Chen Zhong and Andreas Springer. Analysis of a Novel Media Access Control Protocol for LoRa. *IEEE Internet of Things Journal*, 10(1): 341–356, January 2023. ISSN 2327-4662. doi: 10.1109/JIOT.2022.3200435.
- [79] Sergio Herrería-Alonso, Andrés Suárez-González, Miguel Rodríguez-Pérez, and Cándido López-García. Enhancing LoRaWAN scalability with Longest First Slotted CSMA. *Computer Networks*, 216:109252, October 2022. ISSN 1389-1286. doi: 10.1016/j.comnet.2022.109252. URL <https://www.sciencedirect.com/science/article/pii/S138912862200322X>.
- [80] Congduc Pham and Muhammad Ehsan. Dense Deployment of LoRa Networks: Expectations and Limits of Channel Activity Detection and Capture Effect for Radio Channel Access. *Sensors*, 21(3):825,
-

January 2021. ISSN 1424-8220. doi: 10.3390/s21030825. URL <https://www.mdpi.com/1424-8220/21/3/825>. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

- [81] Dimitrios Zorbas and Brendan O’Flynn. Autonomous Collision-Free Scheduling for LoRa-Based Industrial Internet of Things. In *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–5, Washington, DC, USA, June 2019. IEEE. ISBN 978-1-72810-270-2. doi: 10.1109/WoWMoM.2019.8792975. URL <https://ieeexplore.ieee.org/document/8792975/>.
- [82] Kawtar Lasri, Yann Ben Maissa, Loubna Echabbi, Oana Iova, and Fabrice Valois. Probabilistic and distributed traffic control in LPWANs. *Ad Hoc Networks*, 143:103121, April 2023. ISSN 1570-8705. doi: 10.1016/j.adhoc.2023.103121. URL <https://www.sciencedirect.com/science/article/pii/S1570870523000410>.
- [83] Christophe Moy and Lilian Besson. Decentralized Spectrum Learning for IoT Wireless Networks Collision Mitigation. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 644–651, May 2019. doi: 10.1109/DCOSS.2019.00117. ISSN: 2325-2944.
- [84] Amin Azari and Cicek Cavdar. Self-Organized Low-Power IoT Networks: A Distributed Learning Approach. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, December 2018. doi: 10.1109/GLOCOM.2018.8647894. ISSN: 2576-6813.
- [85] Ruben M. Sandoval, Antonio-Javier Garcia-Sanchez, Joan Garcia-Haro, and Thomas M. Chen. Optimal Policy Derivation for Transmission Duty-Cycle Constrained LPWAN. *IEEE Internet of Things Journal*, 5(4):3114–3125, August 2018. ISSN 2327-4662. doi: 10.1109/JIOT.2018.2833289.
- [86] Xiaolan Liu, Zhijin Qin, Yue Gao, and Julie A. McCann. Resource Allocation in Wireless Powered IoT Networks. *IEEE Internet of Things Journal*, 6(3):4935–4945, June 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2019.2895417.
- [87] Aoto Kaburaki, Koichi Adachi, Osamu Takyu, Mai Ohta, and Takeo Fujii. Autonomous Decentralized Traffic Control Using Q-Learning in

-
- LPWAN. *IEEE Access*, 9:93651–93661, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3093421.
- [88] Wei Wu, Yan Li, Yanhe Zhang, Bin Wang, and Wennai Wang. Distributed Queueing-Based Random Access Protocol for LoRa Networks. *IEEE Internet of Things Journal*, 7(1):763–772, January 2020. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2019.2945327. URL <https://ieeexplore.ieee.org/document/8856256/>.
- [89] Yohei Hasegawa and Kazuya Suzuki. A Multi-User ACK-Aggregation Method for Large-Scale Reliable LoRaWAN Service. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019. doi: 10.1109/ICC.2019.8761253. ISSN: 1938-1883.
- [90] Junhee Lee, Wun-Cheol Jeong, and Byeong-Cheol Choi. A Scheduling Algorithm for Improving Scalability of LoRaWAN. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1383–1388, October 2018. doi: 10.1109/ICTC.2018.8539392. ISSN: 2162-1233.
- [91] Rajeev Piyare, Amy L. Murphy, Michele Magno, and Luca Benini. On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT. *Sensors (Basel, Switzerland)*, 18(11), November 2018. ISSN 1424-8220. doi: 10.3390/s18113718. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6263638/>.
- [92] Nour El Hoda Djidi, Matthieu Gautier, Antoine Courtay, Olivier Berder, and Michele Magno. How Can Wake-up Radio Reduce LoRa Downlink Latency for Energy Harvesting Sensor Nodes? *Sensors*, 21(3):733, January 2021. ISSN 1424-8220. doi: 10.3390/s21030733. URL <https://www.mdpi.com/1424-8220/21/3/733>. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [93] Van-Truong Truong, Anand Nayyar, and Showkat Lone. System Performance of Wireless Sensor Network Using LoRa–Zigbee Hybrid Communication. *Computers, Materials & Continua*, 68(2):1615–1635, 2021. ISSN 1546-2218, 1546-2218. doi: 10.32604/cmc.2021.016922. URL <https://www.techscience.com/cmc/v68n2/42201>. Number: 2 Publisher: Tech Science Press.
- [94] Luca Davoli, Emanuele Pagliari, and Gianluigi Ferrari. Hybrid LoRa-IEEE 802.11s Opportunistic Mesh Networking for Flexible UAV Swarming. *Drones*, 5(2):26, June 2021. ISSN 2504-446X. doi: 10.339
-

