

## **Towards Ubiquitous and Efficient LoRaWAN**

### **MAC-Layer Protocols and APP-Layer Coding Mechanisms for Scalable and Energy-Efficient Long-Range Wide-Area Networks (LoRaWAN)**

Kouvelas, N.

#### **DOI**

[10.4233/uuid:8021b680-5186-4beb-becc-f8799763ee21](https://doi.org/10.4233/uuid:8021b680-5186-4beb-becc-f8799763ee21)

#### **Publication date**

2022

#### **Document Version**

Final published version

#### **Citation (APA)**

Kouvelas, N. (2022). *Towards Ubiquitous and Efficient LoRaWAN: MAC-Layer Protocols and APP-Layer Coding Mechanisms for Scalable and Energy-Efficient Long-Range Wide-Area Networks (LoRaWAN)*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:8021b680-5186-4beb-becc-f8799763ee21>

#### **Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

#### **Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

#### **Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# **TOWARDS UBIQUITOUS AND EFFICIENT LoRAWAN**

MAC-LAYER PROTOCOLS AND APP-LAYER CODING  
MECHANISMS FOR SCALABLE AND ENERGY-EFFICIENT  
LONG-RANGE WIDE-AREA NETWORKS (LoRAWAN)



# **TOWARDS UBIQUITOUS AND EFFICIENT LoRAWAN**

MAC-LAYER PROTOCOLS AND APP-LAYER CODING  
MECHANISMS FOR SCALABLE AND ENERGY-EFFICIENT  
LONG-RANGE WIDE-AREA NETWORKS (LoRAWAN)

## **Dissertation**

for the purpose of obtaining the degree of doctor  
at Delft University of Technology,  
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,  
chair of the Board for Doctorates,  
to be defended publicly on Friday 6 May 2022, at 12:30 hours

by

**Nikolaos KOUVELAS**

Master of Science in Electrical Engineering,  
Delft University of Technology, the Netherlands  
born in Tripoli, Greece.

This dissertation has been approved by the

promotor: prof. dr. K.G. Langendoen  
promotor: dr. R. R. Venkatesha Prasad

Composition of the doctoral committee:

Rector Magnificus	Chairman
Prof. dr. K. G. Langendoen	Delft University of Technology, The Netherlands
Dr. R. R. Venkatesha Prasad	Delft University of Technology, The Netherlands

*Independent Members:*

Dr. C. A. Boano	Graz University of Technology, Austria
Prof. dr. ir. A. Bozzon	Delft University of Technology, The Netherlands
Prof. dr. P. Chatzimisios	International Hellenic University, Greece
Prof. dr. ir. I. G. M. M. Niemegeers	Eindhoven University of Technology, The Netherlands
Prof. dr. D. H. J. Pesch	University College Cork, Ireland

*Reserve Member:*

Prof. dr. ir. F. A. Kuipers	Delft University of Technology, The Netherlands
-----------------------------	---



This dissertation describes work partially undertaken in the context of the SCOTT project (<http://www.scott-project.eu>) funded from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This joint undertaking receives support from the European Union's Horizon 2020 research and innovation program and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, Norway.

*Keywords:* LoRaWAN, Channel Sensing, Scalability, Energy Efficiency, Coding, MAC Layer, Application Layer

*Printed by:* Ipskamp Printing, Enschede

Copyright © 2022 by N. Kouvelas. All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, without the permission of the author.

ISBN 978-94-6421-737-7

An electronic version of this dissertation is available at  
<http://repository.tudelft.nl/>.

*To my mother Fani.*



# CONTENTS

<b>Summary</b>	<b>xi</b>
<b>Samenvatting</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The Scalability Requirements of IoT Applications . . . . .	3
1.2 Ubiquitous Connectivity Using LoRaWAN . . . . .	4
1.3 Contributions and Outline . . . . .	9
<b>2 Long Range Wide Area Networks (LoRaWAN): Physical and Medium Access Control layer</b>	<b>13</b>
2.1 Brief introduction to LoRaWAN . . . . .	13
2.2 MAC-layer protocols for LoRaWAN . . . . .	16
2.3 Characterization of Capture Effect and Channel Activity Detection . . . . .	21
2.3.1 Capture Effect . . . . .	21
2.3.1.1 Power Difference . . . . .	22
2.3.1.2 Delay Offset . . . . .	24
2.3.2 Channel Activity Detection. . . . .	26
2.3.2.1 Characterization of CAD . . . . .	27
2.4 Summary and Outlook . . . . .	31
<b>3 A persistent distributed MAC protocol for LoRaWAN</b>	<b>33</b>
3.1 $p$ -CARMA . . . . .	35
3.1.1 Working of $p$ -CARMA . . . . .	36
3.1.2 Adaptive $p$ -value Algorithm . . . . .	38
3.1.2.1 Heuristic Approach . . . . .	40
3.1.2.2 Collision Delay Ratio (CDR) . . . . .	40
3.1.2.3 Parameters $\bar{D}$ , $D_{min}$ , and $D_{max}$ . . . . .	42
3.1.2.4 Network Bootstrapping . . . . .	42
3.2 Simulation Parameters and Metrics . . . . .	43
3.2.0.1 Simulation parameters . . . . .	43
3.2.0.2 Metrics . . . . .	44
3.3 Performance Evaluation . . . . .	45
3.4 Practical Evaluation. . . . .	50
3.5 Conclusions. . . . .	52
<b>4 A non-persistent distributed MAC protocol for LoRaWAN</b>	<b>55</b>
4.1 $np$ -CECADA . . . . .	56
4.1.1 Network Bootstrapping . . . . .	57
4.1.2 Initial backoff mechanism . . . . .	58

4.1.3	Adaptive transmission power algorithm . . . . .	58
4.1.4	Adaptive Backoff Algorithm . . . . .	58
4.1.4.1	Heuristic Approach . . . . .	60
4.1.4.2	Persistence probability $p$ via local information . . . . .	60
4.1.4.3	Regional information – periodic device beaconing $R$ . . . . .	61
4.1.4.4	Global information – ratio of collided frames $Q$ . . . . .	62
4.2	Simulation Specifics . . . . .	62
4.3	Simulation-based Evaluation . . . . .	64
4.3.1	Single Gateway . . . . .	65
4.3.2	Multiple Gateways . . . . .	69
4.4	Practical Evaluation . . . . .	74
4.5	Conclusion . . . . .	75
<b>5</b>	<b>Virtual TDMA for LoRaWAN: scheduling transmissions through dedicated channels</b>	<b>77</b>
5.1	Design of MAC protocol . . . . .	80
5.1.1	Design of control channel . . . . .	81
5.1.1.1	Optimization problem . . . . .	81
5.1.1.2	Throughput increase based on SF . . . . .	82
5.1.1.3	Determination of optimal length and Spreading Factor of control packets . . . . .	83
5.1.2	SFMAC model . . . . .	84
5.1.3	Optimization of SFMAC parameters . . . . .	87
5.1.3.1	Duration of transmission control packet . . . . .	88
5.1.3.2	Contention window size, update policy, and number of attempts . . . . .	88
5.2	Parameters of simulation and evaluation . . . . .	89
5.2.1	Evaluation metrics . . . . .	90
5.3	Simulation-based Evaluation . . . . .	91
5.3.0.1	Effect of imposing failures in CAD on the performance of SFMAC . . . . .	94
5.3.1	Comparison with state-of-the-art . . . . .	94
5.3.1.1	Comparison with LMAC . . . . .	95
5.3.1.2	Comparison with $p$ -CARMA . . . . .	95
5.3.1.3	Comparison with $np$ -CECADA . . . . .	96
5.3.2	Multiple gateway scenario . . . . .	96
5.3.3	SFMAC evaluation in a practical application . . . . .	98
5.4	Practical Evaluation . . . . .	100
5.5	Conclusions . . . . .	101
<b>6</b>	<b>Application layer coding techniques for the recovery of data from lost and corrupted LoRa frames</b>	<b>103</b>
6.1	Related Work . . . . .	104
6.2	Memory-based coding for data recovery from lost LoRa frames . . . . .	106
6.2.1	Setup and Data Collection . . . . .	107
6.2.1.1	Measurement Setup . . . . .	107

6.2.1.2	Data Collection Scenarios	108
6.2.2	Frame Loss Characterisation	108
6.2.2.1	Frame Loss over Distance	108
6.2.2.2	Burstiness	110
6.2.3	DaRe: Data Recovery in LoRaWAN	110
6.2.3.1	Benchmark: Conventional Repetition Coding and LT Codes	112
6.2.4	Numerical Results	112
6.2.5	Practical Evaluation	113
6.2.5.1	Implementation	113
6.2.5.2	Evaluation Results	114
6.2.5.3	Measurement Results	115
6.3	Memoryless coding for efficient and real-time data recovery from corrupted LoRa frames	118
6.3.1	Characterization of Error Pattern	118
6.3.2	DC: Divide & Code	121
6.3.2.1	Encoder	121
6.3.2.2	Decoder	122
6.3.2.3	Proposed Schemes for Real-time Decoder	123
6.3.3	Performance Evaluation	123
6.4	Conclusions	127
<b>7</b>	<b>Conclusion</b>	<b>129</b>
7.1	Recapitulation	130
	<b>Bibliography</b>	<b>133</b>
	<b>Propositions</b>	<b>145</b>
	<b>Acknowledgements</b>	<b>147</b>
	<b>List of Publications</b>	<b>149</b>



# SUMMARY

The Internet of Things (IoT) disrupted technology and society by introducing the concept of "smart things", or IoT-devices. These devices can interact with their environment through sensing and actuation, and then communicate the results of these actions through the Internet. Their ability to monitor the ambient environment and provide the data required for proper decisions on the operation of a system introduced the usage of IoT-devices in many diverse domains, like healthcare, automotive, and intelligent buildings. Today, there are billions of IoT-devices involved in several aspects of our everyday life, from gauges monitoring the levels of fuel in our cars, to CO<sub>2</sub> sensors monitoring the air quality in our cities. However, this level of pervasiveness introduced a number of challenges, obstructing the seamless integration of IoT in cities and industries. These challenges –namely related to energy constraints, low computational capabilities, and harsh conditions of deployment– led to the requirement of protocols that guarantee long-range transmission of information in an energy-efficient manner.

To meet these needs, Low Power and Wide Area Networks (LPWAN) were introduced to interconnect the masses of IoT-devices using robust signal encoding and simple protocols for channel access. With data rates of up to 300 kbps and a multi-kilometer range of transmissions, LPWANs are the cornerstone of low-power IoT-devices in terms of communication. Among the popular LPWAN technologies, Long Range (LoRa)WAN has been the most accomplished since it includes a number of appealing characteristics for IoT applications: (i) it is easily deployable by both corporate parties and individuals, involving zero costs for spectrum usage as it operates in the ISM bands, (ii) it establishes transmissions of several kilometers that are robust against multi-path fading and noise, and (iii) it communicates using a star topology wherein devices transmit directly their generated information-frames to gateways, which forward the information to a network server. LoRaWAN offers a series of different transmission parameters, whose configuration can lead to data rates of up to 50 kbps.

However, due to operating in the ISM band LoRa-signals are interfered by the signals of other ISM technologies. Also, LoRa-devices do not assess the idleness of the channel before transmitting, leading to collisions of frames that are transmitted simultaneously. On top of that, the energy constraints of LoRa-devices and the asynchronous communication that is mostly directed from devices to gateways do not allow more sophisticated ways to regulate transmissions. These limit the ability of LoRaWAN to interconnect IoT-devices in large-scale network deployments that involve more than a few hundred of devices connected per single gateway. Under these circumstances, LoRaWAN cannot enable the connectivity of the masses of IoT-devices that will be operating on the applications of the future Smart Cities and Industry 4.0.

In this thesis, we aim to change that by improving the scalability of LoRaWAN in an energy-efficient manner and without deviating from the LoRaWAN standard. Our approach is holistic, involving research on the three main layers of LoRa networks, the

physical (PHY), the Medium Access Control (MAC), and the application (APP) layer. Based on this research we design algorithms for MAC and APP layers.

To design effective and efficient protocols for the MAC layer of LoRaWAN, we need to first delve into the specifics of the LoRa-PHY, namely the Capture Effect (CE) phenomenon and the Channel Activity Detection (CAD) mechanism. CE is manifested among frames received slightly delayed compared to each other and/or with different signal-strengths. By conducting exhaustive in-field experiments we infer probabilistic rules modeling the reception of a LoRa-frame under interference. CAD mechanism can be used to assess the existence of ongoing LoRa-signals on the channel. By performing a long series of real-world experiments under different fading conditions and device setups we find the success rate of CAD as transmission range increases.

After having characterized CE and CAD, the models we created are incorporated into three novel, distributed protocols we designed for the MAC layer of LoRaWAN: *p*-CARMA, *np*-CECADA, and SFMAC. In these protocols, our model of CE is used by devices to decrease their transmission power in order to reduce network interference and CAD is used as an economical means of channel sensing to assess channel idleness. *p*-CARMA and *np*-CECADA adopt principles of Carrier-Sense Multiple-Access (CSMA) protocols in accessing the channel while estimating the traffic in their vicinity through heuristics. SFMAC dedicates low data rate transmissions to devices transmitting control-frames to inform their neighbors every time they transmitted a data-frame. High data rate transmissions are utilized to transmit data-frames. *p*-CARMA, *np*-CECADA, and SFMAC not only outperform vanilla LoRaWAN but also enhance the state-of-the-art in terms of channel utilization and ratio of received over transmitted frames. At the same time, the energy consumed per frame is minimal, while the portion of energy wasted for collided frames is diminished by multi-folds compared to vanilla LoRaWAN. The above observations are further confirmed by field experiments we perform using our testbed of 30 LoRa-devices mimicking scenarios of high levels of traffic.

On the APP layer of LoRaWAN, we design novel coding mechanisms for data recovery from lost and corrupted frames. At first, we conduct a series of in-field experiments to closely study the fading conditions that lead to frame losses and symbol corruptions in LoRa-frames. We utilize our findings to create two coding mechanisms for the LoRaWAN-APP: Data Recovery (DaRe) and Divide & Code (DC). DaRe extends each currently transmitted frame with encoded information from a window of previously transmitted frames. Thus, the information of a lost frame can still be recovered by decoding part of the received frames after it. DC pre-encodes LoRa-frames using lightweight coding schemes. On decoding, DC prioritizes the symbols with the highest probability of being corrupted. Therefore, it decreases the time needed for decoding while achieving high numbers of decoded frames. Both DaRe and DC outperform vanilla LoRaWAN in terms of data recovery and save energy from retransmissions, while they add minimum encoding overhead. Furthermore, DaRe outperforms Luby Transform codes, especially in scenarios of high frame-loss probability, and DC outperforms Reed-Solomon coding.

To sum up, in this dissertation, we asked how to improve the scalability in LoRaWAN while being energy-efficient. We answered by designing novel MAC layer protocols and APP layer mechanisms that are backward compatible with the LoRaWAN standard.

# SAMENVATTING

Het Internet of Things (IoT) heeft de technologie en de samenleving ontwricht door het concept van 'smart things' te introduceren, ook wel IoT-apparaten genoemd. Deze apparaten kunnen reageren op hun omgeving door middel van sensoren en actuatoren, en vervolgens de resultaten van deze acties communiceren via het internet. Het vermogen van IoT-apparaten om de omringende omgeving te monitoren en de gemeten data te leveren die nodig is voor juiste beslissingen over de werking van een systeem, introduceerde het gebruik van IoT-apparaten in veel verschillende domeinen, zoals de gezondheidszorg, auto's en intelligente gebouwen. Tegenwoordig zijn er miljarden IoT-apparaten betrokken bij verschillende aspecten van ons dagelijks leven, van meters voor het brandstof niveau in onze auto's, tot CO<sub>2</sub>-sensoren die de luchtkwaliteit in onze steden monitoren. De brede inzet van IoT-apparaten heeft een aantal uitdagingen met zich mee gebracht, waardoor de naadloze integratie van IoT in steden en industrieën wordt belemmerd. Deze uitdagingen - met name gerelateerd aan energie beperkingen, beperkte reken capaciteit en lastige installatie locaties - hebben geleid tot de behoefte aan protocollen die de overdracht van informatie over een lange afstand op een energie-efficiënte manier garanderen.

Om aan deze behoeften te voldoen, werden Low Power en Wide Area Networks (LPWAN) geïntroduceerd om de massa's IoT-apparaten met elkaar te verbinden met behulp van robuuste signaal coderingstechnieken en eenvoudige protocollen voor kanaaltoegang. Met datasnelheden tot 300 kbps en een transmissiebereik van meerdere kilometers zijn LPWAN's de hoeksteen van low-power IoT-apparaten op het gebied van communicatie. Van de populaire LPWAN-technologieën is Long Range (LoRa)WAN het meest succesvol omdat het een aantal aantrekkelijke kenmerken voor IoT-toepassingen bevat: (i) het is gemakkelijk inzetbaar door zowel zakelijke partijen als individuen, zonder extra kosten voor spectrumgebruik aangezien het in de ISM-banden opereert, (ii) het brengt transmissies van meerdere kilometers tot stand die robuust zijn tegen multipath fading en ruis, en (iii) het communiceert met behulp van een ster topologie waarin apparaten hun gegenereerde informatie frames rechtstreeks naar gateways verzenden, die de informatie vervolgens doorsturen naar een netwerkserver. LoRaWAN biedt een reeks verschillende transmissieparameters, waarvan de configuratie kan leiden tot datasnelheden tot 50 kbps.

LoRa-signalen kunnen worden gestoord door de signalen van andere technologieën die in dezelfde ISM-band werkzaam zijn. Ook luisteren LoRa-apparaten niet of het kanaal al gebruikt wordt voordat een nieuwe transmissie gestart wordt, wat leidt tot botsingen van frames die tegelijkertijd worden verzonden. Bovendien laten de energiebeperkingen van LoRa-apparaten en de asynchrone communicatie naar de gateways geen geavanceerde manieren toe om transmissies te reguleren. Deze aspecten beperken de mogelijkheden van LoRaWAN om IoT-apparaten met elkaar te verbinden in grootschalige netwerk implementaties. Onder deze omstandigheden kan LoRaWAN de connectiviteit niet realiseren voor de massa's IoT-apparaten die zullen werken binnen de toekomstige Smart Cities en Industry 4.0 toepassingen.

In dit proefschrift willen we daar verandering in brengen door de schaalbaarheid van LoRaWAN te verbeteren op een energiezuinige manier en zonder af te wijken van de LoRaWAN-standaard. Onze benadering is holistisch en omvat onderzoek naar de drie hoofdlagen van LoRa-netwerken, de fysieke (PHY), de Medium Access Control (MAC) en de applicatielaag (APP). Op basis van dit onderzoek ontwerpen we algoritmen voor de MAC- en APP-lagen.

Om effectieve en efficiënte protocollen voor de MAC-laag van LoRaWAN te ontwerpen, moeten we ons eerst verdiepen in de specifieke kenmerken van de LoRa-PHY, namelijk het Capture Effect (CE) fenomeen en het Channel Activity Detection (CAD) mechanisme. CE komt tot uiting bij frames die in vergelijking met elkaar enigszins vertraagd worden ontvangen en/of met variërende signaalsterkte. Door middel van uitgebreide veldexperimenten, leiden we probabilistische wetten af die de ontvangst van een LoRa-frame onder interferentie modelleren. Het CAD-mechanisme kan worden gebruikt om het bestaan van LoRa-signalen op het kanaal te beoordelen. Door een lange reeks praktijkexperimenten uit te voeren onder verschillende omstandigheden en apparaatinstellingen, vinden we het succespercentage van CAD in relatie tot het transmissiebereik.

Nadat we CE en CAD hebben gekarakteriseerd, zijn de gemaakte modellen opgenomen in drie nieuwe, gedistribueerde protocollen die we hebben ontworpen voor de MAC-laag van LoRaWAN:  $p$ -CARMA,  $np$ -CECADA en SFMAC. In deze protocollen wordt ons CE-model door apparaten gebruikt om hun transmissie vermogen te verlagen voor het verminderen van de netwerkinterferentie. Daarnaast wordt CAD gebruikt als een economisch middel om kanaalinactiviteit te identificeren.  $p$ -CARMA en  $np$ -CECADA gebruiken de principes van Carrier-Sense Multiple-Access (CSMA)-protocollen bij het verkrijgen van toegang tot het kanaal, terwijl ze het verkeer in hun omgeving inschatten door middel van heuristieken. SFMAC gebruikt transmissies met een lage datasnelheid om controleframes verzenden om burens te informeren wanneer er een dataframe verzonden gaat worden. Transmissies met een hoge datasnelheid worden gebruikt om dataframes te verzenden.  $p$ -CARMA,  $np$ -CECADA en SFMAC presteren niet alleen beter dan klassieke LoRaWAN, maar verbeteren ook de state-of-the-art in termen van kanaalgebruik en de verhouding tussen ontvangen frames over verzonden frames. Tegelijkertijd is het energieverbruik per frame minimaal, terwijl het deel van de energie dat wordt verspild aan botsende frames vele malen lager is dan in klassieke LoRaWAN. De bovenstaande waarnemingen zijn bevestigd middels veldexperimenten die we uitvoerden met behulp van ons testbed van 30 LoRa-apparaten dat scenario's met veel verkeer nabootst.

Op de APP-laag van LoRaWAN ontwerpen we nieuwe coderingsmechanismen voor gegevensherstel van verloren frames. Eerst voeren we een reeks veldexperimenten uit om de omstandigheden te bestuderen die leiden tot frameverlies en symboolcorruptie in LoRa-frames. We gebruiken onze bevindingen om twee coderingsmechanismen voor de LoRaWAN-APP te creëren: Data Recovery (DaRe) en Divide & Code (DC). DaRe voegt aan elk verzonden data frame gecodeerde informatie toe over eerder verzonden frames. De toegevoegde informatie kan worden gedecodeerd om de informatie in voorgaande verloren frames te herstellen. DC codeert LoRa-frames vooraf met behulp van lichtgewicht coderingsschema's. Bij het decoderen geeft DC prioriteit aan de symbolen met de grootste kans op corruptie. Daarom vermindert het de tijd die nodig is voor het decoderen terwijl het bereiken van hoge aantallen gedecodeerde frames. Zowel DaRe als DC presteren

beter dan klassieke LoRaWAN op het gebied van gegevensherstel en besparen energie door het aantal hertransmissies te beperken, terwijl de frames minimaal worden vergroot. Daarnaast presteert DaRe beter dan Luby Transform-codes, vooral in scenario's met een hoge kans op frameverlies, en presteert DC beter dan de standaard Reed-Solomon codering.

Samenvattend hebben we ons in dit proefschrift afgevraagd hoe we de schaalbaarheid in LoRaWAN kunnen verbeteren en tegelijkertijd energie-efficiënt kunnen zijn. We hebben hier een oplossing voor gevonden door nieuwe MAC-protocollen en APP-mechanismen te ontwerpen die backwards-compatible zijn met de LoRaWAN-standaard.



# 1

## INTRODUCTION

*Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.*

Mark Weiser

THE Internet of Things (IoT) has connected the world at an unprecedented scale, by utilizing the Internet to interconnect objects and humans. Through the Internet, interconnected devices, or IoT-devices, can transmit/receive information regarding their operation and act accordingly. Networks of such devices can be used to monitor and control the physical environment in a *smart* way, i.e., without the need for human intervention. Today numerous applications are controlled by such networks, wherein IoT-devices monitor their ambience and provide data streams that are extensive enough to help in decision making. Indicative examples of networks of IoT-devices are smart grids and smart households. Smart grids integrate information technology into the power grid to make it responsive, i.e., grid monitoring supports decision making to cover outages/interruptions and fosters peer-to-peer energy trading [1, 2]. Smart households include hundreds of sensors handling the load shifting of each electrical appliance in the most efficient and/or comfortable way [3]. Furthermore, networks of IoT-devices interoperate and become subsystems (or nodes) of larger networks, e.g., each household –with all its smart appliances– is regarded as a single node for a smart grid. The above level of pervasiveness is further confirmed by the immense volume of IoT-devices. Specifically, 8 billion IoT-devices are registered today only in the domain of smart buildings [4], while the total number of connected devices is expected to be 41 billion by 2025 [5]. By altering significantly the way of operation of multiple industries and changing everyday life in modern cities, IoT has been characterized as one of the most disruptive technologies [6]. The data generated from IoT-devices refer to numerous diverse applications of all market verticals.