-
- 0/drones5020026. URL <https://www.mdpi.com/2504-446X/5/2/26>. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.
- [95] Luca Beltramelli, Aamir Mahmood, Paolo Ferrari, Patrik Österberg, Mikael Gidlund, and Emiliano Sisinni. Synchronous LoRa Communication by Exploiting Large-Area Out-of-Band Synchronization. *IEEE Internet of Things Journal*, 8(10):7912–7924, May 2021. ISSN 2327-4662. doi: 10.1109/JIOT.2020.3041818.
- [96] Xihai Zhang, Mingming Zhang, Fanfeng Meng, Yue Qiao, Suijia Xu, and Senghout Hour. A Low-Power Wide-Area Network Information Monitoring System by Combining NB-IoT and LoRa. *IEEE Internet of Things Journal*, 6(1):590–598, February 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2018.2847702.
- [97] Rajeev Piyare, Amy L. Murphy, Csaba Kiraly, Pietro Tosato, and Davide Brunelli. Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey. *IEEE Communications Surveys & Tutorials*, 19(4):2117–2157, 2017. ISSN 1553-877X. doi: 10.1109/COMST.2017.2728092.
- [98] Antonios Prapas, Konstantinos F. Kantelis, Petros Nicopolitidis, and Georgios I. Papadimitriou. An On-Demand TDMA Approach Optimized for Low-Latency IoT Applications. *Sensors*, 22(17):6461, January 2022. ISSN 1424-8220. doi: 10.3390/s22176461. URL <https://www.mdpi.com/1424-8220/22/17/6461>. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute.
- [99] Georgia A. Beletsoti, Konstantinos F. Kantelis, Anastasios Valkanis, Petros Nicopolitidis, and Georgios I. Papadimitriou. A Multi-level TDMA approach for IoT applications with WuR support. *IEEE Internet of Things Journal*, pages 1–1, 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2022.3185750.
- [100] Fayçal Ait Aoudia, Matthieu Gautier, Michele Magno, Mickaël Le Gentil, Olivier Berder, and Luca Benini. Long-short range communication network leveraging LoRa™ and wake-up receiver. *Microprocessors and Microsystems*, 56:184–192, February 2018. ISSN 01419331. doi: 10.1016/j.micpro.2017.12.004. URL <https://linkinghub.elsevier.com/retrieve/pii/S0141933117300455>.
- [101] Xiaofan Jiang, Heng Zhang, Edgardo Alberto Barsallo Yi, Nithin Raghunathan, Charilaos Mousoulis, Somali Chaterji, Dimitrios Per-