However, IoT includes a number of constraints hindering its ubiquitous adoption in smart city and industrial applications. First, the grand majority of current IoT-devices is energy constrained, operating on batteries and/or utilizing energy harvesting to a

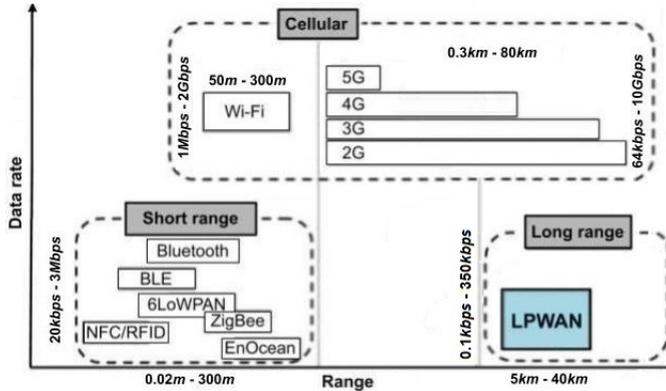


Figure 1.1: Positioning of the main wireless technologies regarding the range of communication and data rate. LPWAN is designed for long-range communications while using low transmission power (keeping energy consumption at low levels).

smaller or larger extent. Further, many of such devices are deployed under conditions that do not favor the frequent replacement of batteries, e.g., under bridges, on top of buildings, or under railway lines. In addition, their low cost and simple design allow only limited computational capabilities. Also, their massive numbers cannot be followed by an analogous production of gateways, especially for non-critical applications, due to the differences in the involved costs. Note that the network/deployment costs of a gateway mostly vary between 100\$-1,000\$, while the average price of an IoT-sensor is 0.4\$ [7, 8]. All the above usually lead to thousands of IoT-devices being served by a few gateways, often located multiple kilometers away. Therefore, IoT-devices are unable to transmit frames frequently since their access to the wireless channel is limited. But even when they access the channel they risk transmitting simultaneously with other devices, which often leads to collisions. Assessing whether frames of other devices are being transmitted, i.e., channel sensing, is expensive energy-wise and computationally demanding, draining the batteries of IoT devices and/or rendering their harvesters useless [9, 10].

To keep up with the monitoring requirements of IoT applications under the aforementioned constraints, network technologies that do not compromise throughput to save energy should be designed. This need –combined with the recent progress in Radio Frequency (RF) technology– led to the rise of Low-Power and Wide-Area Networks (LPWAN). By applying robust encoding on the Physical (PHY) layer LPWANs support long-range communication between devices<sup>1</sup> and gateways efficiently and in a single hop. Further, by using simple, lightweight protocols on the Medium Access Control (MAC) layer, LPWANs can guarantee energy-efficient communication. Through establishing long-range and efficient communication, LPWANs enabled IoT applications that require minimal amounts of energy for small byte-streams to be transmitted to gateways located multiple kilometers away in a single hop. Such applications could be event-driven, served by batteryless devices [11, 12] that need a few hundred micro-watts to communicate, or periodic, served by energy-constrained devices that use batteries [13]. Their needs

<sup>1</sup>The terms devices and IoT-devices are used interchangeably

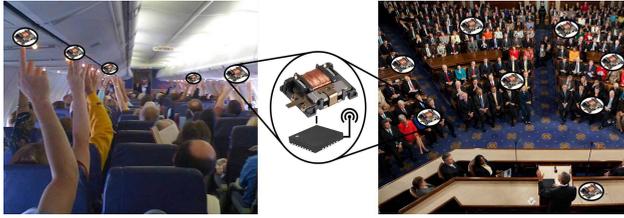


Figure 1.2: IoT applications utilizing wireless and battery-less energy harvesting devices: flight attendant buttons (left) and vote-casting systems (right).

in terms of throughput can reach up to the level of a few hundred kbps. The protocols of the IEEE 802.15.4 family can serve these throughput requirements but they are not designed for transmission-ranges of more than a few hundred meters ( $\approx 200\text{ m}-300\text{ m}$ ). At the same time, long-range technologies are costly energy-wise, requiring the use of high transmission powers. Note that the transmission power of a typical 2G-4G antenna reaches up to 23 dBm ( $\approx 0.2\text{ W}$ ). Actually, as shown in Fig. 1.1, apart from LPWAN none of the other wireless technologies is focused on serving applications like the above, which require the usage of low transmission power for long-range transmissions. On the other hand, LPWAN technologies like Sigfox, Long Range (LoRa)-WAN, Narrowband (NB)-IoT, and LTE-M can offer data rates of 0.1 kbps, 50 kbps, 200 kbps, and 350 kbps, respectively. These rates are provided at ranges of up to 10 km-40 km, depending on the environment (urban/rural), while requiring a maximum of 27 dBm of transmission power [13, 14, 8, 15]. To put transmission power (and thus energy consumption) in perspective, consider that the maximum allowed value of 14 dBm ( $\approx 0.025\text{ W}$ ) for LoRaWAN in Europe is around  $800\times$  lower than the aforementioned transmission power of typical 2G-4G antennas. Therefore, LPWAN technologies have become the backbone of most of today's networks of low-power IoT devices, whose throughput requirements are less than 400 kbps.

## 1.1. THE SCALABILITY REQUIREMENTS OF IOT APPLICATIONS

Although LPWANs can serve certain IoT applications, we are still far from achieving the vision of computer ubiquity as stated by Mark Weiser, wherein the integration of computers into the background of our lives will be seamless [16]. It is stated by Ghen<sup>a</sup> *et al.* that unlicensed LPWANs cannot yet guarantee ubiquitous connectivity for the future masses of IoT-devices and cannot cover their increased needs regarding data transmission due to capacity limitations [15]. Capacity describes a network's throughput over its coverage area. Let us observe certain examples of IoT-applications, and discuss whether their requirements can be covered by current LPWANs. For some applications the throughput requirements are pretty low, e.g., a system of devices reporting twice a day when trash cans are full in an urban territory needs around 1.4 kbit per hour [18]. In the same context of low data requirements, event-driven applications using energy harvesting can be added, like wireless and batteryless flight attendant call buttons or vote-casting systems, see Fig. 1.2 [11]. Such applications can be covered with a few hundred devices connected to a single gateway. However, there are applications that utilize large numbers of devices and report frequently. Such applications either occupy a



Figure 1.3: Partially deployment of CitySee: red nodes and yellow nodes represent carbon dioxide and temperature/humidity sensors, respectively [17].

large part of the network's capacity dedicated strictly to their service, i.e., around one-fifth or more, which is considered non-reasonable [15], or require the deployment of more gateways. For example, urban air quality monitoring applications, like CitySee demand up to 20.4 kbit per second, see Fig.1.3 [17]. In specific, CitySee would take up 40.5% of the capacity of a LoRa-gateway operating at its highest data rate, and thus the deployment of multiple gateways would be required. Furthermore, other applications like the IMT-2020, used to assess performance characteristics of 5G technologies [19, 20], have even higher throughput requirements. IMT-2020 utilizes 1 million devices per square kilometer, each transmitting 16 B per hour. This would result in 71% and 18% of the capacity of a single LoRa and NB-IoT gateway being used up, respectively, just to cover the needs of one square kilometer. However, the capacity requirement mentioned above is expected to become the norm in the near future, since according to 3GPP release 14/15 the Massive IoT standard for 5G is anticipated to support a similar density of IoT-devices (millions per square kilometer) [21, 22]. Considering that the data rate per device may be well above the aforementioned 16 B per hour, in order to serve such applications current LPWANs will have to either deploy multiple gateways, or improve their scalability, i.e., increase their capacity per gateway while enabling the devices to collectively use the most out of this capacity. Deploying more gateways is bounded by the level of reduction in transmission range that can be applied [15]. However, since scalability is the ability of the network to handle even large increases in the load of transmitted data, by improving it the needs of future IoT applications can be met.

## 1.2. UBIQUITOUS CONNECTIVITY USING LORAWAN

Among LPWAN technologies, LoRaWAN has been the most successful, providing easy and inexpensive access to the network and freedom of deployment to both corporate and private parties. In specific, the expenses of setting up a LoRa base station are around 1,000 euros, contrary to Sigfox and NB-IoT for which at least 4,000 and 15,000 euros are required, respectively. Further, LoRa networks do not involve any costs of spectrum usage as they operate at the Industrial Scientific and Medical (ISM) band, with their operating frequencies varying depending on the region (e.g., 868 MHz in Europe, 915 MHz in North America). On the contrary, NB-IoT requires around 500 million euros per acquired MHz of spectrum [14]. Through a series of configurable parameters –including transmission power, carrier frequency, channel bandwidth, Spreading Factor (SF), and

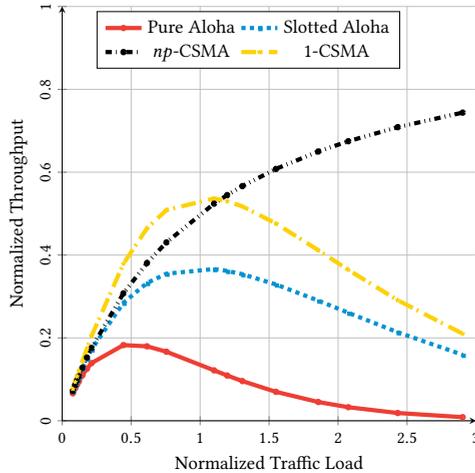


Figure 1.4: Throughput for four main contention-based MAC schemes.

Coding Rate (CR)– LoRaWAN trades off data rate, which can reach up to 50 kbps, to operational range [23, 24]. On the other hand, the data rate of Sigfox is bounded to a maximum of 100 bps, limiting Sigfox to a handful of IoT-applications. LoRa networks offer a multi-kilometer range of communication, which varies between 3-5 km in (dense) urban environments to 10-20 km in (semi-)rural environments, depending on the fading conditions, i.e., shadowing, scattering, Line of Sight (LoS) [25]. Bor *et al.* performed extensive research on the various IoT radios, concluding that LoRa has longer communication ranges over other solutions, such as SigFox and Weightless [26]. Further, free space range measurements, testing different physical layer configurations, showed that the Semtech SX127x LoRa radio family has significant benefits for range, robustness, and battery lifetime compared to competing technologies [27]. Although other LPWANs perform better than LoRaWAN in certain aspects of communication (e.g., Sigfox provides larger range), LoRaWAN outperforms the other leading LPWANs (NB-IoT, Sigfox, and Weightless) in the overall comparison over set-up expenses, battery life, range of communication, and number of applications served [8, 13, 14]. More information on the specifics of LoRa networks will be given in Section 2.1.

In this section, we discuss the main characteristics of LoRa-networks and LoRa-devices that limit the scalability of LoRaWAN, allowing only a few hundred devices to be served per gateway [28, 29]. Further, we explain why methods of improving the scalability are not energy efficient for LoRaWAN. Then we outline the research question of this thesis. We consider a network as scalable if it can achieve high levels of throughput even for very high levels of traffic without collapsing as traffic increases. We define normalized traffic as the fraction of time in which the LoRa-channel is utilized for data (re-)transmissions. For traffic-values above 1, the channel is saturated, since (re-)transmissions need more capacity than what is available from the channel. We define normalized throughput as the fraction of time in which the LoRa-channel is utilized by (re-)transmitted data that are received correctly, which is obviously bounded at 1. To put throughput in perspec-

tive, Fig. 1.4 presents the normalized throughput of four main contention-based MAC schemes [30]. When a frame is generated, depending on the MAC scheme, a device acts as follows: (a) in Pure Aloha, it transmits the frame immediately, (b) in Slotted Aloha, time is divided into slots and the device transmits only at the start of the following slot, (c) in 1-Carrier-Sense Multiple-Access (1-CSMA), the device is continuously sensing the channel and transmits when it is found idle, and (d) in non-persistent ( $np$ )-CSMA, the device backs off for a random duration when the channel is sensed busy and transmits when the channel is sensed free. CSMA-schemes achieve higher throughput by sensing the channel, which helps evade simultaneous transmissions of frames, thus evading collisions. However, if two (or more) devices are transmitting to the same gateway while not being able to sense each other's transmissions, i.e., they are hidden to each other, then simultaneous transmissions will happen [31]. When the traffic is relatively light 1-CSMA achieves higher throughput since the chances hidden devices are transmitting when a device has found the channel idle are relatively low. However, as the traffic increases 1-CSMA collapses under the sheer number of undetected transmissions. On the other hand, the  $np$ -CSMA protocol is the only one that manages an increasing normalized throughput—even after channel saturation (1 at x-axis)—due to the "pessimistic" approach it adopts; each device expects a heavily loaded network when the channel is sensed as busy and reschedules sensing further in time, allowing possible transmissions by hidden devices to happen without interference. Therefore, we consider LoRaWAN as scalable if it can achieve a performance similar to  $np$ -CSMA. However, the performance of the current vanilla LoRaWAN resembles that of pure Aloha in large deployments with high traffic needs, with the network collapsing at traffic values higher than 0.5 (normalized) [32]. The low scalability of LoRaWAN is due to the following reasons:

1. **The coexistence of LoRaWAN and other network technologies operating at the ISM spectrum.** ISM spectrum is unlicensed, so the operating frequencies are not reserved for the applications of specific networks, for example, in the context of smart cities. Since coexistence among different network technologies is unresolved, they operate simultaneously without coordination. This leads to interference, packet collisions, and ultimately decreases the upper bound of scalability of LoRaWAN.
2. **The energy constraints of LoRa-devices.** This fact indirectly limits capacity, and thus scalability, by not allowing CSMA techniques to be applied on the MAC layer of LoRaWAN to increase the channel throughput. CSMA approaches require active sensing of the medium to assess its idleness, and this would drain the battery of any LoRa-device.  
  
Since active sensing is not an option, the LoRaWAN-MAC uses an unslotted Aloha-like protocol, without assessing the channel for other ongoing transmissions. The normalized throughput of unslotted Aloha channels is bounded at 0.18 for a traffic load of 0.5, as seen in Fig. 1.4. Any further increase in the fraction of transmissions leads to humongous numbers of frame collisions rendering the network unable to scale and serve the traffic load of dense deployments [30, 33].
3. **The long communication distances.** LoRa is designed for single-hop and multi-kilometer communication, avoiding the use of intermediate nodes and multiple

Characteristic	Consequence
Use of unlicensed spectrum	Interference from coexisting networks
Energy-constrained devices	Unable to perform Carrier Sensing Bounded throughput at 0.18 (normalized)
Long range of communication (multi- km)	Frame corruption/loss Sectors of hidden devices
Asynchronous and unidirectional communications	Unable to use ACKs and scheduling

Table 1.1: The main characteristics of LoRaWAN and LoRa-devices as limiting factors of scalability.

gateways. However, despite the robust design of LoRa-frames, multipath propagation and physical phenomena, like shadowing and scattering, have a strong impact on LoRa transmissions, especially in Non-Line of Sight (NLoS) environments, leading to frames being received with very low Received Signal Strength Indicator (RSSI), i.e., -115 dBm. Because of such low values of RSSI, many times LoRa frames are lost and/or suffer from symbol corruption, including bursts of frames/symbols [28, 25, 34], regardless of their robustness against fading. These lost/corrupted frames need to be retransmitted in case acknowledgment (ACK) messages are used, creating an overhead that limits the scalability of LoRaWAN and increases energy consumption.

Furthermore, the increased impact of fading in long-range communications results in the creation of several sectors of devices hidden from each other in the network. Even if a non-expensive (energy-wise) application of carrier sensing was created for the MAC of LoRaWAN, scalability would be hindered by frame collisions from hidden devices [31].

- Asynchronous and unidirectional communication.** LoRa-networks are asynchronous in communication, with uplinks being the grand majority of transmissions. In addition, there is no immediate feedback channel from the gateway to each device. Therefore, Time Division Multiple Access (TDMA) approaches cannot be used to regulate transmissions through scheduling messages among the devices and a central authority. TDMA is proven to increase the ratio of received frames and the utilization of the channel for successful transmissions by granting access to the channel to each device only at specific time-slots [35, 36]. Without TDMA, as the number of overlapping transmissions increases collisions occur more frequently, obstructing the network's scalability.

Table 1.1 sums up the aforementioned characteristics of LoRaWAN which limit the capacity and hinder the scalability in large LoRaWAN deployments. Although by utilizing more LoRa-gateways we can meet the capacity requirements of current IoT-applications, this is not economically viable due to the sheer number of IoT-devices that are expected to be deployed in the future. Consider that the usual cost of an indoor LoRaWAN gateway is at 300\$ while gateways for outdoor usage surpass the 1,000\$, i.e., at least  $750 \times -2,500 \times$  the cost of a IoT-sensor [7, 37, 8]. But even if the aforementioned costs are not taken into account, applying large numbers of gateways is limited by the reduction in transmission range that is achievable without having gateways receiving from the same (multitudes of) devices [15]. Furthermore, since there are already numerous applications using LoRaWAN, any proposal that deviates from the LoRaWAN-standard would incur issues of

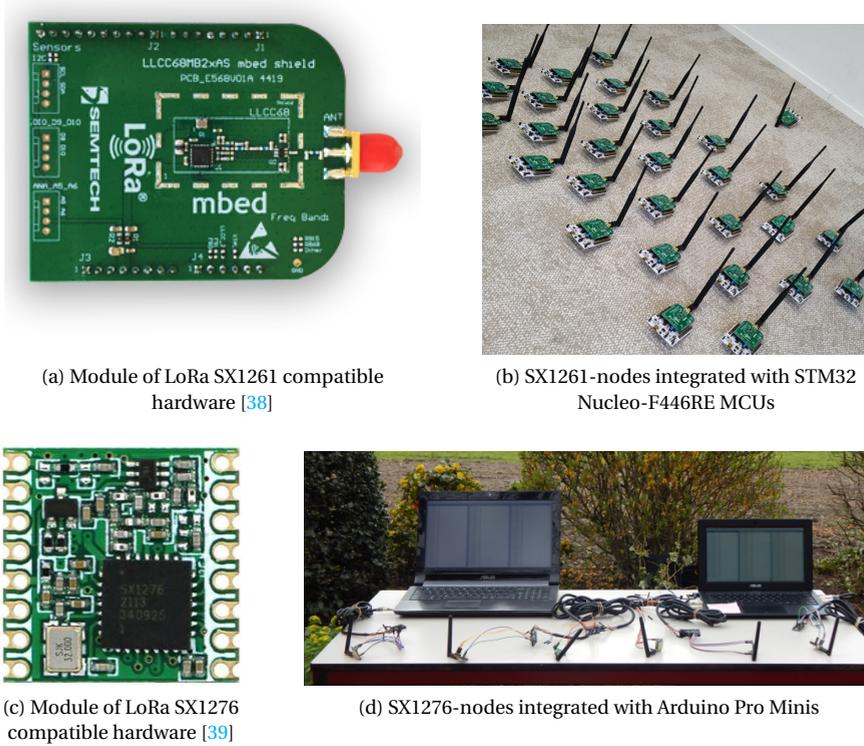


Figure 1.5: The testbed of LoRa-nodes utilized in this thesis.

interoperability among LoRa-devices. Taking the above into account, this thesis poses the following research question:

**How to improve the scalability of LoRaWAN  
in an energy-efficient, economical, and backward-compatible manner?**

We will leverage a mechanism of LoRa called Channel Activity Detection (CAD) (despite its limitations) and the Capture Effect (CE) phenomenon taking place at the physical layer. CAD is a built-in mechanism used by LoRa-devices to assess the existence of ongoing transmissions on the channel [40]. CE refers to the successful reception of a frame regardless of being interfered by other transmissions due to their lower received signal strength and/or slight delay of reception [41, 42, 43]. Our approach is to develop algorithms for the MAC and the APP layer of LoRaWAN incorporating our observations regarding CAD and CE. It is crucial that our algorithms are energy efficient and of low complexity to align with the devices' energy constraints and low computational capabilities. We utilize analytical/numerical methods in order to reason the design of our algorithms. Further, we perform simulations for the cases in which our protocols must be evaluated on thousands of devices, utilizing the *ns-3* simulator [44]. In *ns-3* we build all the modules needed

to model complete LoRa-PHY and LoRa-MAC layers, incorporating all the attributes of our novel algorithms. Finally, we conduct field experiments using our own testbed of SX1276 and SX1261 LoRa devices (see Fig. 1.5), to validate our protocols and algorithms in real-world conditions.

### 1.3. CONTRIBUTIONS AND OUTLINE

In this thesis, we introduce multiple approaches at the MAC and the APP layer to improve the scalability of standard LoRaWAN while being energy efficient and cost-effective. Through our algorithms LoRaWAN achieves high levels of normalized throughput, reaching even up to 75%, which is  $4.17\times$  more than the current maximum value of vanilla LoRaWAN. Further, these values of throughput are still observed at high levels of traffic, e.g., three times above the level of saturation, i.e., normalized traffic of 1. In general, we contribute as follows:

1. We conduct in-field experiments: (i) to evaluate the performance and limitations of the CAD mechanism for channel sensing and (ii) to evaluate the capture effect in LoRa with respect to both power differences and delay offsets and derive the probabilities of successful transmissions (Section 2.3).
2. Taking into account the findings of the above in-field evaluations, we create distributed MAC protocols based on CSMA principles (Chapters 3 and 4) and scheduling principles (Chapter 5), adapted to the specifics of LoRaWAN, which improve channel utilization up to  $15.74\times$  over vanilla LoRaWAN.
3. We introduce application-layer coding techniques for the recovery of data from lost and corrupted LoRa frames (Chapter 6), which consume 17% less energy compared to repetition coding while achieving 99% data recovery.

Let us outline the contributions of each individual chapter.

**Presentation of LoRaWAN and evaluation of CAD and CE – Chapter 2.** In this chapter, first we provide the background of a LoRa network, explaining the operations taking place at the physical and the MAC layer. Further, we discuss the related work regarding the MAC layer protocols designed for LoRaWAN. Then, we focus on the results and insights acquired from our in-field experiments studying the behavior of CAD and CE. These insights are incorporated into the algorithms of the MAC layer protocols that are presented in Chapters 3, 4, and 5. We present the relation between CAD performance and the distance between LoRa-devices and gateways. Also, we show that a non-trivial percentage of frame detections refers to different SFs than the configuration of the attempted CADs (up to 17%). These false detections can lead to channel underutilization due to unnecessary avoidance of transmissions. Regarding CE, we observe that if the first 4-8 symbols of a LoRa-frame, called preamble, are received then the corresponding transmission has great chances to capture the channel even if it is interfered in the remaining symbols.

**A persistent distributed MAC protocol for LoRaWAN – Chapter 3.** LoRaWAN's minimalist design in the MAC layer cannot handle the traffic from deployments with more than a few hundred devices connected to a single gateway, because each LoRa-device transmits data-frames without any information regarding the availability of the medium.

This hinders the scalability of LoRa networks. To this end, we propose carrier sensing on the MAC layer of LoRaWAN to reduce the collisions and to maximize channel utilization in the network. We exploit the CAD functionality of LoRaWAN to assess channel occupancy while being energy efficient. We combine our findings from the in-field evaluation of CAD with the principles of persistent-CSMA ( $p$ -CSMA) in order to design our protocol, called  $p$ -persistent Channel Activity Recognition Multiple Access ( $p$ -CARMA) [45]. Since LoRaWAN uses minimum feedback from the gateway and includes many hidden devices we design a localized and distributed algorithm that adapts the persistence  $p$ -value per device. Considering Packet Reception Ratio (PRR), i.e., the number of received over the number of transmitted packets,  $p$ -CARMA outperforms vanilla LoRaWAN from  $3\times$  to  $20\times$  while handling thousands of devices. In terms of energy consumption,  $p$ -CARMA spends 37.31%-58.17% less energy compared to vanilla LoRaWAN.

- Nikolaos Kouvelas, Vijay S Rao, R. Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. “ $p$ -CARMA: Politely Scaling LoRaWAN”. in: EWSN '20. Lyon, France

**A non-persistent distributed MAC protocol for LoRaWAN – Chapter 4.** The focus of the chapter remains on the MAC layer of LoRaWAN, targeting the improvement of the scalability of LoRa networks. However, our approach is different because (i) we incorporate and leverage critical observations from in-field evaluation of the Capture Effect (CE) phenomena, and (ii) we combine CAD to the principles of non persistent-CSMA ( $np$ -CSMA). We design non persistent-Capture Effect Channel Activity Detection Algorithm [46].  $np$ -CECADA is an adaptive, distributed, backoff time-based CSMA protocol for LoRa networks of multiple hidden terminals and minimum feedback.  $np$ -CECADA utilizes the CAD mechanism to evade collisions and the probabilities of CE to regulate the transmission power per device, saving energy consumption per device without compromising PRR. At the same time, this reduces the interference to all networking technologies coexisting in the ISM band (including LoRaWANs). Each device adapts its backoff time in a distributed manner by estimating the traffic load in its vicinity.  $np$ -CECADA outperforms state-of-the-art algorithms, increasing PRR by  $5.13\times$  compared to  $p$ -CARMA [45] and channel utilization by  $11.24\times$  compared to LMAC [47], while consuming solely between 2 mJ-47 mJ per frame transmission. Furthermore, when evaluated in-field against vanilla LoRaWAN using a testbed of 30 LoRa-devices,  $np$ -CECADA increased PRR by 5 times at SF7 and traffic load of 1.5 (normalized per second).

- Nikolaos Kouvelas, R Venkatesha Prasad, Niloofar Yazdani, and Daniel E. Lucani. “ $np$ -CECADA: Enhancing Ubiquitous Connectivity of LoRa Networks”, IEEE Conference on Mobile Ad-Hoc and Smart Systems (MASS), 4-7 October 2021, Denver, CO, USA.

**A transmission scheduling protocol through dedicated SFs – Chapter 5.** This chapter, although referring again to the MAC layer of LoRaWAN, deviates from the strict contention-character of accessing the medium that was described in the previous chapters, and applies a virtual TDMA technique called Spreading Factor (SF)MAC. For this, we turn to low data-rate channels (i.e., high SFs) that are underutilized in Smart City scenarios due to being energy inefficient and unable to carry large payloads [24, 48]. Upon transmitting a data-frame at high-data rates (SF7-SF8), SFMAC-devices switch to low data-rate channels (i.e., SF9-SF11) solely for the transmission of control-frames,

notifying other SFMAC-devices attempting to transmit data-frames that they just finished their transmissions. Therefore, high SF channels are used to establish an order among transmissions of random arrival rates that happen to coincide, i.e., performing TDMA virtually. Committing high-SF traffic to sensing comes without any considerable cost in terms of capacity, since a commonly used method for increasing data rate for Smart City applications is to increase the number of gateways, which creates smaller cells of devices transmitting at low SFs [15]. At the same time, it does not require any considerable amount of energy because SFMAC-devices switch among SFs only whenever a new frame is generated, in order to sense for control frames. In order to choose the proper SF for the control channel in terms of energy efficiency and reduced channel interference, we defined and solved an optimization problem. SFMAC improves capacity compared to vanilla LoRaWAN by  $2.16\times$ . Further, compared to LMAC [47], SFMAC increases goodput by  $6.25\times$  and compared to  $p$ -CARMA [45], SFMAC increases PRR by  $1.72\times$ .

**Coding for data recovery at the application layer of LoRaWAN – Chapter 6.** LoRaWAN experiences high amounts of lost and corrupted frames due to its coexistence with other ISM-networks, to its Aloha-like MAC, and to the multi-kilometer range of communication. As observed through real-world experiments the correction capability of LoRa gets outperformed by the sheer number of corrupted symbols [28, 25, 34]. Frame loss and corruption can also happen in bursts, especially in scenarios involving mobile devices. Any corrupted frame is rendered useless and thus needs to be retransmitted. Minimizing the corruption/loss of data is critical for LoRaWAN because, apart from increasing the data reception ratio, it saves energy from frame retransmissions. Furthermore, minimizing retransmissions creates less interference not only for LoRaWAN but also for all networks that coexist in unlicensed bands.

To recover the data from lost frames we present a novel coding scheme called Data Recovery [49]. DaRe works on the application layer of LoRaWAN, combining techniques from convolutional and fountain codes. DaRe extends each LoRa-frame with redundant information that results from the combination of linear operations on previous frames. Therefore, the produced redundancy by a single frame is spread across other upcoming frames depending on a sliding window, i.e., memory-based coding. Thus, if the frame is lost, a considerable amount of its information can still be reconstructed by decoding the following frames that were correctly received. To exploit DaRe to its fullest potential we survey an extensive range of configuration parameters in simulation giving a clear insight into the trade-off involved between energy consumption and data reception ratio. Furthermore, we compare DaRe to the widely accepted state-of-the-art Luby Transform (LT) fountain codes [50], showing that DaRe outperforms LT especially at high values of frame-loss probability, achieving up to  $2.29\times$  improvement in data reception ratio [51].

- Paul J. Marcelis, Nikolaos Kouvelas, Vijay S. Rao, and R. Venkatesha Prasad. “DaRe: Data Recovery Through Application Layer Coding for LoRaWAN”. in: *IEEE Transactions on Mobile Computing* 21.3 (2022), pp. 895–910

However, due to the use of a sliding window DaRe cannot be applied to time-critical applications since the recovery of a lost frame’s data depends on a number of upcoming frames. To tackle data corruption in real-time we introduce Divide and Code (DC), a novel coding scheme for the application layer of LoRa networks [52]. DC pre-encodes

LoRa-payloads by using a lightweight coding scheme that adds a limited number of extra symbols before the addition of the CRC. This pre-encoding state uses linear combinations among the symbols of the current frame, i.e., memory-less coding. We evaluate realistically the frame-corruption in LoRaWAN and draw insights on the patterns in which (bursts of) errors occur and on the correlation among errors in the same frame. We incorporate these insights into the decoder of DC to reduce the decoding time while increasing the decoding ratio. DC was evaluated in-field using LoRa SX1261 transceivers. Compared to Reed Solomon coding and vanilla LoRaWAN, DC boosts the decoding ratio by  $\approx 2\times$  and  $2.5\times$ , respectively, while consuming  $1.5\times$  and  $\approx 2\times$  less energy per correctly received data-bit, correspondingly.

- Niloofar Yazdani, Nikolaos Kouvelas, R. Venkatesha Prasad, Daniel E. Lucani, “Energy Efficient Data Recovery from Corrupted LoRa Frames”, IEEE Global Communications Conference (GLOBECOM), 7-11 December 2021, Madrid, Spain
- Niloofar Yazdani, Nikolaos Kouvelas, R. Venkatesha Prasad, Daniel E. Lucani, “Divide and Code: Efficient and Real-time Data Recovery from Corrupted LoRa Frames”, Under review (2022).



# 2

## LONG RANGE WIDE AREA NETWORKS (LoRAWAN)

### *Physical and Medium Access Control layer*

**I**N this chapter, we first provide thorough information on the operation of LoRa networks (Section 2.1). Then, we discuss the related work referring to MAC layer protocols for LoRaWAN (Section 2.2). Finally, we elucidate our observations based on the in-field experiments to characterize CAD and CE in LoRaWAN and present the results and the main take-away messages (Section 2.3). These results act as the basics for the design of our MAC protocols in Chapters 3, 4, and 5.

### 2.1. BRIEF INTRODUCTION TO LoRAWAN

**Physical layer.** The PHY layer, called LoRa, is proprietary and owned by Semtech [54]. LoRa utilizes a type of frequency modulation called Chirp Spread Spectrum (CSS), where each symbol is encoded in a number of chirps, i.e., signals of continuously increasing/decreasing frequency over the given bandwidth of operation (125, 250, 500 kHz), as seen in Fig. 2.1. The Spreading Factor (SF) dictates how many raw bits of information are used per transmitted symbol, trading-off data rate to encoding robustness (e.g., each SF7 symbol has 7 bits =  $2^7 = 128$  chips). Through several real-world evaluation campaigns, it was proven that, under the proper configuration of LoRa-parameters, LoRa-CSS can provide encoded signals which are robust against noise, multi-path fading, and resistant to the Doppler effect, even in mobile scenarios. In fact, the robustness of the LoRa-links allows chirps to be decoded even below the noise level [55, 56, 57]. Fig. 2.2 shows the main structure of a LoRa frame. Its preamble is usually comprised of 8 continuous upchirps (or downchirps), although 4.25 is the lowest number of chirps required for the preamble to get locked on a LoRa-receiver. Then, the header and the payload follow. The payload

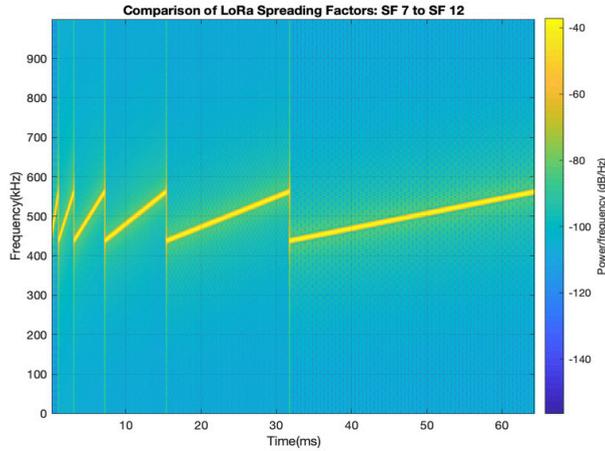


Figure 2.1: Chirps for different SFs operating at 125 kHz [53].

Preamble	Header	Payload	CRC (opt)
Min: 4.25 symbols	2B-4B	Max: 51B (SF12) - 255B (SF7)	2B

Figure 2.2: The basic structure of LoRa-frames.

varies in size depending on the SF that is utilized. As seen in Fig. 2.1, higher SFs encode the information into "longer" symbols (i.e., higher Time on Air (ToA)) in order to increase the robustness of the signals. Therefore, their maximum allowable payload is limited. Frames modulated by different SFs at the same bandwidth can be received simultaneously even in the same channel, as SFs manifest signal orthogonality. Although the simultaneous transmission of frames at the same SF and channel generally leads to collisions, under certain circumstances one of the transmissions can be received successfully by dominating the channel, because of Capture Effect (CE). See subsection 2.3.1 for the presentation and thorough evaluation of CE in LoRaWAN.

**Medium Access Control layer.** The MAC layer of LoRa, called LoRaWAN, is an open standard, developed by the LoRa Alliance [58]. LoRa-networks form star topologies, wherein the frames broadcasted by devices are directly received by one or more gateways, which forward the information to the network server, as seen in Fig. 2.3. Devices are half-duplex, with most transmissions being uplink, i.e., from LoRa-devices to gateways. Operating at the ISM band involves also a series of regulations that are in place to ensure the coexistence of LoRaWAN with other networks, such as WiFi, and Bluetooth. First, the duty-cycle is limited to 1%, allowing 36 s per hour for transmission per device; for certain frequency bands and only for gateways the limitation of the duty-cycle can be relaxed to 10%. Further, the maximum value of power that can be used for the transmission of payloads is bounded depending on the region (e.g., 14 dBm in Europe and 27 dBm in the United States). Due to duty-cycle and energy constraints, devices are rarely found in

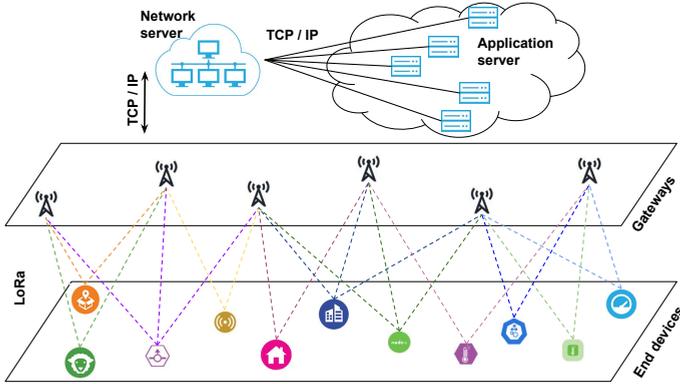


Figure 2.3: Schematic overview of a LoRaWAN network.

receiving mode, i.e., for acknowledgment (ACK) packets, network joining, and/or gateway broadcasts. The duration spent on receiving mode by a device depends on its class and allows the device to improve its channel utilization and packet reception. There are three classes of communication – A, B, and C. In *Class A*, devices can use up to two receiving windows for listening after each transmission. In *Class B*, devices add extra receiving windows at predetermined times communicated to them by the gateways through beacons. In *Class C*, devices are continuously receiving (except when transmitting), being powered by mains [59]. When packets are generated, LoRa-devices access the channel in a manner similar to pure Aloha, transmitting their frames unconditionally regardless of any ongoing transmission in the medium<sup>1</sup>. Furthermore, LoRaWAN provides two mechanisms for the improvement of the network capacity and the avoidance of frame collisions at the gateway. The first mechanism is *Adaptive Data Rate (ADR)* and allows the devices to dynamically change the SF they use for transmission and reception. When ADR is enabled the gateway broadcasts the SF at which the network server can receive uncorrupted frames of adequate Received Signal Strength Indicator (RSSI), based on an analysis performed by the server on the robustness of the already received signals [60]. The second mechanism is *Channel Activity Detection (CAD)*, a built-in mechanism used by devices to assess the existence of ongoing transmissions of LoRa-frames of a specific spreading factor on the medium. CAD includes two phases; reception and processing. During the reception phase, the device searches for LoRa-symbols of a preconfigured SF. Then, in the processing phase, any captured signal is cross-correlated to a buffered waveform in order to determine whether the channel is occupied [61, 62].

**Data Recovery.** LoRa-modems utilize Forward Error Correction (FEC) for data recovery. In particular, Hamming codes [63, 64] are used to create data-redundancy, allowing a Code Rate (CR) of  $\frac{4}{4+x}$  where  $x \in \{1, 2, 3, 4\}$ . Namely, every 4-bit part of the frame is encoded into  $4 + x$  bits. CR-values of  $4/5$  and  $4/6$  are capable only of single error detection, while  $4/7$  and  $4/8$  can correct single errors or detect double errors. As seen in Fig. 2.2, a Cyclic Redundancy Check (CRC) of 2 B is optionally added at the end of the payload

<sup>1</sup>Throughout this thesis the term medium refers to the channel by default.

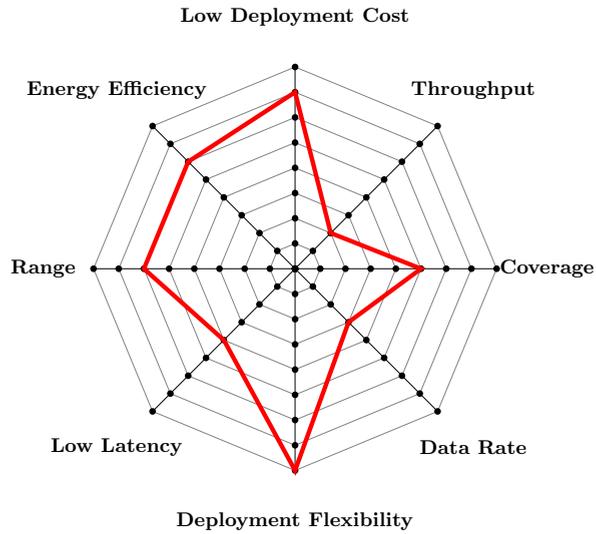


Figure 2.4: Qualitative evaluation of LoRaWAN.

to support the error detection. Further, the header of a LoRa-frame may include a 4-bit CRC for the same purpose. Finally, to alleviate the tampering of messages LoRa-payloads offer the choice of a 4 B Message Integrity Code (MIC). MIC is a Cipher-based Message Authentication Code (CMAC) that assigns device-specific signatures, which are created by using keys shared only between the device and the network server. The reception of LoRa-frames relies primarily on detecting the frame-preambles, which –if not detected– usually lead to complete loss of the frame [65].

Fig. 2.4 characterizes LoRa networks based on 8 network characteristics. As seen, due to the use of CSS, unlicensed ISM bands, and a simple MAC protocol, LoRaWAN achieves a long range of communication, easy/cheap deployment, and high energy efficiency. However, with respect to other aspects like throughput and coverage, LoRaWAN does not perform equally well. This hinders its scalability as it cannot cover the capacity requirements of many applications.

## 2.2. MAC-LAYER PROTOCOLS FOR LORAWAN

In this section, we discuss MAC protocols proposed to increase the capacity of LoRa networks.

**MAC protocols for LoRaWAN.** Most of the approaches not involving CSMA are variants of time-scheduling. In gateway-initiated approaches, devices are aperiodic transmitters triggered by sensing queries from the gateways [66, 67, 68]. Polonelli *et al.* introduce a slotted Aloha-variant on top of the pre-existing LoRaMAC, wherein devices use time-slots for the synchronization of transmissions at the beginning of each time-slot [66]. In On-demand LoRa gateways use long-range asynchronous transmissions to disseminate requests to nodes acting as Cluster Heads (CH). Then, CHs broadcast beacons to their

neighbors which use wake-up radios for efficient reception. Finally, using the beacon arrival times, devices synchronize their transmissions directly to the gateway using their long-range LoRa-antennas [67]. Hasegawa and Suzuki proposed to increase the data rates by reducing the traffic due to ACK-frames. To this point, devices use synchronized receiving windows to receive cumulative ACKs simultaneously. Each device uses Bloom filters to keep only its corresponding ACK [68].

In device-initiated approaches, each IoT-device requests channel access from the gateway by transmitting information regarding its transmission characteristics, i.e., the size of data-frames, packet inter-arrival time, and clock accuracy. Based on this information, and accounting for the traffic characteristics of the network and positions of the nodes, the gateway schedules the communication [69, 70, 71, 72]. To and Duda introduce gateways that construct temporal transmission schedules [69], while Haxhibeqiri *et al.* utilize Bloom filters to disseminate such schedules in a space-efficient format [70]. In FREE each device buffers its data packets in local storage and transmits them in bulk to the gateway. Based on the transmission requirements of each device, the gateway divides transmission time per channel into slots, and schedules sequential transmissions of the same SF and simultaneous transmissions of different SFs, before disseminating the transmission schedule to the network [71]. In S-MAC, adaptive scheduling is introduced to describe a MAC protocol that perceives the clock-drifts of devices that transmit periodically to update transmission schedules dynamically, according to the transmission characteristics of these devices. Specifically, the frequency band is divided, and the maximum number of collision-free, concurrent, periodic transmissions per SF are grouped and assigned to the same frequency channel for transmission, i.e., Frequency Division (FD)MA [72]. TS-LoRa and CRAM still use device-initiated scheduling but in a relatively distributed fashion [73, 74]. The time-slots in TS-LoRa are assigned using a hash-algorithm with a modulo operation that translates the ID of each device into a unique time-slot inside every transmission round (whose duration is empirically determined). The only input needed from the gateway is the length of the frame [73]. CRAM distributes transmissions across the available LoRa frequency spectrum by using cryptographic frequency hopping, i.e., transforming an arbitrary channel mask into a transmission frequency by employing cipher encryption to protect frames of high importance from selective jamming. Coordination is achieved between gateway and devices through shared keys. Devices sharing the same channel frequency are assigned to different time-slots [74].

In hybrid approaches, a part of the communication is time or frequency scheduled, and another part is based on Aloha [75, 76, 77]. MAC on Time (MoT) is a centralized synchronous MAC protocol for LoRaWAN, wherein each IoT-device transmits connection requests to the base station in an Aloha fashion during the connection phase. The centralized base station approves/denies these requests, scheduling accordingly the data-communication in TDMA, which takes place in the reporting phase, in order to prioritize critical applications [75]. In RS-LoRa the gateway schedules devices in equally divided transmission-frames per channel, and subframes per SF. Then, the medium-access in subframes takes place in Aloha-fashion. The scheduling is lenient, as the gateway defines allowable value-regions of SFs and transmission power per subframe to minimize frame losses. Each device adapts its transmission parameters in a distributed manner following beacons broadcasted by the gateway carrying information [76]. In RT-LoRa aperiodic

devices transmit in contention time-frames, in which the channel is accessed using slotted Aloha, and periodic devices transmit in contention-free time-frames using TDMA communication based on three different classes of Quality of Service (QoS) [77].

Centenaro and Vangelista propose to use different channels for transmission and reception. They utilize 868.1 MHz-868.5 MHz for transmitting, and dedicate 869.525 MHz strictly for receiving, thus allowing 10% duty-cycle and an adaptable receiving power of at most 27 dBm. This reduces the interference between transmitted and received frames. Further, they suggest only one receiving window of increased length per device, and use the 27 dBm on 869.525 MHz to multiplex six parallel downlink channels of a higher power. In this way, the gateway can transmit simultaneously on downlink-frames of all SFs with higher power allowing extra protection against interference [78].

The MAC protocols that involve time-division, either throughout the procedure of accessing the channel or in parts of it (hybrid approaches), can improve channel utilization and evade frame-collisions when the network traffic is high because each (group of) device(s) transmits in predetermined time-slots. However, these protocols can become inflexible when applied to dynamic networks like LoRaWAN, wherein devices join/leave the network at random and often transmit aperiodically. This is because the gateway needs to be continuously updated of any change in the network to recalculate channel-time for all devices. This can cause delay, leading to underutilization of the channel, and thus to low levels of network capacity.

**CSMA mechanisms in LoRaWAN.** To and Duda initially investigated how CSMA protocols could be adapted to LoRa networks to reduce frame collision while maintaining energy consumption at acceptable levels. They evaluated CSMA with backoff periods in LoRaWAN, using simulation and taking also into account the impact of capture effect on the drop rate [82]. Ortín *et al.* designed heterogeneous LoRa networks involving –apart from vanilla LoRaMAC operating devices– nodes utilizing unslotted Listen Before Talk (LBT) from IEEE 802.15.4 standard. They used the Markov chain model to assess channel occupancy using energy detection and perform Binary Exponential Back-off (BEB) if the channel is busy [79]. Beltramelli *et al.* derived an analytical model for the performance of LoRaWAN under (slotted) Aloha and  $np$ -CSMA regarding throughput, coverage, and consumption, accounting CE in their analysis. They validated their model using Monte-Carlo simulations. Further, they characterized numerically and experimentally the capture effect and accounted for it in their analysis [83]. Triantafyllou *et al.* introduced FCA-LoRa, wherein the gateway broadcasts beacon-frames periodically to synchronize the communication with the devices. Upon receiving these beacons the devices initiate a CSMA procedure based on sensing the channel and backing off in case of ongoing transmissions [84]. Liu *et al.* proposed a combination of  $p$ -CSMA and CDMA for LoRaWAN-MAC.  $p$ -CSMA assigns different probabilities of transmission,  $p$ , depending on the traffic. CDMA is performed in the context of the  $p$ -CSMA to increase further the throughput by dividing the transmissions into multiple channels [85]. Baddula *et al.* compared through simulation the performance of Aloha and CSMA MAC for LoRaWAN in terms of throughput, frame collisions, and energy consumption, showing that CSMA should be preferred for large LoRa networks [86].

Pham first investigated the utilization of CAD in CSMA approaches for the MAC layer of LoRaWAN, by mapping the DIFS and the backoff windows of IEEE 802.11 and IEEE

MAC protocol	Effectiveness (PRR × G)	Consumption per byte [mJ]	Overhead	Medium Access	Complexity	Gateway Dependence	Evaluation
Polonelli <i>et al.</i> [66]	0.15	0.635	+8 B per ACK	slotted Aloha	Low	Medium	R,S
On-demand LoRa [67]	0.05	0.156-4.137	beacon-system (2 B/beacon)	beacon-synchronized	High	High	R
MoT [75]	0.41	N/A	ACKs + scheduling-beacons	Aloha (request)/ TDMA (data)	Medium	Medium	S
Hasegawa and Suzuki [68]	0.09	N/A	Aggregated ACKs of 4-15 B	Aloha	Medium	Medium	S
To and Duda [69]	0.13	4.831-6.78	Join/Sync/Scheduling Messages	TDMA	High	High	S
Haxhibeqiri <i>et al.</i> [70]	0.09	N/A	Syns of 15 B and 28 B	TDMA	High	High	S
TS-LoRa [73]	0.29	2.255 plus 7830 (once)	Join (20 B) and ACK (100-251 B)	distributed time-slotting	High	Low	R,S
CRAM [74]	0.15	0.59-11.89	~ 5 B sync beacons	frequency hopping/ TDMA	High	Low	R
FREE [71]	0.1	~0.521 plus 262.5 (once)	Join (16-24 B) and Sync (7-23 B)	TDMA	High	High	S
RS-LoRa [76]	0.17	N/A	sync-data beacons of 8-20B	TDMA/ Aloha	Medium	Medium	S
Rf-LoRa [77]	0.06	0.049-0.201	extra field in UL and ADR messages	FDMA	High	High	R
Cantennaro and Vangelista [78]	0.14	0.425-2.188	~ 10% communication time for beacons/ACKs	TDMA/ slotted Aloha	Medium	Medium	S
Ortún <i>et al.</i> [79]	0.05	0.082-1.109 plus LBT	-	Aloha	Low	Low	S
Rochester <i>et al.</i> [80]	0.09	0.399-10.335 plus CAD	-	LBT/ Aloha	Low	Low	S
LMAC [81]	0.13	3.02	periodic beacons of 49 B (class-B)	CSMA	Low	Low	S
				channel-hopping using CS	Medium	Low	R

Table 2.1: Comparison of state-of-the-art MAC protocols for LoRaWAN (G: normalized traffic R: real-case, S: simulation)

802.15.4 to consecutive CADs [87]. Kim *et al.* designed a policy that allocates specific SFs to pairs of transmitting/receiving devices in order to improve the throughput in multi-hop LoRaWAN. Before assigning a new SF, this policy inspects the SF in which every device transmits by sensing the channel using CAD [88]. Liando *et al.* investigated the implementation of preamble-CAD detection as Carrier-Sensing (CS) mechanism in LoRaWAN. If any ongoing non-orthogonal transmission was sensed, the device attempting to transmit had to change its SF [56]. Rochester *et al.* utilized single-symbol CAD, enabling devices to infer the channel occupancy with minimum energy consumption through sensing for frame preambles. If the channel was found busy, the device was dropping the frame evading a possible collision and any further energy consumption due to sensing [80]. Gamage *et al.* utilized LoRa-devices able to sense any part of a frame (preamble, header, payload) through CAD. Based on their study of CAD as a CS-mechanism they designed LMAC [81], a series of CSMA-based MAC protocols of increasing complexity; spanning from LMAC-1 that translates DIFS in a number of CADs to LMAC-3 wherein the gateway broadcasts global views of the network through beacons, to assist devices in their distributed method of channel hopping. They evaluated LMAC through a testbed of 50 LoRa devices and a gateway. O’Kennedy *et al.* evaluated the CAD both in-field and by simulation to determine its applicability as part of a CSMA mechanism in a LoRa mesh network for wildlife monitoring [89]. They observed successful frame detection using CAD even at distances of 4 km among LoRa devices. Further, they compared the effectiveness of CAD in preamble-sensing versus payload-sensing of LoRa frames. Xu and Zhao introduce REDS, a strategy in which devices first use CAD to decide whether the channel is occupied. Then, depending on the SNR of CAD the devices estimate if the network traffic is high or low, using dynamic duty-cycling or CSMA, respectively, to access the channel [90]. Kamonkusonman and Silapunt attempted to improve the frame reception in LoRaWAN by introducing CAD-MB, a protocol in which every device backs off before applying CAD. If the channel is found free after two consecutive CADs the frame transmission takes place [91].