-
- oulis, Ali Shakouri, and Saurabh Bagchi. Hybrid Low-Power Wide-Area Mesh Network for IoT Applications. *IEEE Internet of Things Journal*, 8(2):901–915, January 2021. ISSN 2327-4662. doi: 10.1109/JIOT.2020.3009228.
- [102] Mohammad Afhamisis and Maria Rita Palattella. SALSA: A Scheduling Algorithm for LoRa to LEO Satellites. *IEEE Access*, 10:11608–11615, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3146021.
- [103] Muhammad Asad Ullah, Junnaid Iqbal, Arliones Hoeller, Richard Demo Souza, and Hirley Alves. K-Means Spreading Factor Allocation for Large-Scale LoRa Networks. *Sensors*, 19(21):4723, January 2019. doi: 10.3390/s19214723. URL <https://www.mdpi.com/1424-8220/19/21/4723>. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.
- [104] Jin-Taek Lim and Youngnam Han. Spreading Factor Allocation for Massive Connectivity in LoRa Systems. *IEEE Communications Letters*, 22(4):800–803, April 2018. ISSN 1089-7798. doi: 10.1109/LCOMM.2018.2797274. URL <https://ieeexplore.ieee.org/document/8268120/>.
- [105] Deepak Saluja, Rohit Singh, Sukriti Gautam, and Suman Kumar. EWS: Exponential Windowing Scheme to Improve LoRa Scalability. *IEEE Transactions on Industrial Informatics*, 18(1):252–265, January 2022. ISSN 1941-0050. doi: 10.1109/TII.2021.3074377.
- [106] Riccardo Marini, Walter Ceroni, and Chiara Buratti. A Novel Collision-Aware Adaptive Data Rate Algorithm for LoRaWAN Networks. *IEEE Internet of Things Journal*, 8(4):2670–2680, February 2021. ISSN 2327-4662. doi: 10.1109/JIOT.2020.3020189.
- [107] Hakkı Soy. An adaptive spreading factor allocation scheme for mobile LoRa networks: Blind ADR with distributed TDMA scheduling. *Simulation Modelling Practice and Theory*, 125:102755, May 2023. ISSN 1569-190X. doi: 10.1016/j.simpat.2023.102755. URL <https://www.sciencedirect.com/science/article/pii/S1569190X23000321>.
- [108] Khaled Q. Abdelfadeel, Victor Cionca, and Dirk Pesch. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. In *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 14–15, Chania, Greece, June 2018. IEEE. ISBN 978-1-5386-4725-7. doi:
-

-
- 10.1109/WoWMoM.2018.8449737. URL <https://ieeexplore.ieee.org/document/8449737/>.
- [109] Brecht Reynders, Wannes Meert, and Sofie Pollin. Power and spreading factor control in low power wide area networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, Paris, France, May 2017. IEEE. ISBN 978-1-4673-8999-0. doi: 10.1109/ICC.2017.7996380. URL <http://ieeexplore.ieee.org/document/7996380/>.
- [110] Wha Sook Jeon and Dong Geun Jeong. Adaptive Uplink Rate Control for Confirmed Class A Transmission in LoRa Networks. *IEEE Internet of Things Journal*, 7(10):10361–10374, October 2020. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2020.2986494. URL <https://ieeexplore.ieee.org/document/9060905/>.
- [111] Gabriel Germino Martins de Jesus, Richard Demo Souza, Carlos Montez, and Arliones Hoeller. LoRaWAN Adaptive Data Rate With Flexible Link Margin. *IEEE Internet of Things Journal*, 8(7):6053–6061, April 2021. ISSN 2327-4662. doi: 10.1109/JIOT.2020.3033797.
- [112] Ruben M. Sandoval, David Rodenas-Herraiz, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Deriving and Updating Optimal Transmission Configurations for Lora Networks. *IEEE Access*, 8:38586–38595, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2973252.
- [113] Aghiles Djoudi, Rafik Zitouni, Nawel Zangar, and Laurent George. LoRa network reconfiguration with Markov Decision Process and Fuzzy C-Means clustering. *Computer Communications*, 196:129–140, December 2022. ISSN 0140-3664. doi: 10.1016/j.comcom.2022.08.020. URL <https://www.sciencedirect.com/science/article/pii/S0140366422003358>.
- [114] Inaam Ilahi, Muhammad Usama, Muhammad Omer Farooq, Muhammad Umar Janjua, and Junaid Qadir. LoRaDRL: Deep Reinforcement Learning Based Adaptive PHY Layer Transmission Parameters Selection for LoRaWAN. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pages 457–460, Sydney, NSW, Australia, November 2020. IEEE. ISBN 978-1-72817-158-6. doi: 10.1109/LCN48667.2020.9314772. URL <https://ieeexplore.ieee.org/document/9314772/>.
- [115] Alexander Valach and Dominik Macko. Upper Confidence Bound Based Communication Parameters Selection to Improve Scalability of
-