The distributed manner of accessing the channel that is offered by CSMA can be leveraged to accommodate higher traffic. However, the application of CSMA on the MAC layer of LoRaWAN has been mostly hindered by the energy requirements of active sensing on the energy-constrained LoRa-devices. By using the CAD mechanism as a means of channel sensing, LoRaWAN can enjoy relatively effective avoidance of collisions while consuming low amounts of energy. Furthermore, the improved versions of the CAD mechanism that can detect any part of an ongoing LoRa-transmission (SX126x-devices) open the path to efficiently employ CSMA-type of access on LoRa networks. In Table 2.1, we compare the primary protocols among the aforementioned in this section based on seven evaluation characteristics. As observed, the protocols utilizing time-division in accessing the channel are characterized by high complexity and increased overhead with regards to transmission scheduling. To evaluate the effectiveness of these protocols we look into the product of Packet Reception Ratio (PRR), i.e., the ratio of correctly received over transmitted packets, and normalized traffic, i.e., the part of time occupied by transmissions. Note that we utilize the pairs of PRR and traffic that give the highest product. As observed in Table 2.1, although most of these protocols can reach more than 80% of PRR [67, 75, 69, 70, 73, 74, 71, 72, 80, 81], they collapse as traffic increases (see

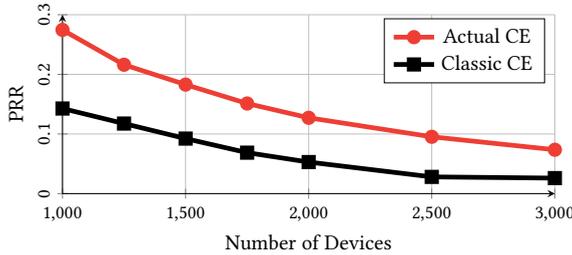


Figure 2.5: Packet Reception Ratio for an increasing number of nodes on different CE conditions - Actual CE (verified in-field) and classic CE (6 dBm difference). The devices are uniformly distributed in ring-bands around the gateway and transmit 20 B frames periodically at 868.1 MHz using SF10.

effectiveness-column). This motivates us to design MAC-protocols for LoRaWAN that can scale effectively.

## 2.3. CHARACTERIZATION OF CAPTURE EFFECT AND CHANNEL ACTIVITY DETECTION

To increase the capacity by accommodating higher traffic loads, decentralized solutions are needed at the MAC layer, incorporating observations from the physical layer (LoRa), and leveraging the phenomena that are manifested, i.e., Capture Effect (CE) and Channel Activity Detection (CAD). In this section, we present the insights acquired from the real-world evaluation of CE and CAD in LoRa-PHY. In terms of CE, the goal is to define strict rules according to which a frame is successfully received after being interfered with by one or more frames of the same or different RSSI that are received at the same time or slightly earlier/later. Therefore, we utilize one receiving LoRa-device and up to three LoRa-transmitters in different configurations of transmission. In terms of CAD, the aim is to define the success probability of CAD as a means of medium sensing under different scenarios of fading and range for every SF. Therefore, we utilize mobile transmitters and stationary receivers for every SF and different elevation levels in LoS and NLoS cases. The findings of this section are incorporated into the design of our MAC-layer algorithms (see Chapters 3, 4, and 5).

### 2.3.1. CAPTURE EFFECT

CE defines the successful reception of a frame against its adversaries either due to its relatively higher power and/or due to having initiated its reception slightly earlier [41, 42, 43]. CE is shown to be prevalent in LoRa networks by Bor *et al.* [92]. Bankov *et al.* designed a mathematical model of the LoRa transmission process that accounts for CE due to both power difference and delay offset, and thus infers the network capacity [93]. Further, Fernandez *et al.* showed that the probabilities of frame loss change even with slight differences of RSSI, necessitating the detailed approach towards the impact of capture effect in the capacity of LoRaWAN [94].

We perform extensive collision scenarios involving multiple LoRa-devices. Focusing on the receiver, by observing the reception results of interferers of different powers and



Figure 2.6: Setup for in-field evaluation of CE.

Parameter	Value
Transmitters Number	2, 3
Transmission Times per Scenario	30
Transmitters-Receiver Distance	100 m
Power Difference (abs values)	[0,6] dBm
Power Difference Increase Rate	0.25 dBm
Delay Offset	(0, 230) ms
Delay Offset Increase Rate	5 ms
Frame Size	20 B
Spreading Factors	7, 10, 12
Operating Frequency	868.1 MHz
Bandwidth	125 kHz
Coding Rate	4/5

Table 2.2: Parameters of in-field experiments of CE.

delay-offsets, we define probabilistic rules under which data-frames are correctly received. By employing such rules on the MAC, the IoT-devices can adapt their transmission power to manage successful transmissions to at least one receiving gateway while at the same time creating the least possible interference to other transmitters. Fig. 2.5 presents the potential of observing CE in-field and defining detailed probabilistic rules of frame reception, instead of following the classic approach that considers power capture due to 6 dBm higher power [95, 80]. As seen, by taking into account the in-field results of CE more devices are served in reality for the same values of PRR, e.g., 1000 more devices for PRR of 0.1.

In the works of Bor and Rahmadhani, the LoRa-frame structure was partitioned, and it was shown that one of two overlapping transmissions could be correctly received as long as critical symbols of its data-frame were not interfered [92, 34]. However, the above works are far from complete in studying the CE phenomenon in LoRa networks as they utilize only two transmitters and only specific differences in power (i.e., 1 dBm and 6 dBm). Further, they present results only for the high SFs (SF11 and SF12) which are the least utilized in Smart City scenarios [15]. In this work, we perform extensive collision experiments, evaluating a variety of differences in transmission power and delay offset with high granularity. We test for low, medium, and high SFs using 2 and 3 transmitting devices. The distance between transmitters and receiver are large enough to avoid antenna perturbations and the path loss for each device is identical. The specifics of our experiments are seen in Table 2.2. Fig. 2.6 presents a photo of an experimental setup with three interferers and one receiver.

### POWER DIFFERENCE

Fig. 2.7 presents the PRR achieved by each of the two devices transmitting simultaneously with different power and the ratio of corrupted frames. Device A (Tx A) transmits with 7 dBm while device B (Tx B) sweeps from 0 dBm to 14 dBm.

**Observation #1.** At SF7 and SF10, a power difference below the value of  $-1.25$  dBm and above  $2 - 2.5$  dBm can guarantee that the medium is captured by one of the devices, i.e.,  $PRR \geq 90\%$  in Fig. 2.7a and Fig. 2.7b. At SF12, the CE is more evident as transmission power difference increases. The mean value of PRRs of the dominating devices is 0.923 and the deviation 0.092, see Fig. 2.7c.

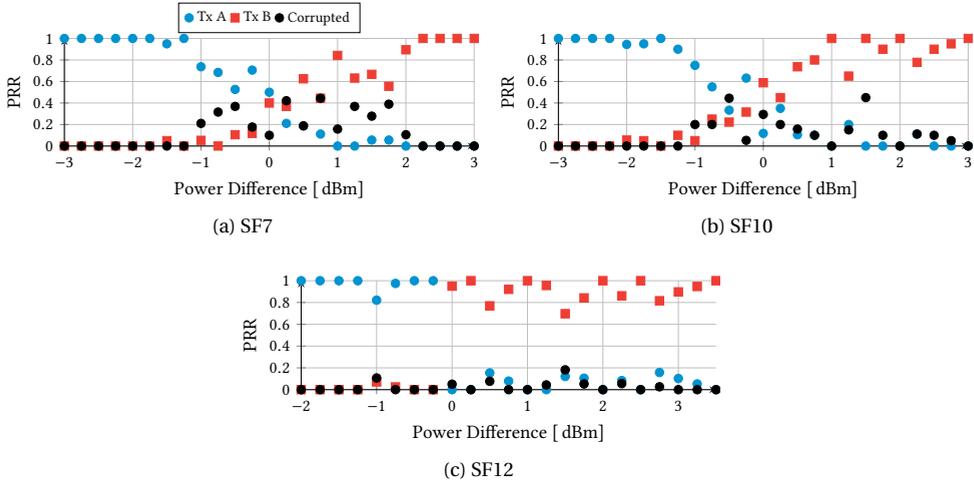


Figure 2.7: PRR based on power differences for different SFs – two transmitters.

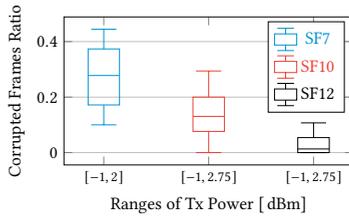


Figure 2.8: Distribution of corrupted frame ratios per SF.

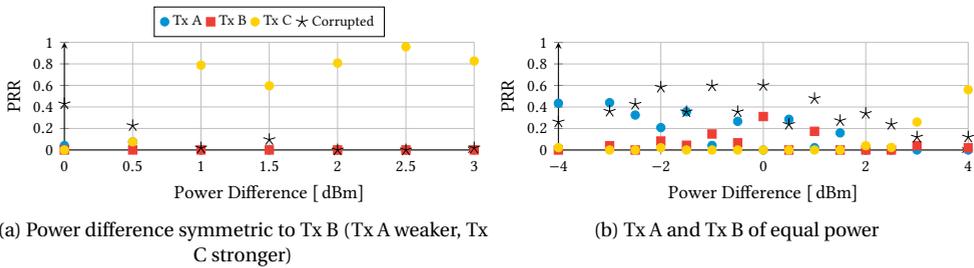


Figure 2.9: PRR based on power differences for SF7 – three transmitters.

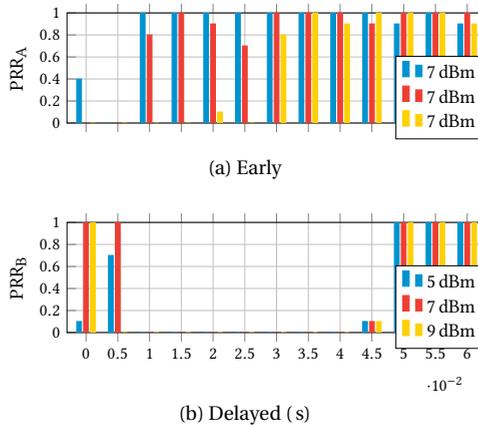


Figure 2.10: PRR based on delay offset at SF7 – two transmitters.

**Observation #2.** When both devices use high transmission powers, the dominating one needs higher power difference in absolute terms to establish a guaranteed CE. For example, in Fig. 2.7a when difference is -1.5 dBm (Tx A: 7 dBm, Tx B: 5.5 dBm),  $PRR_A = 0.95$ , while for +1.5 dBm (Tx A: 7 dBm, Tx B: 8.5 dBm),  $PRR_B = 0.65$ .

**Observation #3.** As seen in Fig. 2.8 the number of corrupted frames is increased at SF7, while almost absent at SF12. Since higher SFs utilize longer chirps to establish more robust transmissions they are less prone to corruption.

When three or more devices are involved, the chances of capturing the channel are lower. Nevertheless, in the vast majority of cases, the channel will be either captured by the strongest transmitter or all the frames will be lost.

**Observation #4.** Depending on the power difference between the two strongest signals a difference between 2-4 dBm can guarantee a PRR higher than 60% when three or more devices are involved, as seen in Fig. 2.9.

### DELAY OFFSET

The delay offset affects CE more critically than the power difference if the *early* device, i.e., the device transmitting first, manages to transmit at least the preamble of its frame without being interfered. In Fig. 2.10 and Fig. 2.11, the early device A transmits at stable periodicity, so the x-axes of Fig. 2.10a and Fig. 2.11a have no metric (since there is no delay to be shown). The delayed device B transmits with an offset from the transmission of device A that is stated in the x-axes of Fig. 2.10b and Fig. 2.11b. Each color corresponds to remaking the same experiment but with different transmission power per device. For example, as seen in Fig. 2.10a, device A participated in three different experiments, but in all of them it used 7 dBm to transmit. On the other hand, as seen in Fig. 2.10b, device B, which participated in the same three experiments, used 5 dBm, 7 dBm, and 9 dBm. Thus, we applied the same increase in delay offset to device B, but the power difference among the two devices was +2 dBm (experiment showed by the blue bars), 0 dBm (experiment showed by the red bars), and -2 dBm (experiment showed by the yellow bars). Note that each color corresponds to a different experiment here, and not each subfigure.

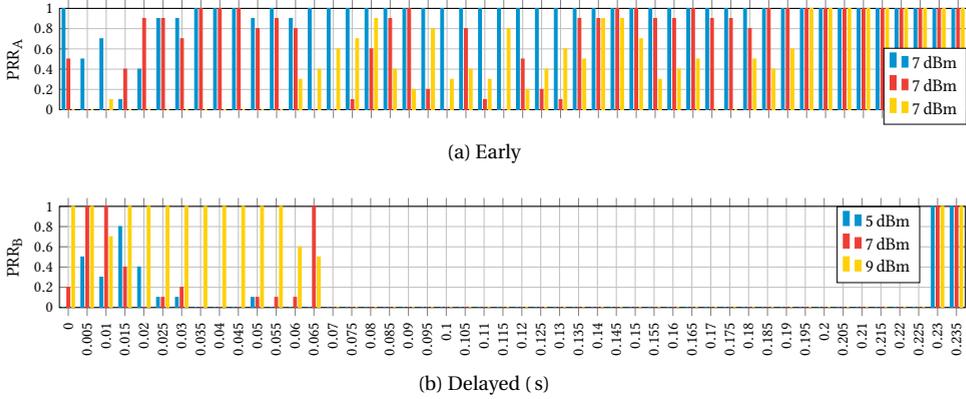


Figure 2.11: PRR based on delay offset at SF10 – two transmitters.

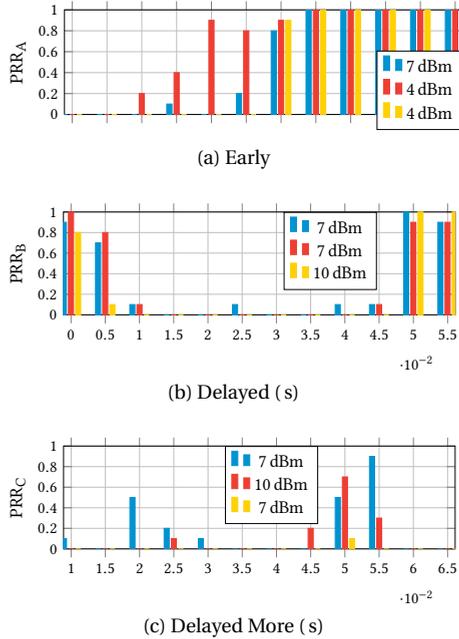


Figure 2.12: PRR based on delay offset at SF7 – three transmitters.

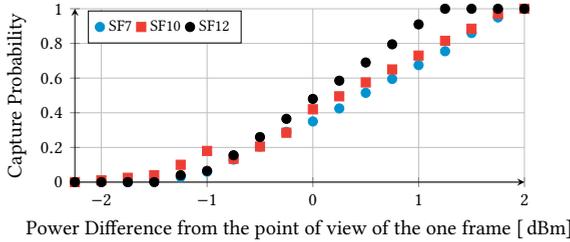


Figure 2.13: Probability of capturing the channel when two frames are received simultaneously with no delay offset.

**Observation #5.** As observed in both figures, during the preamble period the stronger transmitter has more chances to capture the medium, even if it transmits a few preamble-symbols later, e.g., yellow color in the first 65 ms of Fig. 2.11. However, if a device manages to transmit its whole preamble, then the interferer can only corrupt the ongoing transmission, rather than capturing the channel. If the delay is long enough, both devices transmit as the transmitted frames do not interfere, i.e., after 50 ms in SF7 and 230 ms in SF10.

**Observation #6.** When three or more transmitters are involved, as seen in Fig. 2.12, the capture effect is mostly determined by the interaction between the two earliest transmissions, i.e., between Fig. 2.12a and Fig. 2.12b. The rules of the previous observation decide the channel access outcome. When more than three transmissions overlap the results of PRR do not change considerably. Nevertheless, the existence of more interference corrupts further the frames of the early transmitter.

**Observation #7.** When more than half of the payload is transmitted without interference, the early node has even higher chances to capture the channel. To sum up, if most of the frame is received without interference the chances of CE for the early transmitter are increased.

In Fig. 2.13, we show the probabilities of a frame being received correctly depending on the RSSI-difference between the frame and the interfering frames that are transmitted simultaneously. These probabilities are derived from Fig. 2.7 and determine the chances of CE if a frame is interfered at its preamble-part (see Observation #5). If a frame is interfered at its payload part, the above probabilities are increased by using a weight that is determined by the portion of the frame that is already received (see Observation #7) as follows,

$$P_c = P_s(1 + r). \tag{2.1}$$

In Eq. 2.1,  $P_c$  is the actual probability to capture the channel,  $P_s$  is the probability to capture the channel in case of no delay offset (see Fig. 2.13), and  $r$  is the portion of the frame already received without interference.

### 2.3.2. CHANNEL ACTIVITY DETECTION

CAD is an optional functionality of LoRaWAN in which either the preamble or the payload of a frame can be detected while being transmitted by using cross-correlation with chirp-signals, in order to assess the idleness of the medium in an energy-efficient way and

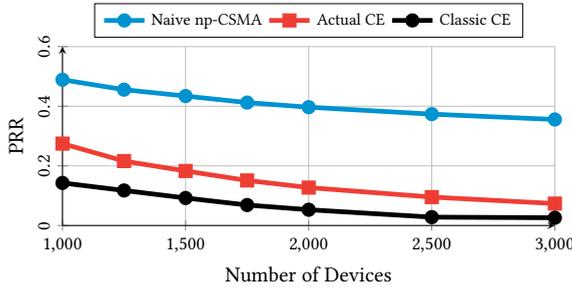
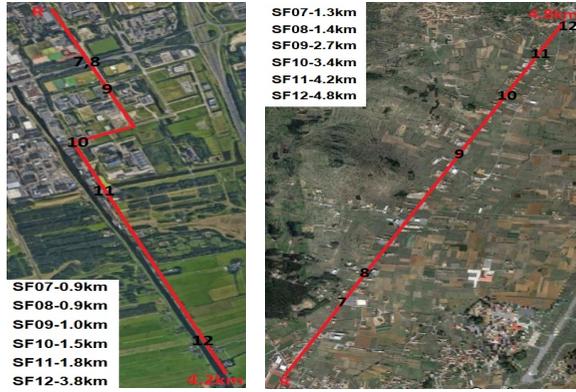


Figure 2.14: Packet Reception Ratio for an increasing number of nodes - Naive np-CSMA (BEB and Actual CE) Actual CE (verified in-field) and classic CE (6 dBm difference). The devices are uniformly distributed in ring-bands around the gateway and transmit 20 B frames periodically at 868.1 MHz using SF10.

without interfering when nodes duty-cycle [62, 61]. This functionality makes CAD a candidate-mechanism for the application of Carrier-Sensing Multiple-Access (CSMA) on the MAC layer of LoRaWAN. CSMA-based schemes are widely used as they have been proven to decrease collisions and increase the channel goodput [96, 30, 97]. As far as the CAD mechanism is concerned, although it is not a sophisticated Carrier-Sensing (CS) mechanism and was *not* designed for the CS as introduced by the literature [30], it is shown that it can be leveraged to implement CSMA in order to evade collisions and increase the network throughput in LoRaWAN, even without requiring any modification in the existing gateway and/or infrastructure [81]. Fig. 2.14 shows the potential of a non-persistent approach in claiming the medium. As observed, the addition of a naive Binary Exponential Backoff (BEB) mechanism [98, 99] on vanilla LoRaMAC along with accounting for the actual (in-field) CE, i.e., naive np-CSMA, increases, even more, the PRR achieved by serving a certain number of devices. For example, by applying naive np-CSMA the PRR increases from 2.6% to 35.6% for 3000 IoT-devices, compared to vanilla LoRaMAC. Despite this tremendous gain of 13.68 $\times$ , applying more sophisticated distributed algorithms will further improve the capacity of LoRaWAN.

### CHARACTERIZATION OF CAD

Since we plan to exploit CAD to reduce collisions, we must characterize the performance of CAD before employing it in our designs. Furthermore, older versions of LoRa-devices, SX1272/SX1276, could detect only frame-preambles, whose typical length is 8 symbols. We evaluate the improvement in performance between these versions and current SX1261 LoRa-devices which can sense any part of a LoRa-frame by performing in-field experiments. We set up in-field experiments in Line of Sight (LoS) and Non-LoS (NLoS) environments in the Netherlands (some in-band interference can occur) and Greece (without any interference from any other LoRa devices). SX1261 LoRa chipsets are used with six devices configured as stationary receivers on continuous CAD (SF7-SF12) and another device as a mobile transmitter. We place the receiving devices on different heights, from 2 m to 86 m to test the effects of blockage and fading on the signals. The different walking routes chosen in each scenario are depicted in Fig. 2.15. The points at which 10%-20% of CADs are successful are depicted, as maximum sensing ranges. The experimental setup is presented in Fig. 2.16 and the configuration specifics are reported in Table 2.3. The



(a) NLoS – Receiver at 22 m (b) LoS – Receiver at 2 m



(c) LoS – Receiver at 86 m

Figure 2.15: Maps indicating the route taken. The receivers were stationary (indicated by the point ‘R’ in the figures) and the transmitter was on foot. The SF-points on the map indicate maximum distance with at least 10-20% successful CAD ratio. The calculated ranges are straight-line distances.

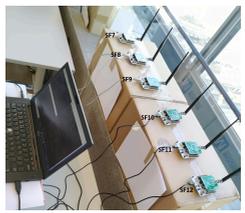


Figure 2.16: Setup of receivers for the in-field evaluation of CAD.

Parameter	Value
Spreading Factors	7-12
Measuring spots	per 50 m
Transmissions per spot	100 (SF7-SF11), 70 (SF12)
Frame Size	20 B
Transmit Power	14 dBm
CAD symbols	2 (SF7-8), 4 (SF9-12)
CAD duration	0.046 s (SF7) - 1.319 s (SF12)

Table 2.3: Parameters of the CAD in-field experiments.

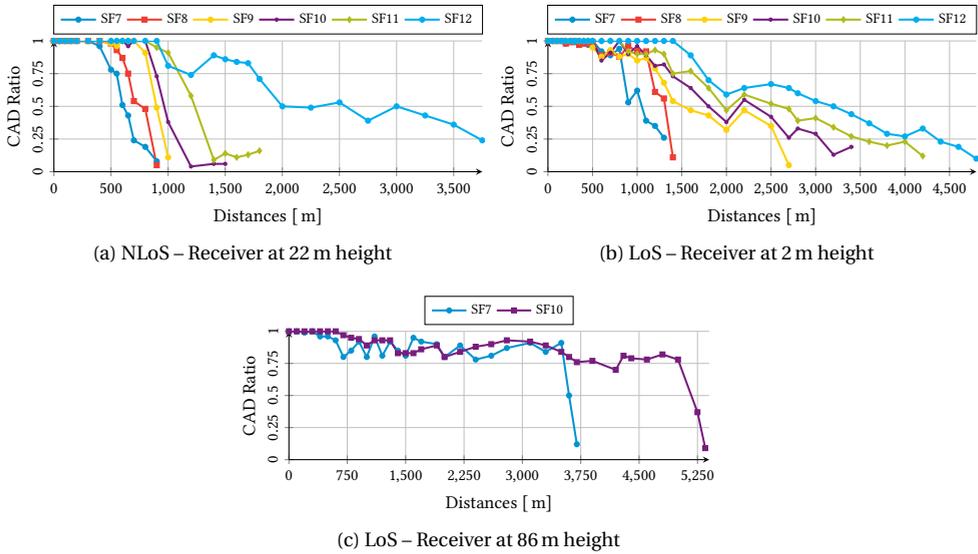


Figure 2.17: Ratio of successful CAD detections over distance.

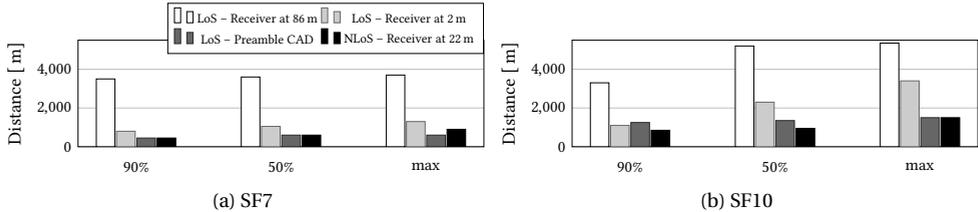


Figure 2.18: Achieved ranges based on different CAD success ratios.

values of operating frequency, code rate, and bandwidth remain the same as for the CE experiments.

**Observation #8.** The CAD performance is reduced as the distance increases between the transmitter and the receiver. In Fig. 2.17, wherein at low SFs the performance deteriorates abruptly, contrary to high SFs. At SF7 the difference between the distances where CAD is 90% and 25% is just 500 m, while the corresponding difference at SF10 is 2300 m.

**Observation #9.** NLoS affects the performance of CAD critically because of the multipath effect due to buildings, trees, and other blockages. For example, by comparing Fig. 2.17a and Fig. 2.17b, at SF10 the LoS case achieves 90% successful CAD at 1.3 times the distance of the NLoS case, even though the receiver at the LoS case is only at 2 m from the ground. In case lower levels of achievable CAD rates and higher SFs are considered, this improvement in the distance is higher, e.g., in SF11-LoS 50% successful CAD is achieved at 2× larger distance than in NLoS.

**Observation #10.** The height of the receiver affects CAD detection significantly, due to the clear LoS it offers. To evaluate this, we performed one more in-field experiment in the

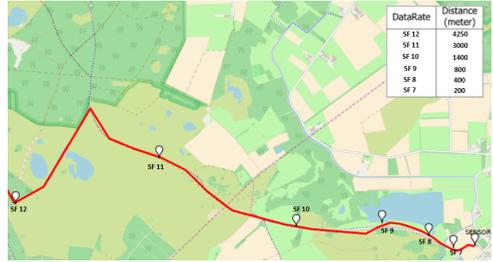


Figure 2.19: A map indicating the route taken (red line) and the sensing ranges for different SFs. The receivers were stationary (indicated by the point ‘SENSOR’ in the figure) and the transmitter was on a bicycle. The calculated ranges are the straight-line distances (not the distance the bicycle traversed).

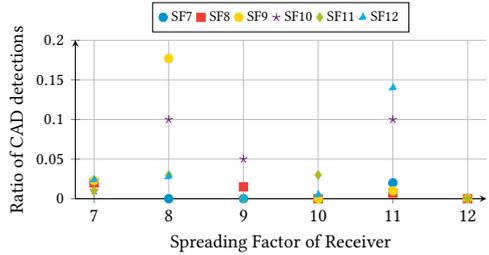


Figure 2.20: Percentage of false detections of CAD across SFs.

Netherlands only for SF7 and SF10 with the receiver positioned at 86 m from the ground, with the route taken seen in Fig. 2.15c. Elevating the receiver at 86 m boosts 1.3-3 times the operating distance, compared to the receiver being at 2 m from the ground.

**Observation #11.** In Fig. 2.18 we compare the achievable range for different LoS scenarios and receiver positions. Further, we add in the comparison the results of CAD experiments performed with SX1276 devices, which sense only for frame preambles. The achievable range of CAD, in this case, is seen in Fig. 2.19. In low SFs, the performance deteriorates abruptly, contrary to high SFs. Although applying CAD to the LoRa preamble symbols is less prone to frame-detection failures over its application to the payload symbols [89], the application of CAD to the whole frame (i.e., preamble and payload) obviously improves the ratio of CAD detections. This is observed in Fig. 2.18, wherein the performance of CAD in NLoS when the whole frame is sensed is similar to the performance of CAD in LoS when only preambles are sensed.

**Observation #12.** In Fig. 2.20, we evaluate experimentally the false detections of CAD using six transmitting and one receiving LoRa-devices with a LoS distance of 100 m between them. The receiver employed CAD at all six SFs, one-by-one. For each SF used by the receiver, each transmitter, configured always on the same SF (7 - 12), transmitted 100 frames of 20 B, i.e., 600 frames in total per transmitter. Although SFs are claimed to be orthogonal, our experiments revealed a non-trivial percentage of frame-detections at a different SF than the receiver’s configuration, as high as 19%. As seen in Fig. 2.20, this happens mostly between adjacent SFs, without excluding cases of high SFs falsely detected by low SF-CADs due to their long frame duration.

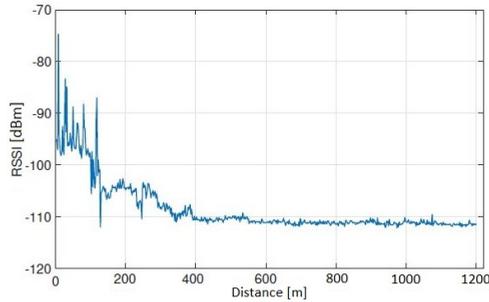


Figure 2.21: RSSI over distance from the receiver for SF9.

**Observation #13.** The performance also decreased when two or more transmitters were active. Referring again to Fig. 2.20, the accuracy of CAD detection for transmitter-receiver pairs of the same SF was at least 96%. We found that considering the results of three consecutive CADs compensated for such errors because when the majority over three CAD-outcomes was taken  $\approx 100\%$  of transmissions were detected correctly.

**Observation #14.** We found that the noise floor had an average RSSI value of -100 dBm. Fig. 2.21 shows the RSSI variation over distance for SF9. RSSI cannot be used to reliably sense the channel as LoRa frames can be successfully received even when the signal power is below the noise floor.

## 2.4. SUMMARY AND OUTLOOK

The observations drawn from the real-world study of CE and CAD helped in the thorough investigation of these phenomena. Among them, the following are critical for the development of our MAC-layer algorithms in Chapters 3, 4, and 5, and hence deserve more attention.

In terms of CE, when there are overlapping transmissions from several devices, CE depends on the two (rarely three) strongest and earliest transmissions. Additionally, a frame whose preamble is received successfully can be lost/corrupted by interference, but none of the interfering frames can capture the channel. Further, the number of corrupted frames is increased at low SFs, contrary to high SFs wherein frames are usually either received correctly or lost. These observations help us focus our attention only on the transmissions that are critical, and thus easily identify the devices that can "politely" reduce their transmission power for the benefit of the LoRaWAN. We employ them in *np*-CECADA (Chapter 4) and SFMAC (Chapter 5).

Regarding CAD, the success in detection is majorly affected by (N)LoS, multipath fading, and elevation of the receiver. Also, when multiple transmitters operate a considerable fraction (up to 20%) of CAD-detections is false in terms of SF and 4% of CAD-detections between transmitters and receivers of the same SF fail. This necessitates the application of two extra consecutive CAD-attempts per frame whenever the first CAD-attempt shows the channel occupied (CAD duration is only a small fraction of frame duration). Then, the majority rule over the results of these three CAD-attempts reveals if the channel is likely occupied by a frame of the same, non-orthogonal SF. Furthermore, utilizing high SFs

allows CAD to take place successfully at higher distances, revealing more hidden devices, e.g., the maximum distance for successful CADs at SF12 is 5 times higher than at SF7 in Fig. 2.17b. This specific attribute is leveraged especially in SFMAC (Chapter 5), wherein high SFs are dedicated to channel sensing in order to evade simultaneous transmissions among hidden devices at low SFs.



# 3

## A PERSISTENT DISTRIBUTED MAC PROTOCOL FOR LoRaWAN

CARRIER Sense Multiple Access (CSMA) based techniques have been heavily investigated in the literature since it was first introduced by Kleinrock and Tobagi [30]. They introduced three variants of CSMA: (a) 1-persistent, (b) non-persistent, and (c)  $p$ -persistent. The first two variants were already explained in Chapter 1. In the  $p$ -persistent case, a device continuously senses the medium until it becomes idle and then transmits with probability  $p$ . The inherent assumption in these protocols was that devices are in the sensing range of each other. They relaxed this assumption in their subsequent work and tackled the hidden-terminal problems [31]. The authors proved the degradation of CSMA's performance and found throughput-bounds for 1-persistent CSMA.

We focus on increasing the number of IoT-devices served by a single gateway in a LoRaWAN from hundreds to thousands. As frames can take a long time to transmit (several ms for larger spreading factors) CAD will introduce false negatives as no preamble will be detected during the transmission of the payload. To remedy this -phrasing it politely-imperfection, we combine CAD with the principles of persistent-CSMA ( $p$ -CSMA) [30] as it can be made to operate within the constraints listed above. We design a protocol, called  $p$ -persistent Channel Activity Recognition Multiple Access ( $p$ -CARMA) that tries to evade collisions with (most) neighboring devices using medium-sensing and probabilistically minimizes collisions due to hidden terminals using a localized and adaptive persistence  $p$ -value algorithm. Reducing the probability of collisions is a pragmatic approach. The "aggressiveness" of 1-CSMA cannot be tuned, leading to collisions as traffic increases. On the other hand,  $np$ -CSMA can create large delays in transmitting messages, leading to a lack of freshness of sensor data. Thus, we adopt  $p$ -CSMA, whose  $p$ -persistence in transmitting can be tuned by choosing the right value of  $p$ .

**Notice 1:** In this chapter the results of the Capture Effect showed in subsection 2.3.1 are not accounted for. Instead, the classical approach is followed, where a signal captures the channel only if it is received with RSSI 6 dBm higher than the interference. Therefore, the performance results of this chapter show the worst-case scenario taking place.

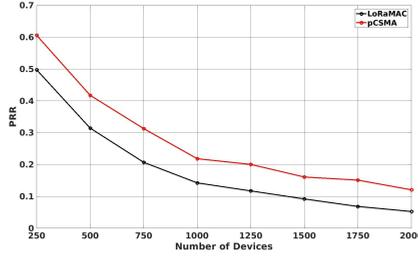


Figure 3.1: Packet Reception Ratio for increasing number of nodes for different MAC protocols - LoRaMAC (Aloha-like) and  $p$ -CSMA based with  $p = \frac{1}{N}$ . The devices transmit frames of 20 B periodically at 868.1 MHz using SF10.

	LoRaMAC	$p$ -CSMA	Static $p$ -CARMA	adaptive $p$ -CARMA
Low consumption	✓	✗	✓✓	✓✓
Low overhead	✓✓	✗	✓	✓
Low complexity	✓✓	✓	✓	✓
High PRR (effectiveness)	✗	✓✓✓	✓	✓✓
High channel utilization	✗	✓✓✓	✓	✓✓

Table 3.1: Comparison of Carrier Sensing approaches.

**Notice 2:** For this chapter, we have utilized the LoRa SX1276 devices, which are able of detecting only frame-preambles. Therefore, the sensing capability of  $p$ -CARMA is restricted only in preambles.

Fig. 3.1 shows the potential of being polite, yet persistent. Using the open-source code of [32, 100] for  $ns$ -3 [44], we simulated according to the LoRAWAN specifications the Packet Reception Ratio (PRR)<sup>1</sup> of standard LoRaMAC and a naive CAD-based version of  $p$ -CSMA for an increasing number of devices. Without loss of generality, the value of  $p$ , determining the probability that a device transmits once the channel is found clear, is set to  $1/N$ , where  $N$  equals the number of devices. That probability is optimal when all devices are within range of each other and carrier sensing is working perfectly, which is not the case in a star topology LoRAWAN where CAD only looks for preambles instead of any activity on the channel. Nevertheless,  $p$ -CSMA (red curve) increases the number of devices that can be effectively supported at a desired PRR with quite a margin compared to standard LoRaMAC (black curve). For example, for a PRR of 0.3, the number of devices scales from 500 to 750. Despite this 50% gain, there is a lot of room for improvement as the corresponding channel utilization is only 9.44%.

We aim to increase the scalability of LoRAWAN, focusing on per gateway scaling, and specifically by raising PRR when the number of devices increases while being energy efficient. This is achieved by reducing collisions using  $p$ -CSMA principles in a network with a large number of hidden terminals with minimum feedback. Our approach involves devices *indirectly* estimating the  $p$ -value by sensing the channel using CAD, and learning about the collective channel occupancy. Based on this, each device independently selects a persistence value ( $p$ ) deciding when to transmit in order to increase the chance of

<sup>1</sup>PRR is the ratio of received over transmitted frames

successful transmission. Table 3.1 sums up the comparison of our adaptive *p*-CARMA protocol to classical *p*-CSMA, naive CAD, and LoRaMAC, showing that *p*-CARMA is the preferred protocol to increase scalability under the constraints given by LoRa networks. Notice that vanilla LoRaMAC consumes more energy since the nodes simply transmit frames. This requires more energy than sensing (CAD) and dropping a frame.

Our contributions are the following:

1. We design *p*-CARMA, an adaptive, distributed *p*-value CS MAC protocol for LoRaWAN.
  - (a) *p*-CARMA uses the CAD mechanism to increase the number of devices by at least twofold as compared to LoRaMAC.
  - (b) In *p*-CARMA the gateway provides extra (secondary) assistance to devices in adapting *p*-value by periodically reporting its observations regarding the delay of packet deliveries.
  - (c) This protocol can be used on existing LoRaWAN deployments without requiring any changes in the gateway/infrastructure.
2. We created several modules for simulations of LoRaWAN with CARMA capabilities in *ns-3* in order to simulate several scenarios involving thousands of devices including energy measurements. The CAD module employs the results found in our real-world experiments for enabling closer-to-reality simulations.
3. Taking into account the information acquired from our real-world experiments, we tune our simulation parameters for different SFs and evaluate the performance of our proposed *p*-CARMA against the classic LoRaWAN and also deterministic  $p = 1/N$ -CARMA taking into account parameters such as the number of devices, PRR, and energy consumption.
4. Through our extensive simulations, we provide key insights on *p*-CARMA's performance. We have also developed two metrics that must be used for the evaluation of CARMA-based protocols in LoRaWAN.
5. We applied *p*-CARMA on our LoRa testbed with 30 devices, increasing their duty-cycle of transmission to mimic large-scale LoRaWAN deployments. The in-field evaluation showed 25 to 27 out of 30 devices achieving higher PRR when using *p*-CARMA in SF7 than when vanilla LoRaWAN is used.

### 3.1. *p*-CARMA

*p*-CARMA is less aggressive to transmit packets than 1-persistent CSMA and more aggressive than non-persistent CSMA. *p*-CARMA cannot eliminate collisions due to hidden terminals, because solutions such as Request-To-Send/Clear-To-Send (RTS/CTS) that require two-way communication between a LoRa device and the gateway are not possible in LoRa. Despite this, *p*-CARMA can achieve a higher PRR than LoRaMAC as it attempts and often succeeds in choosing the 'right' moment to transmit. *p*-CARMA reduces the possibility of collisions with devices within its sensing range through CAD; and more collisions would be reduced by deferring transmissions due to the choice of '*p*'. Before we proceed to the algorithm, let us note the objectives of our design. At first, the distinctive characteristics of the current LoRaMAC must be respected.

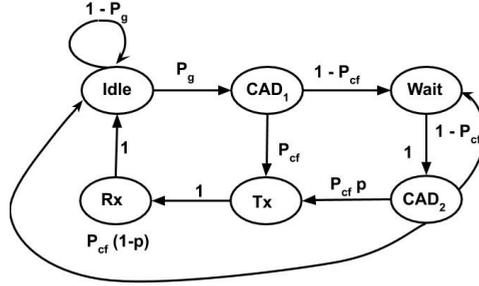


Figure 3.2: Markov model of  $p$ -CARMA protocol for a single device.

- The protocol should be distributed, i.e., each device decides autonomously when to transmit in an unslotted manner. The gateway may only assist in deciding but never dictate transmission times.
- The MAC must be of low complexity since LoRa-devices are of low computational capability.
- The changes to the current frame structure and *modus operandi* of LoRaWAN should be minimum [54, 58].

Along with the above, we add the following design goals to increase the performance of LoRaWAN.

- **Increase capacity.** Capacity refers to throughput per coverage area. By increasing channel utilization for correctly received frames, throughput is increased as higher traffic loads can be accommodated.
- **Increase scalability.** Scalability refers to collision avoidance which improves PRR and thus increases the effectiveness of communication. In other words, under the same value of PRR, more devices can be served, allowing for a dynamic LoRaWAN in which multitudes of devices can be added/removed without compromising the quality of service.
- **Minimize energy overheads.** The functions of the algorithms must consume low amounts of energy as they are introduced to energy-constrained devices.

### 3.1.1. WORKING OF $p$ -CARMA

The characterization of CAD performance reveals that it is far from ideal to assess reliably if the medium is idle or busy. The problem is aggravated due to two facts: (a) false negatives creating a lot more hidden terminals and (b) no unicast feedback per packet from the gateway, which eliminates the adoption of a solution that solves hidden terminals such as RTS/CTS based mechanisms. Reducing the probability of collisions is a pragmatic approach in this scenario. In this chapter, we adopt  $p$ -CSMA, whose aggressiveness and collisions can be tuned by choosing the right value of ' $p$ '. Before we proceed to outline our distributed solution to set  $p$ , we describe the working of  $p$ -CARMA.

**Algorithm 1:** *p*-CARMA

---

```

/* This function is called when a packet is generated */
Result: Transmit or drop the generated frame
1 Perform CAD1
2 if channel found free then
3   Transmit the frame
4 else
5   /* Sensed a preamble. Be polite to begin transmission after
6     the current one. */
7   pkt_tx_end_time =  $\tau$  + ToA
8   /*  $\tau$  is the current time */
9   backoff_time = min( $\tau$  + ToA * rand(), pkt_tx_end_time)
10  sleep (backoff_time)
11  Perform CAD2
12  if channel found occupied then
13    /* Sensed new preamble within on-going transmission. */
14    go to line 5
15  else
16    if  $\tau < \text{pkt\_tx\_end\_time}$  then
17      /* No new transmission detected. */
18      go to line 6
19    else
20      /* No new transmission detected until the end of
21        on-going transmission. */
22      Transmit the frame with persistence p
23    end
24  end
25 end

```

---

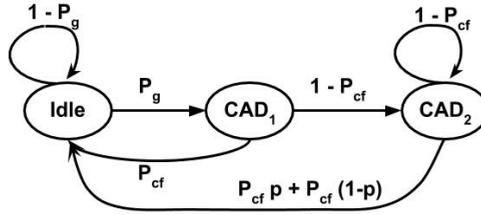


Figure 3.3: Simplified Markov model of  $p$ -CARMA protocol for a single device.

3

$p$ -CARMA is a distributed, unslotted, and low-complexity carrier-sensing based protocol. Every device performs the set of actions under specific probabilities, as denoted in the Markov model of the state transition diagram in Fig. 3.2.

A LoRa device begins its operation in the Idle state. When a new packet is generated and is ready for transmission with probability  $P_g$ , the device will perform CAD to assess the channel state (noted as  $CAD_1$ ). If the channel is found free (with probability  $P_{cf}$ ), the device will proceed to transmission (Tx). As we work with the *class A* devices of LoRaWAN, the node can open up two receive windows (Rx). Finally, it will return to the Idle state.

However, if the channel is occupied (i.e., probability  $1 - P_{cf}$ ), the device proceeds to the state ‘Wait’. In the Wait state, a random duration between 0 s and its payloads’ average Time-on-Air (ToA) is picked, and then the device performs CAD (noted as  $CAD_2$ ). As the device does not continuously perform CAD but waits (or sleeps) in between, the energy consumption for CAD reduces.

As CAD only detects preambles, we would like to sense if there is any other transmission that begins within the ongoing transmission, which is the reason for choosing a backoff value between  $[0, ToA]$ . If the channel is still found to be occupied, an overlapping transmission is detected. The device gets back to waiting randomly before probing the channel. This repeats until the channel is found free. Upon finding the channel free, in order to be polite, the device will wait until the end of any ongoing transmission. At the end of any such transmission, the device will either transmit (Tx) with probability  $p$  or will refrain from transmitting with probability  $1 - p$ . In the figure, there is no packet queue/buffer considered. Therefore, the frame is dropped with probability  $1 - p$ , and the device returns to the Idle state. This is shown in Algorithm 1. In case a buffer is considered, the device returns to the Wait state with a probability of  $1 - p$ .

### 3.1.2. ADAPTIVE $p$ -VALUE ALGORITHM

When all devices are in the sensing range of each other it is easy to set  $p$ -values. In a network of  $N$  devices, the optimal value for  $p$ , assuming that the payload sizes are equal and have one frame to send, can be derived using Binomial distribution to  $\frac{1}{N}$ . This would, however, work optimally only when all the nodes are in the sensing ranges of each other; otherwise, it leads to collisions and hence achieves only a sub-optimal result. Furthermore, as the  $p$ -value is not dependent only on ‘N’, when devices transmit under different periodicities sub-optimality is the result.

However, in LoRaWAN, not all devices are in sensing range or have the same number of neighbors and CAD is not perfect. As the value of  $p$  is critical to the performance of

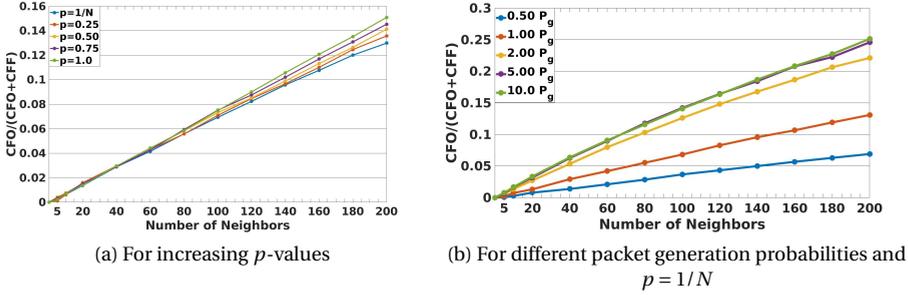


Figure 3.4: Average ratio of number of times the channel sensed as occupied to number of devices in the range of each other, i.e., neighbors. SF10 and  $P_g = 1.25\%$ .