-
- LoRa@FIIT Communication. *IEEE Sensors Journal*, 22(12):12415–12427, June 2022. ISSN 1558-1748. doi: 10.1109/JSEN.2022.3174663.
- [116] Ondrej Perešíni and Tibor Krajčovič. More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols. In *2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–6, September 2017. doi: 10.1109/ICAICT.2017.8686837. ISSN: 2472-8586.
- [117] Brecht Reynders, Qing Wang, Pere Tuset-Peiro, Xavier Vilajosana, and Sofie Pollin. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, June 2018. ISSN 2327-4662. doi: 10.1109/JIOT.2018.2815150. URL <https://ieeexplore.ieee.org/document/8315103/>.
- [118] Hanan Alahmadi, Fatma Bouabdallah, Baraq Ghaleb, and Ahmed Al-Dubai. Sensitivity-Aware Configurations for High Packet Generation Rate LoRa Networks. In *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, pages 240–246, December 2021. doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00048.
- [119] OMNeT++ Discrete Event Simulator, . URL <https://omnetpp.org/>.
- [120] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. Adaptive configuration of lora networks for dense IoT deployments. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, Taipei, Taiwan, April 2018. IEEE. ISBN 978-1-5386-3416-5. doi: 10.1109/NOMS.2018.8406255. URL <https://ieeexplore.ieee.org/document/8406255/>.
- [121] Theodore S. Rappaport. *Wireless Communications: Principles and Practice, 2e*. Prentice Hall PTR, 1996. ISBN 978-93-325-7616-2. URL <https://books.google.com.sa/books?id=gPKODwAAQBAJ>.
- [122] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. Do LoRa Low-Power Wide-Area Networks Scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16*, pages 59–67, Malta, Malta, 2016. ACM Press. ISBN 978-1-4503-4502-6. doi: 10.1145/2988287.2989163. URL <http://dl.acm.org/citation.cfm?doid=2988287.2989163>.

-
- [123] Francesca Cuomo, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, and Patrizio Pisani. EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, Rome, October 2017. IEEE. ISBN 978-1-5386-3839-2. doi: 10.1109/WiMOB.2017.8115779. URL <http://ieeexplore.ieee.org/document/8115779/>.
- [124] Juha Petajajarvi, Konstantin Mikhaylov, Antti Roivainen, Tuomo Hanninen, and Marko Pettissalo. On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In *2015 14th International Conference on ITS Telecommunications (ITST)*, pages 55–59, Copenhagen, Denmark, December 2015. IEEE. ISBN 978-1-4673-9382-9. doi: 10.1109/ITST.2015.7377400. URL <http://ieeexplore.ieee.org/document/7377400/>.
- [125] European Telecommunications Standards Institute. ETSI EN 300 220-1 V2.4.1, 2012. URL <https://shorturl.at/crvQZ>.
- [126] T. T. Network. Duty Cycle. URL <https://shorturl.at/CEGYZ>.
- [127] The things Network. Frequency Plans. URL <https://shorturl.at/efkoU>.
- [128] 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*, 1(1):150597, October 2015. ISSN 2414-1399. doi: 10.4108/eai.26-10-2015.150597. URL <http://eudl.eu/doi/10.4108/eai.26-10-2015.150597>.
- [129] Gopika Premankar, Bissan Ghaddar, Mariusz Slabicki, and Mario Di Francesco. Optimal Configuration of LoRa Networks in Smart Cities. *IEEE Transactions on Industrial Informatics*, 16(12):7243–7254, December 2020. ISSN 1941-0050. doi: 10.1109/TII.2020.2967123.
- [130] Laksh Bhatia, Ivana Tomić, and Julie A. McCann. LPWA-MAC: a Low Power Wide Area network MAC protocol for cyber-physical systems. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems - SenSys '18*, pages 361–362, Shenzhen, China, 2018. ACM Press. ISBN 978-1-4503-5952-8. doi: 10.1145/3274783.3275183. URL <http://dl.acm.org/citation.cfm?doid=3274783.3275183>.
- [131] Air time calculator | LoRaTools, . URL <https://shorturl.at/aowHJ>.

-
- [132] Dimitrios Zorbas. Design Considerations for Time-Slotted LoRa(WAN). In *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks on Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*, EWSN '20, pages 271–276, USA, February 2020. Junction Publishing. ISBN 978-0-9949886-4-5.
- [133] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002. ISSN 1389-1286. doi: 10.1016/S1389-1286(01)00302-4. URL <http://www.sciencedirect.com/science/article/pii/S1389128601003024>.
- [134] Yulei Wu, Hong-Ning Dai, Haozhe Wang, Zehui Xiong, and Song Guo. A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory. *IEEE Communications Surveys & Tutorials*, 24(2): 1175–1211, 2022. ISSN 1553-877X. doi: 10.1109/COMST.2022.3158270.
- [135] Latif U. Khan, Ibrar Yaqoob, Nguyen H. Tran, Zhu Han, and Choong Seon Hong. Network Slicing: Recent Advances, Taxonomy, Requirements, and Open Research Challenges. *IEEE Access*, 8:36009–36028, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2975072.