$p$ -CARMA, this value has to be chosen carefully. The Markov model presented in Fig. 3.2 can be simplified to the chain shown in Fig. 3.3, where the transitions of probability 1 have been removed. This chain is evidently positive, recurrent and irreducible with transition matrix as follows:

$$\begin{array}{l}
 \text{Idle} \\
 \text{CAD}_1 \\
 \text{CAD}_2
 \end{array}
 \begin{array}{l}
 \text{Idle} \\
 \text{CAD}_1 \\
 \text{CAD}_2
 \end{array}
 \begin{array}{l}
 \text{CAD}_1 \\
 \text{CAD}_2
 \end{array}
 \begin{array}{l}
 \text{CAD}_2
 \end{array}
 \begin{array}{l}
 \left[ \begin{array}{cccc}
 1 - P_g & P_g & 0 & \\
 P_{cf} & 0 & 1 - P_{cf} & \\
 P_{cf} & 0 & 1 - P_{cf} & 
 \end{array} \right]
 \end{array}$$

Deriving the steady state distribution will lead to the following equivalent equations:

$$\text{CAD}_1 + \text{CAD}_2 = \frac{P_g}{P_g + P_{cf}} \quad \text{or} \quad \text{Idle} = \frac{P_{cf}}{P_g + P_{cf}} \tag{3.1}$$

The device being in sensing state, either  $\text{CAD}_1$  or  $\text{CAD}_2$ , or in Idle state is determined by unseemingly interdependent variables, *i.e.*, (i) the probability of generating a new packet  $P_g$ , (ii) the probability of finding the channel free  $P_{cf}$ , or occupied  $(1 - P_{cf})$  and (iii) persistence  $p$ . Usually,  $P_g$  is preset by users/operators. The number of devices (or traffic) in the network and the changes in the  $p$ -values of devices affect only  $P_{cf}$ . The  $p$ -values of devices can be viewed as knobs to be less or more ‘aggressive’ in transmissions, which affects the channel traffic and hence  $P_{cf}$ . This, according to Eq. (3.1), leads the device in changing the time it spends in ‘sensing’ state ( $\text{CAD}_1 + \text{CAD}_2$ ). In turn, any change in the time spent for CAD indicates that there are more or fewer devices transmitting (increased or decreased traffic) in one device’s vicinity, leading it to update its  $p$ -value. By the above, it is obvious that we are led to a circular argument, as the network dynamics, *i.e.*, transmission probabilities, hidden sectors of devices, dynamically changing traffic, do not allow an optimal solution regarding the  $p$ -values.

Furthermore, this model does not capture the dependency of other nodes in the network. The Markov model of Fig. 3.3 needs to be generalized into a model involving  $N$  devices. This would allow the analytical procedure of deriving the  $p$ -value per device. However, such an analytical model cannot be derived due to the complexity as the number of possible states increases exponentially as  $\mathcal{O}(3^N)$  with the number of devices  $N$ . Thus, we are forced to determine the  $p$ -value adaptively by using a heuristic approach.

### HEURISTIC APPROACH

To design a well-performing heuristic with the goal of improving PRR, we try to first understand the dependencies involved. From Eq. 3.1, the probability that each device senses the channel as free or occupied affects its sensing period, and as a result how the  $p$ -value is adapted. To adopt this observation in our heuristic, we focused on the number of times the device finds the channel free or occupied,  $CFF$  (Channel Found Free) and  $CFO$  (Channel Found Occupied), which are counted only at the first attempt of CAD for each frame, i.e., state transitions ‘ $CAD_1 \rightarrow Tx$ ’ and ‘ $CAD_1 \rightarrow Wait$ ’ of Fig. 3.2. We evaluated  $(CFO/(CFO + CFF))$  for increasing number of non-hidden devices in their vicinity (called ‘neighbors’). The evaluation took place for  $p$ -values ranging from  $1/N$  to 1.0 (cf. Fig. 3.4a) and for five different packet generation probabilities (cf. Fig. 3.4b).

**Observation #1.** As  $p$ -values increase, devices transmit more often, increasing the probability of finding the channel occupied and the term  $(CFO/(CFO + CFF))$ .

**Observation #2.** An increase in  $P_g$  leads to higher traffic and a higher chance of finding the channel occupied. Specifically, in Fig. 3.4b increasing  $P_g$  by a factor of 2 corresponds to increasing the number of devices by a factor of 2 for the same  $P_g$ . However, for every  $p$ -value there is a threshold above which increasing the probability of generating packets does not affect  $CFO/(CFO + CFF)$  even without adaptive  $p$ . For  $p = 1/N$  this plateau is  $P_g \approx 6.5\%$ . The main reason behind this counter-intuitive behavior is that the device senses only the preambles and not the payloads; the payload part is sensed as a channel free of transmissions. This also explains the low ratio seen in the figure, even when the channel is close to saturation (at least one frame in the air at any instant).

**Observation #3.** When the number of devices in the sensing range of each other (neighbors) increases, the ratio of sensing the channel as occupied increases almost linearly.

To apply the above observations in our heuristic, we use the complementary term  $(CFF/(CFO + CFF))$ , representing the probability to find the channel free on the first attempt based on observations. The equation of the adaptive  $p$ -CARMA is the following:

$$p = \left(1 - CDR\right) \left(\frac{\bar{D} - D_{min}}{D_{max} - D_{min}}\right) \left(\frac{CFF}{CFO + CFF}\right), \quad (3.2)$$

with  $1 \geq p \geq 1/N$ .

Since the term  $(CFF/(CFO + CFF))$  is explained above, we proceed to explain  $CDR$ ,  $\bar{D}$ ,  $D_{min}$ , and  $D_{max}$ .

### COLLISION DELAY RATIO (CDR)

The  $CDR_i$  of a device  $i$  is the ratio between the total delay of its frames that were transmitted and collided,  $d_i^C$ , over the total delay of all three possible conditions for its frames:

- Average delay between generation and transmission of successfully received frames,  $d_i^S$  (computed by the gateway).
- Estimated average delay seen by collided frames,  $d_i^C$  (computed by the gateway).
- Average delay of the frames that were discarded due to persistence  $p$ ,  $d_i^D$ , (computed by the devices).

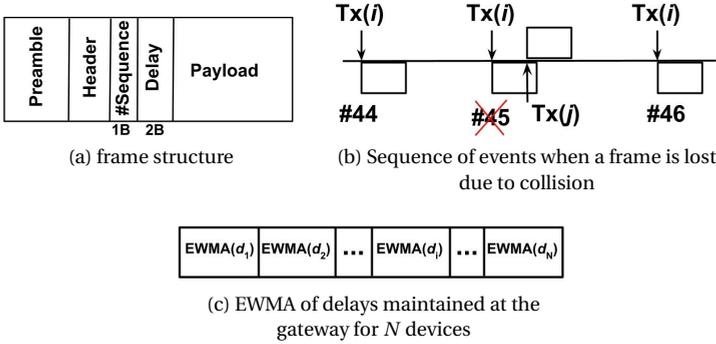


Figure 3.5: Frame Structure and delay estimation of collided frames.

In Fig. 3.5a, we show the simple frame structure, where we use a sequence number field and delay field, which help in finding  $CDR_i$ . To find the delay of successfully received frame  $d_i^S$ , the gateway simply uses the 2 B delay field in the frame. Just before transmission, the device adds the time spent by the packet from its generation to the current time. It is a bit tricky to find the time (delay) the packet spent at the device before colliding at the gateway. Since we get no information from the collided frames, we resort to estimating the (possible) delays of the collided frames. For this, we keep a buffer for every device  $i$  that contains the exponential weighted moving average (EWMA) of the previously successfully received frame as shown in Fig. 3.5c. Then using the sequence number, collided frames are found after receiving a frame successfully. Note, we assume transmitted frames are lost only due to collisions as LoRa is proven to be quite robust to channel conditions. Giving higher weightage to the delay experienced by the latest successful frame, the delay of the collided frames is calculated by taking the average of  $EWMA(d_i)$  and the  $d_i$  of the latest successful frame. Lastly, the devices can easily keep track of the delays experienced by the discarded frames. Consider the example in Fig. 3.5b, where frame 45 is lost due to collision. The gateway computes EWMA after frame 46 is received with its delays. Now CDR is defined by,

$$CDR_i = \frac{\sum_{k=1}^c d_{i,k}^C}{\sum_{k=1}^s d_{i,k}^S + \sum_{k=1}^r d_{i,k}^D + \sum_{k=1}^c d_{i,k}^C}, \quad (3.3)$$

with  $c$ ,  $s$ , and  $r$  being the numbers of collided, successfully received, and discarded frames. This could also be done just by counting number of frames collided, successful and discarded. However, using the delays can give a better picture of the dynamic nature of the events than just counting. Any increase in  $CDR_i$  denotes that less frames are received by the gateway from a device  $i$ .  $CDR$  is updated in stable periods, called ‘observing period’. The observing period is the timespan between two ADR (adaptive data-rate) messages from the gateway to the device  $i$ . During this period, the gateway gathers the results of each device  $i$  regarding the total delays of its successful and its collided frames,  $d_i^S$  and

Transmission Channel	868.1 MHz
Code Rate	4/5
Transmit Power	14 dBm
Bandwidth	125 kHz
Propagation loss model (mean, std)	Log-distance (0.56, 7.11)
Path loss exponent	3.76

Table 3.2: LoRaWAN configuration used for simulations.

3

$d_i^C$ , and is decided by the users/system. In addition, each device is aware of the total delay for its discarded frames,  $d_i^D$ . At the end of each observing period, the gateway clusters the delay-data of devices,  $d_i^S$  and  $d_i^C$ , in three groups, (i) low, (ii) medium, and (iii) high delays [101]. The centroid-values  $C_{ds}$  and  $C_{dc}$  of these three groups are communicated to each device. For example, if a device  $i$  belongs to the low group for the successful case and the high group for collided case, then gateway informs  $i$  regarding the values of the centroids  $C_{ds}^L$  and  $C_{dc}^H$ , respectively. All values are piggybacked on ADR messages in the downlink.

Now, since every device  $i$  knows its  $d_i^D$ , the individual  $CDR_i$  is computed and used in Eq. (3.2) throughout the next observing period in order for every device to adapt its  $p$ -value. During the next observing period, the gateway gathers again the delays (and in turn the centroids) to be used subsequently.

#### PARAMETERS $\bar{D}$ , $D_{min}$ , AND $D_{max}$

These parameters are correspondingly the average, the minimum, and the maximum delay that packets experience in a device from the moment they are generated to the moment they are transmitted or discarded, i.e., states ‘CAD<sub>1</sub>’, ‘CAD<sub>2</sub>’, and ‘Wait’ of Fig. 3.3. Intuitively,  $\bar{D}$  of a device is proportional to the number of devices (or traffic) in its sensing range. A value of  $\bar{D}$  close to the  $D_{max}$  denotes that the traffic on the channel that the device can sense is high (given that many preambles are being sensed despite the considerable false negatives of CAD), i.e., the device spends most time between the states ‘CAD<sub>2</sub>’ and ‘Wait’. This implies that the  $p$ -value must be low. On the contrary, if  $\bar{D}$  is close to the  $D_{min}$ , then the  $p$ -value must be high.

By bounding  $p$  to  $1/N$ , the device would at least have its ‘fair’ chance of medium access. Clearly, using Eq. 3.2 can adapt to the traffic around the device. The total number of devices  $N$  can be piggybacked by the gateway onto a downlink message. Note that to accommodate LoRa devices to join and leave the ambit of a gateway over time (usually days/months), we can recalculate the above parameters over a sliding window, which is a simple extension.

#### NETWORK BOOTSTRAPPING

During the network bootstrapping phase, each device starts with a pre-defined  $p$ -value, which can be in the range  $[1/N, 1]$  (the value of  $p$  converges regardless of the starting value, which is shown in Section 3.3). The statistics for  $\bar{D}$ ,  $D_{min}$ , and  $D_{max}$  are gathered over the first three transmissions. Then, all the parameters are updated every time a device manages to transmit a frame or refrains from the transmission. Furthermore, from

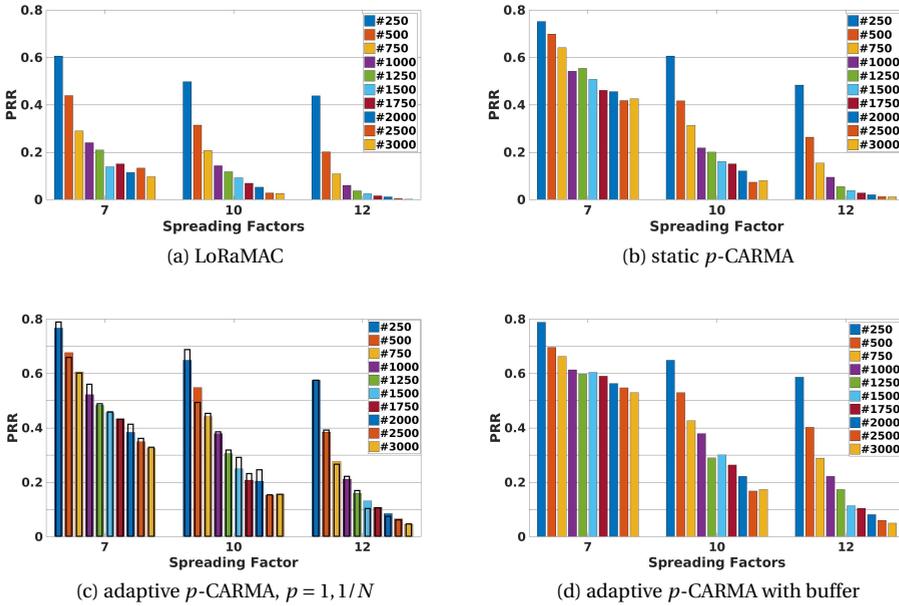


Figure 3.6: Average PRR for increasing number of devices.

the bootstrapping and until the end of the first observing period,  $CDR_i = 0$  for all devices. Thus, the devices still adapt based on local observations.

### 3.2. SIMULATION PARAMETERS AND METRICS

Due to the scale of the number of devices and also for the sake of repeatability for comparison, we had to undertake simulations for evaluating the performance of  $p$ -CARMA. Further, since EU guidelines strictly prohibit more than 1% duty cycle for LoRa devices, it is difficult to modify the behavior of devices to send frames at higher rates to study our proposed changes. We considered the open-source code of [32, 100] for  $ns-3$  to simulate according to the LoRaWAN specifications. To this code-base, we have developed CAD and extensible  $p$ -CARMA modules. Particularly, the CAD modules incorporate our findings from the field experiments of CAD performed with LoRa SX1276 devices, which are able to detect only frame-preambles (see Section 2.3.2). We consider a scenario involving one gateway and all the LoRaWAN devices distributed uniformly around it. All the devices are using one transmission channel. To study the scalability, we simulate the above scenarios by increasing the number of devices in steps. The details of the parameters and metrics used are given below.

#### SIMULATION PARAMETERS

We consider from 250 up to 3000 devices, with a step of 250, deployed around the gateway. Every device transmits periodically, with periodicity randomly chosen between the 1% duty-cycle limit and 3600 s. In each period, a device generates a payload of 20 B includ-

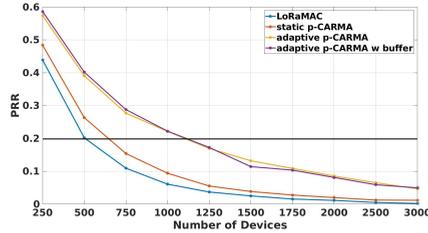


Figure 3.7: Average PRR for an increasing number of devices in SF12.

ing 1 B sequence number, and 2 B Delay (plus 8 B header). The devices are assigned a spreading factor (SF) to transmit on, using Adaptive Data Rate (ADR) messages from the gateway use SF12 and the closest devices use SF7. The other LoRaWAN settings considered are listed in Table 3.2. The energy values for CAD and devices are based on our measurements of SX1276 LoRa radios. These values are higher when the ToA of transmissions increases, i.e., for high SFs and large payload sizes. The simulations were run for all the scenarios 30 times, in order to get more than 95% confidence levels.

## METRICS

We evaluated our simulations using the following metrics.

**Packet Reception Ratio (PRR).** The PRR is the ratio between the total number of frames successfully received at the gateway and the total number of frames transmitted. PRR evaluates the success of the MAC protocols despite collisions. MAC protocols are compared over the number of devices they serve for the same PRR-values, indicating how they scale. Furthermore, PRR indirectly evaluates energy efficiency as it denotes the ratio of received frames, for which the energy of transmission was not wasted.

**Packet Transmittance Ratio (PTR).** The PTR of a device is the ratio between frames transmitted and the number of packets generated by the device. The number of packets generated is equal to the number of frames transmitted if the device employs LoRaMAC as is. PTR is used to evaluate how much  $p$ -CARMA refrains from transmitting in order to reduce collisions.

**Received over Generated packets (RoG).** RoG is the product of PRR and PTR, i.e., it is the ratio between the total number of frames successfully received at the gateway and the total number of packets generated. RoG presents the actual portion of the generated packets that reached the gateway.

**Energy consumption.** Energy consumption is monitored as an indication of the operational longevity of a device and to be compared with LoRaMAC. The total consumed energy of a device is the sum of the energy consumed for performing CAD operations, transmitting a frame, and being in receiving mode.

**Channel utilization.** The time spent for frames that are transmitted and successfully received at the gateway is an indicator of the efficiency in the usage of the channel.

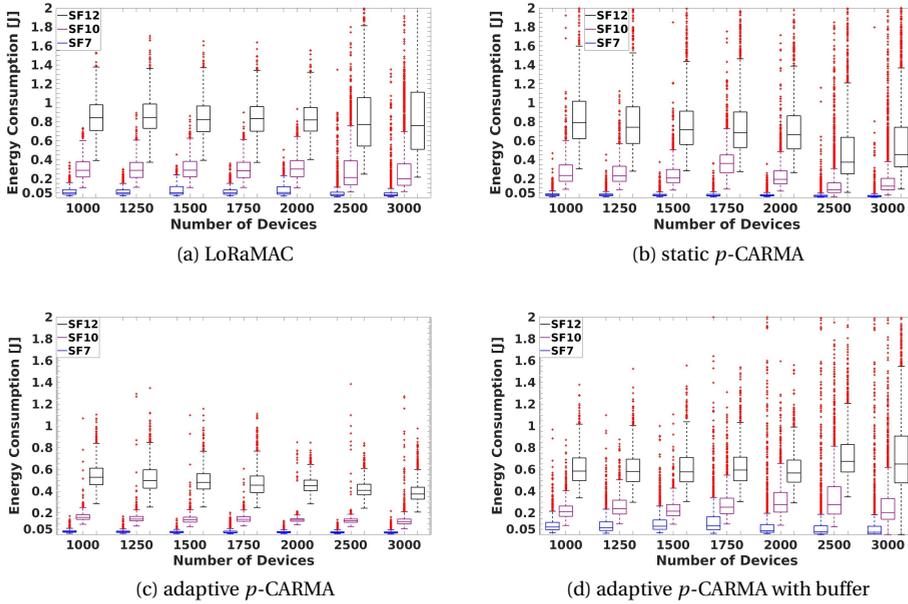


Figure 3.8: Total energy spent for an increasing number of devices.

### 3.3. PERFORMANCE EVALUATION

We evaluate the above metrics for the following protocols for comparison: (i) LoRaMAC (Aloha-like); (ii)  $p$ -CARMA with  $p$  fixed at  $1/N$ ; we call it the ‘static  $p$ -CARMA’ approach that is nothing but the theoretically optimum value of  $p$ ; (iii) adaptive  $p$ -CARMA without buffering and (iv) adaptive  $p$ -CARMA with devices using a buffer of 1 packet. We consider only a buffer size of 1 since if a new packet is generated before the current one is transmitted, the older one is discarded. Further, for adaptive  $p$ -CARMA we consider an ‘observation time’ of 10 hours, i.e., the CDR of each device is updated per 10 hours, as this time-duration is common between two consecutive ADR messages in reality. Furthermore, we show a subset of results for a few SFs for the sake of ease of representation and uncluttered presentation; the results for the other SFs also follow the same trend.

**PRR.** In Fig. 3.6, we compare the protocols with respect to PRR. First, we show the convergence of adaptive  $p$ -CARMA regardless of the starting  $p$ -value. The uncolored, transparent bars of Fig. 3.6c present the PRR of simulating adaptive  $p$ -CARMA under the same environment but for a starting  $p$ -value of  $1/N$ . It is apparent that the  $p$ -values converge and hence the PRRs converge as well.

$p$ -CARMA (static and adaptive) clearly outperforms LoRaMAC in all the SFs and numbers of devices.  $p$ -CARMA improves the PRR of 3000 devices by a factor of 3.4 for SF7 and by a factor of 22.3 for SF12. Regarding the two  $p$ -CARMA approaches (static and adaptive), for SF7 they perform similarly. The static  $p$ -CARMA slightly outperforms the adaptive one by at most 1.29 times for 3000 devices. The adaptive  $p$ -CARMA outperforms the static for 250 and 1000 devices. This is due to the fact that SF7 devices are the ones

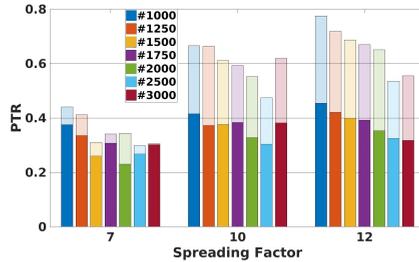


Figure 3.9: PTR (faded colors are for static- $p$ ).

closest to the gateway and are deployed as a disc. This creates a scenario with fewer hidden terminals. For the other SFs, the device deployment takes place in bands of rings around the gateway. In those cases, the hidden terminal issue becomes apparent, and the adaptive- $p$  clearly dominates static- $p$  for up to a factor of 5.25 for 2500 devices at SF12.

An important takeaway is that our adaptive  $p$ -CARMA performs at its best when the traffic/devices are higher, and hence the gains obtained for more than 2000 devices are seen to be the highest. This is because  $p$ -CARMA reduces collisions, even more than the static- $p$  case. The adaptive  $p$ -CARMA with buffer has even higher PRRs because the frames are not dropped with probability  $1 - p$ , which creates higher traffic – the scenario better suited for adaptive  $p$ -CARMA. Using buffer outperforms static  $p$ -CARMA even for SF7.

A higher PRR does not necessarily mean higher scalability. However, for a given PRR, more devices can be accommodated by using  $p$ -CARMA. As indicated by Fig. 3.7 for SF12, adaptive  $p$ -CARMA achieves a PRR of 20% while accommodating double the number of devices compared to the static approach and even more compared to LoRaMAC. Our metrics PTR and RoG that follow show that  $p$ -CARMA indeed transmits as many frames as possible while trying to avoid collisions whenever possible.

**Energy.** As we are interested in a scenario wherein there are numerous devices and many frames being transmitted, the subsequent graphs are for 1000 devices and more, as seen in Fig. 3.8. In energy terms, the static- $p$  strategy outperforms the LoRaMAC for high numbers of devices (above 1000). In particular, LoRaMAC consumes on an average 0.83 J per device regardless of the number of devices for SF12, while the performance of static- $p$  ranges between 0.40-0.79 J for the same case. While CAD is an overhead, as the number of deployed devices increases, CAD pays off due to reduced collisions and reduced transmissions. Consequently, the consumed energy is also reduced.

Adaptive  $p$ -CARMA outperforms both LoRaMAC and static  $p$ -CARMA in energy consumption, regardless of the number of devices. Specifically, the worst case of adaptive  $p$ -CARMA in SF7 consumes 0.4 times the energy of the corresponding LoRaMAC case. This ratio increases to 0.62 for SF12. The frame buffer case is not very different from the no buffer case for energy consumption, while the observed PRRs were higher. This implies that the adaptive  $p$  algorithm is effective and chooses the best time to transmit with its limited and imprecise information from CAD. While in backoff the devices sleep thus consuming a negligible amount of energy.

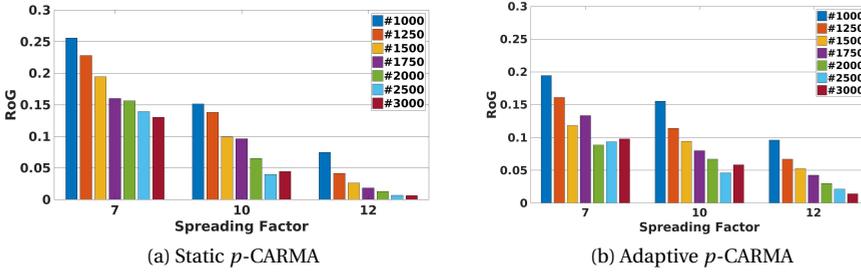


Figure 3.10: RoG.

	LoRaMAC	$p$ -CARMA		LoRaMAC	$p$ -CARMA		LoRaMAC	$p$ -CARMA
SF7	<b>1148</b>	884	SF7	<b>1332</b>	935	SF7	969	<b>979</b>
SF10	529	<b>668</b>	SF10	280	<b>460</b>	SF10	260	<b>582</b>
SF12	114	<b>302</b>	SF12	50	<b>213</b>	SF12	20	<b>143</b>

Table 3.3: Successful deliveries over 10,000 generated packets for 2000 (left), 2500 (mid), and 3000 (right) devices.

**PTR.** The reduction in energy consumption that is achieved by adaptive  $p$ -CARMA is due to the transmission of fewer frames than the other two approaches. To evaluate the reduction in transmissions, we plot PTR for the static and adaptive  $p$ -CARMA. It is evident from Fig. 3.9 that adaptive  $p$ -CARMA, in general, transmits fewer frames than LoRaMAC (which is 1 in this figure). Furthermore, for high SFs the adaptive  $p$ -CARMA transmits considerably fewer frames than static  $p$ , catering to the increasing hidden-terminal issue. The effectiveness of our scheme is exemplified because of the higher PRRs.

**RoG.** The reduction in the number of frame transmissions is clearly visible when referring to the ratio between received and the total generated packets (RoG), as seen in Fig. 3.10a and 3.10b, respectively. Generally, for both  $p$ -CARMA approaches the ratio of received over generated packets varies from around 5% to around 17% depending on the number of devices and SF (see Fig. 3.10).

Although the average  $p$ -value is higher in the adaptive  $p$ -CARMA, the devices transmit fewer frames than with static  $p$ -CARMA that uses  $p = 1/N$ . This seemingly inconsistent behavior is due to the fact that in  $p$ -CARMA, higher  $p$ -values mean a higher probability of non-hidden terminals transmitting and thus more devices refrain from transmitting. We outline an analytical explanation below.

Consider a ‘sensing’ device that has  $k$  devices in its sensing range. For the sake of simplicity, let us assume that the device can sense the entire frame reliably. These  $k$  devices performed CAD, but because another device in their vicinity,  $D$ , was transmitting, they found the channel occupied and are waiting for  $D$  to finish its transmission.  $D$  is not in the sensing range of our ‘sensing’ device. When  $D$  finishes, we will prove that the probability with which the ‘sensing’ device will sense a frame in the adaptive  $p$ -CARMA case,  $P_{\text{ada}}$ , is higher than in the static  $p$ -CARMA case,  $P_{1/N}$ . In the static  $p$ -CARMA, we have,  $P_{1/N} = 1 - (1 - 1/N)^k$ , the probability of sensing a transmission. In the adaptive  $p$ -CARMA, we have,  $P_{\text{ada}} = 1 - \{(1 - p_1)(1 - p_2)(1 - p_3) \dots (1 - p_k)\}$ . Since there are  $k$  elements in

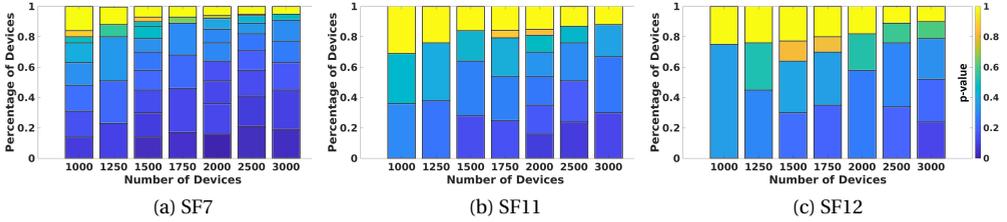


Figure 3.11: Spread of  $p$ -values versus number of devices.

both  $P_{\text{ada}}$  and  $P_{1/N}$ , without loss of generality,  $1 - 1/N \geq 1 - p_1, 1 - 1/N \geq 1 - p_2, \dots, 1 - 1/N \geq 1 - p_k$  (from Eq. (3.2)). Thus, we lead to,

$$(1 - 1/N)^k \geq (1 - p_1)(1 - p_2)(1 - p_3)\dots(1 - p_k). \quad (3.4)$$

and finally  $P_{1/N} \leq P_{\text{ada}}$ . This implies that there is a higher chance of sensing transmissions for the adaptive  $p$ -CARMA devices than for the static  $p$ -CARMA case. While the RoG values seem quite low, comparatively, adaptive  $p$ -CARMA achieves a significantly higher number of packets delivered successfully than LoRaMAC when the number of devices and/or SFs are high. This is observed in Table 3.3, where the successful receptions per 10,000 generated packets are shown. Even for cases of SF7 in which LoRaMAC has achieved more successful receptions, the number of failed transmissions is much higher compared to adaptive  $p$ -CARMA (cf. PRR, Fig. 3.6).

**Spread of  $p$ -values.** Fig. 3.11 presents the spread of  $p$ -values across the number of devices and SFs. Each device starts from  $p = 1$  and at the end of the simulation, we grouped the resulting  $p$ -values applied around prominent  $p$ -centroids using the Expectation-Maximization algorithm ( $p$ -centroids are dictated by the color-bars). Using the centroids we will be able to observe the general trend of how  $p$ -values are converging. As seen,  $p$ -values tend to lower values when the number of devices in the network increases, due to the increased traffic which updates Eq. (3.2) more frequently. This indicates the adaptive operation of our algorithm to the devices (or traffic). For SF7, while 20% of the devices have  $p$ -values higher than 0.8 when the scenario involves 1000 devices, this number decreases to less than 5% when 3000 devices are involved. For SF10 and SF12, although the same trend is seen, the  $p$ -values are higher, due to the higher values of the term  $CFF/(CFO + CFF)$  in Eq. (3.2), that is the result of the increased number of hidden terminals. Further, note that the standard deviation (SD) for all clusters with centroid  $p = 1$  in Fig. 3.11 is 0, while for the other centroids  $SD_p \in [0, 0.077]$ .

**Variations in PRRs of each device.** While there is a difference in  $p$ -values attained by different nodes because of different periodicity and their view of the channel, it is required to check how the PRR is achieved by each device. If the nodes use the channel politely, then they must have similar PRRs. It is important that the PRR across devices should not be heavily skewed even if certain devices have a higher  $p$  value. To ascertain this fact we use boxplots of PRR for various numbers of devices across all the devices in Fig. 3.12. In adaptive  $p$ -CARMA, the  $p$  values are chosen based on the perceived sensing activity.

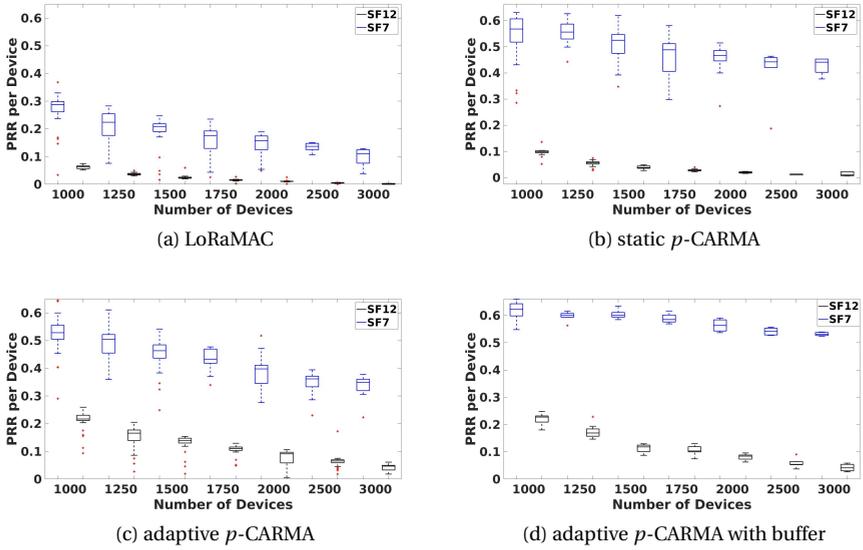


Figure 3.12: Average PRR per device for an increasing number of devices.

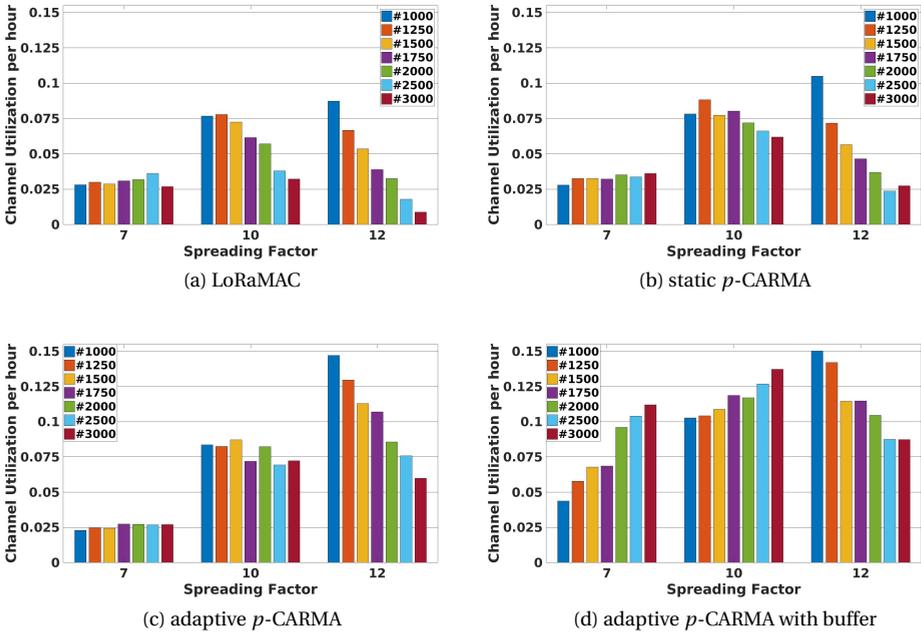


Figure 3.13: Normalized channel utilization for an increasing number of devices.

Group	Non-hidden	Group	Non-hidden	Group	Non-hidden
A	B, C	B	A, C, D	C and GW	A, B, D, E, F
D	B, C, E, F	E	C, D, F	F	C, D, E

Table 3.4: Groups of non-hidden devices for each group.

Therefore, the number of transmissions made per device is adapted. Fig. 3.12 indicates that the PRR variations are extremely small in both static and adaptive  $p$ -CARMA, almost converging to a single  $p$ -value. Furthermore, as the collisions are reduced, the PRR achieved by each node converges. A significant aspect to note here is that we observe higher PRRs for higher SFs with adaptive  $p$ -CARMA. Further, the performance is increased even more when a buffer is used, especially for SF7.

**Channel Utilization.** The efficiency of  $p$ -CARMA against LoRaMAC regarding the usage of the channel is observed in Fig. 3.13, especially for large numbers of devices and high SFs. For 3000 devices static  $p$ -CARMA outperforms classic LoRaMAC by a factor of 1.92 and 3.13 at SF10 and SF12 correspondingly. In the same cases, the adaptive  $p$ -CARMA is 2.24 and 6.86 times more efficient than LoRaMAC. Finally, when a buffer is used, the effective channel utilization is not only increased even more for high SFs, but also for SF7, where we observe a considerable improvement for any number of devices.

### 3.4. PRACTICAL EVALUATION

In order to evaluate  $p$ -CARMA in real-world LoRa deployments, we designed a simple testbed involving 30 LoRa SX1261 *class A* devices (with STM32 Nucleo-F446RE MCU) and one gateway indoors emulating a LoRa network. As shown in Fig. 3.14 the 30 devices are clustered into six groups of five devices. The RSSI and successful reception of frames transmitted from different locations to the gateway (location C) were extensively studied. The groups of devices were positioned in such a way to mimic LoRaWAN with hidden and non-hidden terminals for SF7. Each group can sense only a subset of other groups, as seen in Table 3.4). The indoor environment offers higher fading because of walls hence we observed higher signal attenuation. This setup mimicked an open-field LoRaWAN with devices positioned at large distances also involving hidden devices. To emulate thousands of devices we increased the duty cycle (above 1%) and the size of the transmitted frames to 200 B so that the offered load,  $G$ , is close to what is seen by a gateway in practice to test scalability. We evaluated one hour of operation under  $p$ -CARMA (with and without buffer) and vanilla LoRaMAC for  $G \in [0.25, 0.5, 1.0, 1.5]$  at SF7 with transmission power of 14 dBm. The traffic load  $G$  that is simulated at SF7 using *ns3* does not surpass the value of 0.19 (i.e., when 3000 devices were used). Therefore, this in-field experiment evaluates the performance of  $p$ -CARMA in serving deployments wherein traffic reaches and even surpasses the point of network saturation ( $G = 1$ ).

The PRR of each device is presented in Fig. 3.15 for SF7. As observed, in SF7 the devices which perform better when  $p$ -CARMA is used –either with or without buffer– vary from 25 to 27 out of 30, depending on the traffic. On the contrary, the vast majority of devices have 0% PRR if LoRaMAC is used. This confirms that  $p$ -CARMA is fairer than LoRaMAC regardless of the traffic load, offering higher chances of accessing the medium



Figure 3.14: Floor-plan of group-positions of thirty devices in six groups at SF7 (gateway in group C).

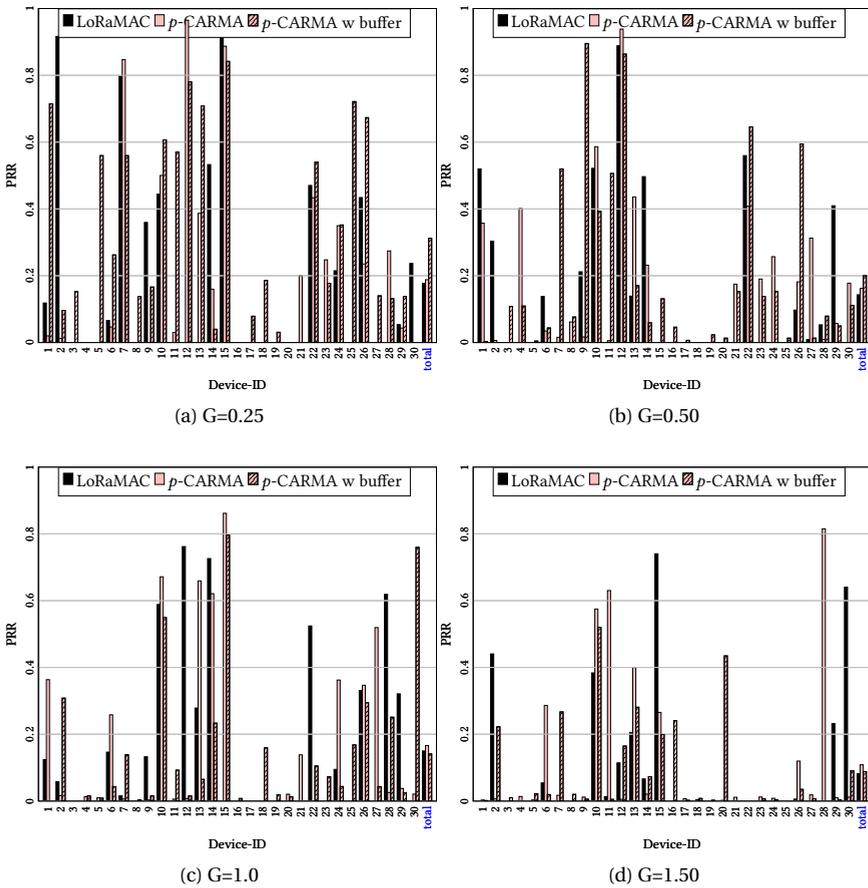


Figure 3.15: PRR per device:  $p$ -CARMA versus LoRaMAC and in total for different traffic loads at SF7.

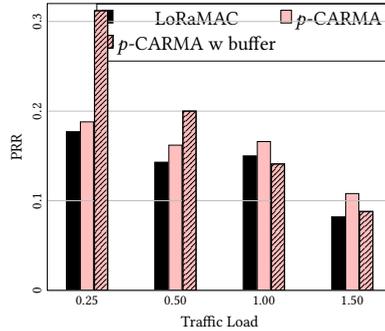


Figure 3.16: PRR for  $p$ -CARMA and LoRaMAC for different traffic loads at SF7.

to each device. The overall PRR for the two variances of  $p$ -CARMA and LoRaMAC is presented in Fig. 3.16. As traffic increases in higher values than the maximum of  $G = 0.19$  with which SF7 was simulated, it becomes obvious that the adaptation of  $p$ -value is not enough to evade collisions from hidden devices. Thus the original  $p$ -CARMA (without buffer) performs only slightly better than vanilla LoRaMAC for  $G = [0.25, 0.5]$ . When a buffer is applied, each packet gets more chances to be transmitted and the  $p$ -value of every device is adapted more accurately since the waiting times of each packet are drawn more elaborately due to the back-off policy followed at state CAD<sub>2</sub>.  $p$ -CARMA with buffer outperforms LoRaMAC by  $1.76\times$  to  $1.4\times$  for  $G = [0.25, 0.5]$ . However, after channel saturation ( $G = 1$ ), due to the sheer numbers of packet-arrivals the adaptation of  $p$ -values does not suffice to evade collisions from hidden terminals, and thus  $p$ -CARMA with buffer performs similar to classic LoRaMAC. At these high values of traffic, the original  $p$ -CARMA without buffer performs slightly better than the rest because it drops packets that would have been collided even if the buffer was utilized.

### 3.5. CONCLUSIONS

Long Range Wide Area Networks (LoRaWAN) need to scale in order to cover large areas and to cater to large numbers of IoT-devices in the future, such as in Smart Cities (Chapter 1). This work is focused on improving the scalability of LoRaWANs in an energy-efficient way by making devices polite. To this point, we proposed a new MAC protocol for LoRaWAN, called  $p$ -persistent Channel Activity Recognition Multiple Access,  $p$ -CARMA, which is based on the classical  $p$ -CSMA protocol.  $p$ -CARMA is distributed, less complex, scalable, and offers unslotted medium access to the LoRaWAN devices utilizing (imperfect) Carrier Activity Detection (CAD) for channel sensing. Real-world experiments were carried out to evaluate the performance and limitations of CAD in terms of distance and accuracy of preamble detection.

Further, a complete *ns-3* model was created to simulate  $p$ -CARMA on LoRaWAN. The design principle of LoRaMAC is to utilize ALOHA to ease the deployment and usage of IoT devices. However, it is known to perform badly with more devices. Our adaptive  $p$ -CARMA was shown to achieve a higher Packet Reception Ratio (PRR) as well as a higher channel utilization than LoRaMAC, especially when the number of devices/traffic is high, as  $p$ -

CARMA reduces collisions. This is achieved through sensing the channel and choosing an appropriate value of  $p$  adaptively. For example, in a deployment involving three different orthogonal SFs (7, 10, and 12) and 2000 LoRa devices per SF having transmitted the same number of frames each, our adaptive  $p$ -CARMA manages to reduce frame collisions by 20% compared to the current LoRaWAN, consuming almost half the energy (0.48) on average per device.

Furthermore, it was shown that the energy consumption was lower than LoRaMAC even though additional CAD operations are executed. This is due to the reduced number of transmissions; it is cheaper energy-wise to drop a frame than to generate a collision, which will possibly trigger retransmissions. Due to the adaptive nature of our proposed algorithm, the PRR-values of each device have a very low deviation from the mean-PRR of the network.

The adaptive  $p$ -CARMA has been designed to perform better when the number of devices/traffic is high, and hence caters to handling scalability requirements. For low traffic scenarios and lower SFs, it is easy to see that static  $p$ -CARMA is sufficient with similar performance, however fails in real-world scenarios with large numbers of devices and increased traffic scenarios. The most important aspect of our adaptive  $p$ -CARMA is that it can be utilized on existing LoRaWAN deployments without requiring any changes in the LoRaWAN specifications but minor computation on gateways. The devices with  $p$ -CARMA can coexist with LoRaMAC devices making them completely interoperable.



# 4

## A NON-PERSISTENT DISTRIBUTED MAC PROTOCOL FOR LORAWAN

THIS chapter aims to raise the capacity of LoRaWAN in multi-gateway scenarios, while still respecting the design principles of LoRaMAC, as stated in Chapter 2, in order to be interoperable with current LoRa deployments. Although  $p$ -CARMA, presented in the previous chapter, increases PRR and channel utilization efficiently, there is room for improvement. First, as explained in Fig. 1.4 of Chapter 1,  $p$ -CSMA, whose principles are shared by  $p$ -CARMA, collapses in terms of throughput after channel saturation (i.e.,  $G \geq 1$ ). Further, the persistent nature of  $p$ -CARMA leads to a high number of CAD attempts. While this is beneficial for collision avoidance, under conditions of high normalized traffic (i.e.,  $G \geq 1$ ) it leads to pointless waste of energy, since the probability of the channel being found occupied is relatively high. Furthermore, in  $p$ -CARMA the probabilistic model of the Capture Effect (CE) that we developed in Section 2.3 is not taken into account. Therefore, LoRa-devices transmit under the maximum allowed power (i.e., 14 dBm) in order to ensure that their frames will be received. This affects critically the energy consumption and creates serious interference to LoRaWAN (and other coexisting ISM-networks).

To solve the aforementioned issues, we design the non persistent-Capture Effect, Channel Activity Detection Algorithm ( $np$ -CECADA), which raises the capacity of LoRaWAN while being energy efficient. Specifically, we obtain higher PRR as the number of LoRa-devices increases by avoiding frame collisions using the principles of  $np$ -CSMA in a LoRa network with significantly low feedback and several hidden devices. With  $np$ -CECADA, LoRa-devices: (i) utilize CAD not only to assess medium-occupancy but also to estimate the traffic in their vicinity and adapt their backoff value before attempting to transmit (see subsection 2.3.2); and (ii) leverage CE-probabilities to reduce their transmission power— saving energy and reducing collisions (see subsection 2.3.1). A distributed mechanism for the regulation of transmission power is designed based on the chances of each device to capture the channel while being interfered by transmissions of hidden devices. Note that the overhead due to  $np$ -CECADA is lower than classic  $np$ -CSMA, as our algorithm utilizes only the very rare downlink channels by the gateways and does

	LoRaMAC	<i>np</i> -CSMA	Naive <i>np</i> -CSMA (BEB)	<i>np</i> -CECADA
Low consumption	●○○○	●○○○	●●○○	●●●○
Low overhead	●●●○	●○○○	●●○○	●●○○
Low complexity	●●●○	●●○○	●●○○	●●○○
High PRR (effectiveness)	●○○○	●●●●	●●○○	●●●○
High channel utilization	●○○○	●●●●	●●○○	●●●○

Table 4.1: Qualitative Comparison of Carrier Sensing approaches (more filled circles indicate better performance).

not make use of RTS/CTS and acknowledgments. Gateways provide extra secondary assistance by broadcasting information on the network-PRR at large time-intervals. Nevertheless, *np*-CECADA is completely distributed and can be also applied without any gateway-assistance. Further, vanilla LoRaMAC consumes more energy as it transmits all generated packets, while CAD –to evade harmful transmissions– requires less energy to be performed. Table 4.1 compares qualitatively *np*-CECADA to LoRaMAC, classic *np*-CSMA, and Naive *np*-CSMA with Binary Exponential Back-off (BEB), demonstrating that *np*-CECADA should be the preferred approach taking into account the constraints imposed by LoRa networks.

Our contributions are multifold:

1. We design *np*-CECADA, an adaptive, distributed, backoff time-based CS MAC protocol for LoRa networks that can be employed on current LoRa-deployments **without** requiring any adaptation of the infrastructure.
2. To assess the performance of *np*-CECADA in scenarios involving tens of thousands of devices we designed its model in *ns-3*. The CAD and CE modules in *ns-3* incorporate the results discovered in our in-field experiments in order to represent closest-to-reality simulations. We compare *np*-CECADA not only with classic LoRa implementation, but also with the state-of-the-art algorithms in the MAC layer of LoRaWAN [45, 81] proving its superiority using many metrics.
3. We employed *np*-CECADA in our LoRa testbed with 30 devices. To compare at scale, we increased the offered load correspondingly. The real-world experiments showed *np*-CECADA improving PRR by 2-4.7× in SF7. It also allows 701 extra frame receptions per gateway per hour in SF10, even with offered traffic of  $G = 1.5$  (normalized per second), i.e., the traffic produced required 1.5× the available channel air-time.

### 4.1. *np*-CECADA

*np*-CECADA is a distributed, unslotted and low-complexity carrier-sensing based protocol. The probabilities governing every action of a LoRa-device under *np*-CECADA are denoted in a Markov model that describes the state transitions in Fig. 4.1. When a packet is generated with a probability  $P_g$  the device goes to CAD<sub>1</sub> state from Idle and senses the channel. If the channel is free (with  $P_{cf}$ ), the device calculates its chances to capture the channel against the interference from hidden devices. The information on interference (in turn hidden devices) is communicated to the device from the gateway once in a while with a broadcast message (see Section 4.1.3). If the device finds that its frame will reach

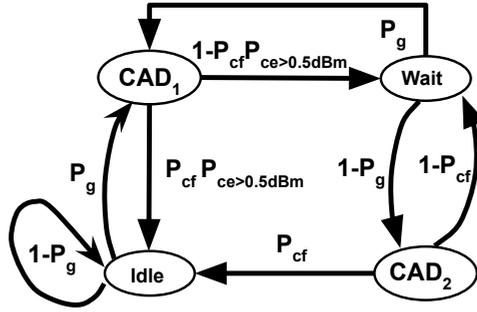


Figure 4.1: Markov model of *np*-CECADA for a single device.

the gateway with RSSI at least 0.5 dBm higher than the interferers (that is with  $P_{ce>0.5dBm}$ ), it transmits, adapts its transmission power for future transmissions, and then returns to Idle state.

On the other hand, if the channel is busy (i.e.,  $1 - P_{cf}$ ), or if the device finds its transmission power is less than the interferers even when it is free ( $P_{cf}(1 - P_{ce>0.5dBm})$ ), it backs off. The backoff time covers at least one ToA and is based on the probability of CE and local information of traffic (see Section 4.1.2). How the backoff value is computed is in Section 4.1.4.

At the end of backoff time, the device executes the second CAD (CAD<sub>2</sub>). If the channel is free, with  $P_{cf}$ , the device transmits and returns to Idle state, otherwise, it again adapts its backoff value and returns to the Wait state, and repeats CAD<sub>2</sub>. While in the Wait state, if a new packet arrives, with  $P_g$ , the current frame is dropped and repeats the above procedure for the new packet. The way of working of *np*-CECADA is presented in Algorithm 2.

#### 4.1.1. NETWORK BOOTSTRAPPING

In the initial phase of the operation of *np*-CECADA each device transmits with the maximum allowed power (e.g., 14 dBm in Europe). According to the LoRaWAN specification [54, 58] its frames will be received by any gateway that can receive them, but the network server will keep only the frames of the three gateways with the highest RSSI. When a device joins the LoRaWAN, in the device-acceptance message the gateway sends the observed path-loss exponent and the distance (devices and gateways are stationary). An **observation window** is defined, accounting for the maximum periodicity that a LoRa-device can have. For example, if each device transmits at least once per hour, then it is one hour. This time is split into time slots with predetermined granularity (we take 5 minutes). Each gateway keeps a count of the average RSSI of successfully received frames in every time slot. At the end of the observation window, the gateway broadcasts this information at every slot to the devices (i.e., 12 per hour in our case). The payload length depends on the granularity in time slots for which the averaged RSSI values are kept. Using this per slot RSSI information, devices can estimate their probability of capturing the channel for each gateway in every time slot knowing the path-loss exponent of its channel to the gateway and the distance. The above mechanism is repeated when new devices join and periodically.

#### 4.1.2. INITIAL BACKOFF MECHANISM

Whenever a device senses the channel free for the first time per generated packet (i.e., state  $CAD_1$ ), it calculates the signal strength of this frame when reaching the gateway. According to our in-field experiment on capture effect, if a frame is received at 0.5 dBm higher than the interference, the probability of being received successfully is higher than 50%, regardless of the SF. Knowing the average strength of interference in the specific time-slot, the device proceeds to transmission only if its frame is expected to be received with at least 0.5 dBm higher RSSI. Otherwise, it backs off with a value depending on its estimated probability of capturing the channel based on our in-field experiments on CE, see line 9 of Algorithm 2.

#### 4.1.3. ADAPTIVE TRANSMISSION POWER ALGORITHM

Each device also collects the statistics of the number of times it backed off or transmitted during an observation period (e.g., 1 hour). At the end of the observation period each device updates its transmission power as follows: if after finding the channel free in  $CAD_1$  state the device transmitted directly all the times or most of the times, it decreases its transmission power by 0.5 dBm or 0.25 dBm, respectively. Otherwise, it continues transmitting with the same power. Steps of 0.25 dBm is used since it provides adequate granularity. By gradually reducing their transmission power over time, devices reach a stage where their frames are received only by one or two gateways, the ones with the most favorable channel conditions, thus saving energy. With such a method energy can be saved as well as causing less interference. The changes in transmission power affect the stability of the whole LoRaWAN which needs to re-converge as the sectors of (non)-hidden devices are updated and the probabilities of finding the channel free/occupied are changed (see Section 4.1.4).

#### 4.1.4. ADAPTIVE BACKOFF ALGORITHM

In case of deployment of  $N$  LoRa-devices with no hidden devices, and assuming equal frame sizes, a naive Binary Exponential Backoff (BEB) algorithm can maximize the medium access. Assuming a uniform distribution of backoff times, the expected backoff time for each device after a certain number of collisions will converge to the average [97]. However, in LoRa networks there are many hidden devices and CAD cannot guarantee reliable channel sensing of non-hidden devices. Further, not all devices have the same periodicity in transmitting or the same number of neighbors in their vicinity. Therefore, an adaptive method of choosing backoff times is a must, fitting the needs of each device.

The Markov chain of Fig. 4.1 is positive, recurrent, and irreducible having the following transition matrix:

$$\begin{array}{c}
 \text{Idle} \\
 \text{Wait} \\
 \text{CAD}_1 \\
 \text{CAD}_2
 \end{array}
 \begin{array}{c}
 \text{Idle} \\
 \text{Wait} \\
 \text{CAD}_1 \\
 \text{CAD}_2
 \end{array}
 \begin{bmatrix}
 1 - P_g & 0 & P_g & 0 \\
 0 & 0 & P_g & 1 - P_g \\
 P_{cf}P_{ce>0.5dBm} & 1 - P_{cf}P_{ce>0.5dBm} & 0 & 0 \\
 P_{cf} & 1 - P_{cf} & 0 & 0
 \end{bmatrix}$$

A device can either be in a sensing state ( $CAD_1$  and  $CAD_2$ ) or in a non-sensing state (Idle and Wait).  $P_g$  is not affected by MAC. Further, the probability of estimating frames being received with more than 0.5 dBm than interference,  $P_{ce>0.5dBm}$ , depends on the

**Algorithm 2:** Pseudo code of np-CECADA

---

```

/* Algorithm works per packet arrival. Initial State: Idle */
Result: Transmit or drop the frame
1 Perform CAD1
2 if channel found free then
3   if Tx power of frame is found to be at least 0.5 dBm stronger than estimated
   interference then
4     Transmit the frame
5     Adapt transmission power
6     Return to Idle
7   else
   /* Probability to capture the channel is low. Backoff for
   at least ToA and retry. */
8      $T_x max = \tau + ToA + (1.0 - P_{ce})ToA$ 
   /*  $\tau$  is the current time */
   /*  $(1 - P_{ce})$  the complementary of probability to capture the
   channel */
9     Backoff with  $backoff\_time \in [\tau + ToA, T_x max]$ 
10  end
11 else
   /* Sensed part of a frame. Be polite to backoff for an
   adaptable amount of time. */
12   $T_x max = \tau + ToA + D_B$ 
   /*  $D_B$ , backoff delay, depends on local traffic (Eq. 4.1) */
13  Backoff with  $backoff\_time \in [\tau + ToA, T_x max]$ 
14 end
15 while  $backoff\_time > 0$  do
16   sleep ( $backoff\_time$ )
17   if new packet has arrived then
18     Drop current frame
19     break /* go to line 1 */
20   else
21     Perform CAD2
22     if channel found occupied then
23        $T_x max = \tau + D_B$ 
24       Backoff with  $backoff\_time \in (\tau, T_x max]$ 
25     else
26       Transmit the frame
27       Return to Idle
28     break
29   end
30 end
31 end

```

---

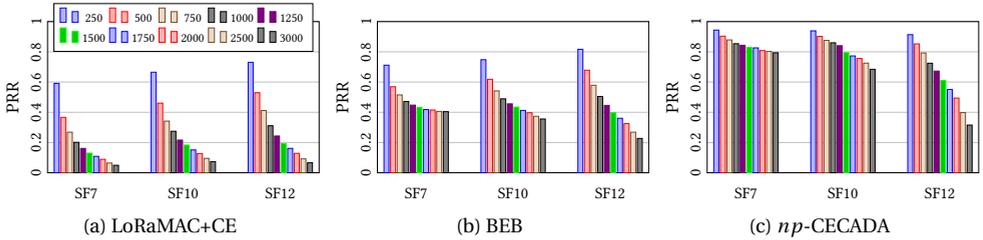


Figure 4.2: Means of PRR for increasing number of devices.

fading environment, the interferers, and the network topology. Therefore, the device being in a sensing state is affected majorly by the probability of finding the channel free,  $P_{cf}$ . According to Fig. 4.1, high  $P_{cf}$  leads to more transmissions and thus less time backing off (Wait state). However, frequent transmissions of one device reduce the probability of finding the channel free for its neighboring devices; they in turn spend more time waiting. This leads its neighbors' neighbors to be more aggressive since they find the medium free more often. From the above, it is obvious that we are led to a circular argument, as the network dynamics, i.e., transmission power and probabilities of transmission, dynamic traffic load, number of hidden devices do not allow an optimal solution regarding the backoff values. This Markov model cannot represent analytically the inter-dependency among devices in a LoRaWAN, and must be generalized to involve  $N$  devices. However, the complexity of such a model is immense as it should involve  $\mathcal{O}(4^N)$  states. Thus, we utilize heuristics to adapt the backoff value.

#### HEURISTIC APPROACH

The backoff value,  $B$ , of a device is adapted based on Eq. 4.1,

$$B = 2^{(1-p)\log_2 N} (1 + R + Q). \quad (4.1)$$

Where  $N$ , the number of devices, changes when devices join/leave the network, and this information is piggybacked on Adaptive Data Rate (ADR) messages by the gateway;  $p$  persistence probability found using the local information of each device regarding the traffic in its vicinity. Periodic device beaconing  $R$ , representing the traffic seen in the vicinity of a device's neighbors (regional parameter).  $Q$  is the percentage of collided frames (in each observation window), representing the traffic seen by the gateway. Note that  $R$  and  $Q$  are only weights used as feedback to support the computation of the backoff value  $B$ . The backoff value is majorly dependent on the value of exponent  $p$ , reconfirming the distributed character of  $np$ -CECADA.

#### PERSISTENCE PROBABILITY $p$ VIA LOCAL INFORMATION

$p$  depends only on local information that each device acquires by performing CS through CAD.  $p$  is computed by Eq. 4.2,

$$p = \frac{\bar{d} - d_l}{(d_h - d_l)} \frac{f}{(o + f)}, \quad (4.2)$$

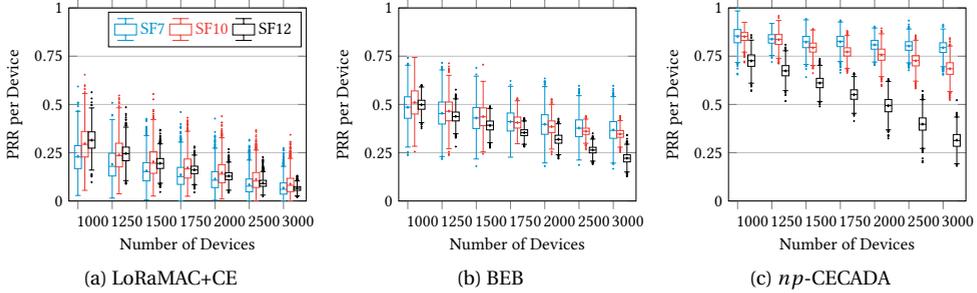


Figure 4.3: Mean values of PRR per device for increasing device numbers.

Transmission Channel	868.1 MHz
Code Rate	4/5
Initial Transmission Power	14 dBm
Bandwidth	125 kHz
Propagation loss model (mean, std)	Log-distance (0.56, 7.11)
Path loss exponent	3.76
Spreading Factors	7, 10, 12
CAD symbols	2 (SF7-8), 4 (SF9-12)

Table 4.2: LoRaWAN simulation parameters.

where  $\bar{d}$ ,  $d_l$ , and  $d_h$  represent the mean, the lowest, and the highest value of delay that device's frames experience from the moment they are generated till the moment they are transmitted.  $\bar{d}$  is proportional to the traffic load created by the non-hidden devices and to the probabilities of each device capturing the channel.  $f$  and  $o$  are correspondingly the number of times the channel was sensed free or occupied at state  $CAD_1$ .  $f/(o+f)$  denotes the probability to find the channel free on the first sensing, indicating a relatively low/high traffic in the vicinity. Using the Eq. 4.2, the exponent  $p$  adapts to the traffic that each device senses and to its channel capturing capabilities. The values of  $p$  are initiated per device the moment that the device performs its first backoff and  $p \in [1/N, 1]$ . The low bound of  $1/N$  offers to each device the minimum chance of attempting to transmit, since as  $p$  decreases the value of backoff,  $B$ , increases exponentially.

#### REGIONAL INFORMATION – PERIODIC DEVICE BEACONING $R$

The parameter  $R$  in Eq. 4.1 is updated through beacons sent by devices that experience heavy traffic in their vicinity in order to inform their neighbors about it,  $R = o/(o+f)$ . This results in each device being informed about the traffic caused also by hidden devices, i.e., neighbors of its neighbors. This indirect revealing of sectors of hidden terminals leads devices to refine their backoff values.

At predetermined large periodic times, all devices open their two receiving windows, and devices of heavily congested regions of the LoRaWAN broadcast a frame sharing their  $o/(o+f)$ . Each device decides if it will broadcast based on a weighted probability that favors the devices having the highest  $o/(o+f)$ , i.e., being in a more congested region of the network. To minimize regional beacon collisions, overhead, and energy consumption

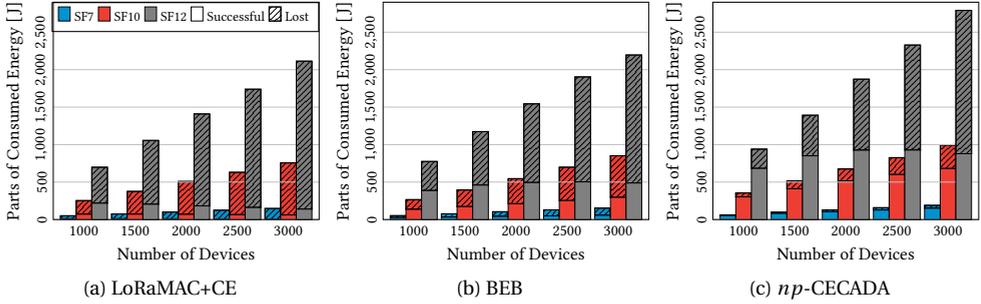


Figure 4.4: Energy consumed for successfully received and lost frames for an increasing number of devices.

4

each device is allowed to transmit at most once per three transmitting periods. Before the first round of  $R$ -values is received, each device  $i$  starts with  $R_i = 0$ .

#### GLOBAL INFORMATION – RATIO OF COLLIDED FRAMES $Q$

The value of  $Q_i$  represents the point of view of the network server regarding the frames of each device  $i$  that collided/dropped over the total number of frames – collided/dropped,  $C_i$ , and successfully transmitted,  $S_i$ – given by Eq. 4.3,

$$Q_i = \frac{C_i}{C_i + S_i} \quad (4.3)$$

The  $Q_i$  is computed by the network server between two ADR messages – using device ID and sequence number in the frame – and passes this information to devices via ADR messages. Sequence numbers help in finding lost frames between two successful transmissions for each device. Although  $Q_i$  is computed separately for each device  $i$ , all  $Q_i$  values are clustered into three groups: low, medium, high loss ratio. The number of groups can be higher depending on the heterogeneity of devices. Expectation-Maximization (EM) is used for clustering. The cluster centroids are embedded in the ADR messages and multicasted to the corresponding LoRa-devices. Multicasting the global view of collisions for each device is valuable feedback from the gateway, that is created without increasing the complexity (just counting frames and grouping) and without adding extra overhead to the communication through new messages.

## 4.2. SIMULATION SPECIFICS

Since we need to evaluate  $np$ -CECADA in Smart City scenarios involving thousands of devices per gateway, simulation was the sole way to encompass the high levels of traffic that are created and be realistic regarding the resulting distribution of transmissions. We utilized the open-source code of Magrin *et al.* [32] for  $ns$ -3 [44] as a basis of LoRa and LoRaWAN. Building on this existing code that simulates a vanilla-LoRaWAN, we designed all the modules and classes required for the application of  $np$ -CECADA. Further, we adapted our observations from the in-field experiments on CE and CAD into parameters of the LoRa PHY layer (cf. Section 2.3.1 and Section 2.3.2).

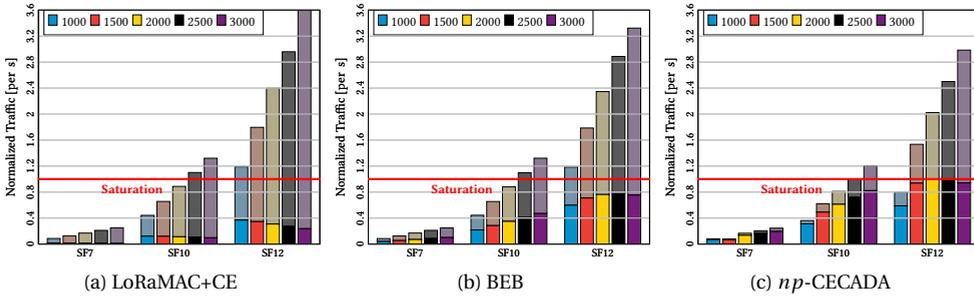


Figure 4.5: Normalized usage of the channel for an increasing number of devices. Faded colors represent the traffic, while vivid colors the useful channel usage, i.e., throughput.

We simulate scenarios that include up to three gateways and up to 45000 stationary devices positioned uniformly in ring-bands around the gateways. The devices transmit frames of 20 B which involve also 2 B for ID and transmission sequence. The configuration settings used for the simulation are the same as in the aforementioned in-field experiments. Further, a bandwidth of 125 kHz and frequency of 868.1 MHz are used for transmission. The parameters of the simulation are shown in Table 4.2. The values of probabilities of CAD-detection by distance are based on the graph of Fig. 2.17, i.e., a LoS scenario but with enough multipath interference that resembles a Smart City environment. This graph corresponds to the route depicted in Fig. 2.15b, which offers more or less the average of the two extremes shown in Fig. 2.15a and Fig. 2.15c, i.e., a LoS scenario but with enough multipath interference that resembles a Smart City environment. Each simulation result is the average of several independent runs (at least 30) to achieve confidence levels close to 100%. In each independent run, the positions of the devices remain the same, but their periodicities of transmission change. The metrics we used to evaluate the performance of *np*-CECADA are the following:

**Packet Reception Ratio (PRR).** The PRR is the ratio between successfully received frames and transmitted frames. It evaluates the ability of *np*-CECADA to scale by maintaining successful transmissions under conditions of increasing traffic.

**Energy consumption.** We monitor the energy consumed per device for transmitting, receiving, and performing CAD in order to evaluate whether and how much *np*-CECADA prolongs the battery life of LoRa-devices. Further, we measure the energy that is needed per single transmission.

**Channel utilization.** We measure useful utilization referring to the time the channel was used for correctly received frames over traffic, i.e., the total time the channel was used for transmissions. High values indicate a proper adaptation of the backoff time in order to handle increasing traffic loads, and thus show improved capacity for the LoRa network.

**Received over Generated packets (RoG).** RoG is the ratio between successfully received packets and generated packets. If combined with PRR, RoG indicates the level of network saturation, as it represents the actual portion of the generated traffic that was finally served.

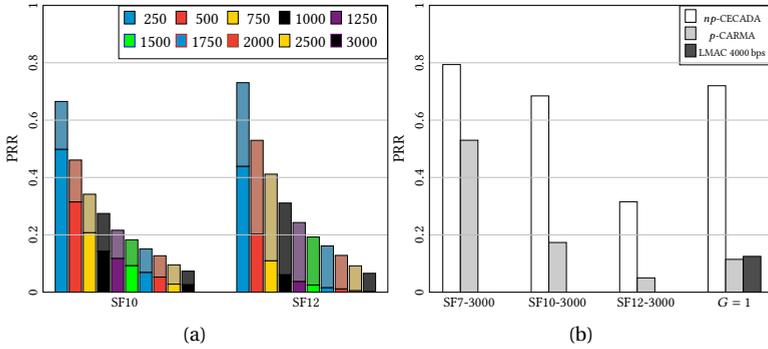


Figure 4.6: (a) The effect of accounting Capture Effect on LoRaMAC (faded colors correspond to CE being accounted). (b) Comparing  $np$ -CECADA to  $p$ -CARMA and LMAC in terms of PRR for different traffic loads (one gateway).

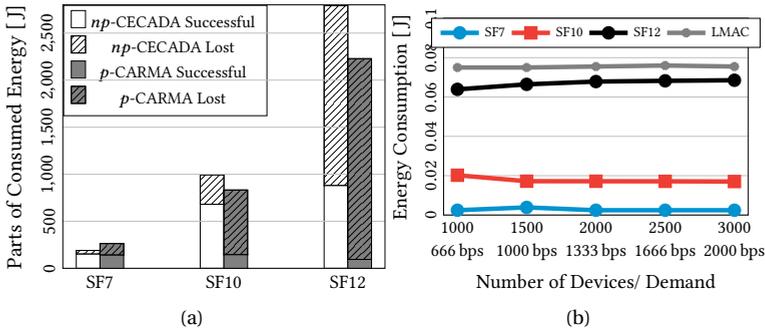


Figure 4.7: Comparison of  $np$ -CECADA to  $p$ -CARMA and LMAC regarding energy consumption. Consumption for successfully received and lost frames versus  $p$ -CARMA for 3000 devices (a). Consumption per frame versus LMAC (b).

**Fairness in Service.** We define fairness by evaluating the convergence of the values of PRR per device, i.e., the chance of every device to a clear channel.

### 4.3. SIMULATION-BASED EVALUATION

$np$ -CECADA is evaluated and compared based on the aforementioned metrics to the following protocols: (a) vanilla LoRaMAC (Aloha-like); (b) Binary Exponential Backoff (BEB); (c)  $p$ -CARMA [45]; (d) LMAC [81]. In the graphs, we specify "LoRaMAC+CE" for case (a) to differentiate this work from the bibliography of simulation-works because we applied specific and evaluated rules of capture effect to make decisions regarding frame reception. The case of BEB refers to MAC that uses CAD for CS and applies binary exponential backoffs once the channel is found occupied [97].

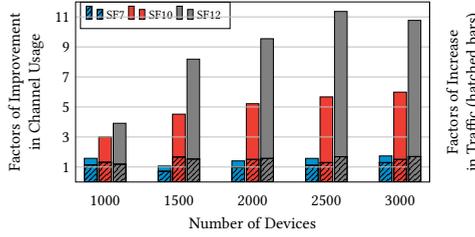


Figure 4.8: Improvement on channel utilization over  $p$ -CARMA for increasing number of devices (values of traffic).

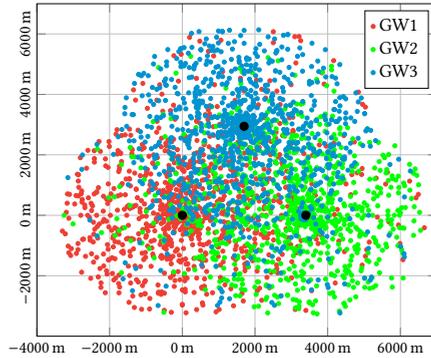


Figure 4.9: Distribution of 3000 SF10 devices to three gateways (with positions as black dots). GW1 at 0 m, GW2 at 3400 m, and GW3 at 1700 m on the x-axis.

Regarding  $np$ -CECADA, the first hour is used as "preparatory time" in which LoRa-devices transmit in Aloha fashion for the gateway to compute the mean RSSI of successful receptions and broadcast to the devices. We consider that during each simulation there are no changes in traffic so the above procedure is not repeated. Then, after each hour-timespan, the devices update their information on their chances of capturing the channel and adapt their transmission power. The regional traffic parameter  $R_i$  of each device  $i$  is updated per 20 minutes, but not broadcasted more than once per hour for reasons of efficiency and overhead. Gateways group devices in three clusters in terms of collisions (i.e., low, medium, high). The global collision parameter  $Q$  is multicasted by the gateway to the devices of each cluster once per hour, instead of once per ADR that was claimed in the Section 4.1.4 in order to converge faster, saving simulation time. We highlight only the results for SF7, SF10, and SF12 for ease of presentation, as the rest SFs behave accordingly. For a complete comparison regarding scalability, we begin with single-gateway scenarios and then proceed to multi-gateway ones, in which the transmission power adaptation mechanism is also used.

#### 4.3.1. SINGLE GATEWAY

**Notice 1:** The mechanism of transmission power adaptation is not applied to single gateway scenarios as the gateway to which all devices transmit is pre-dictated. Therefore,

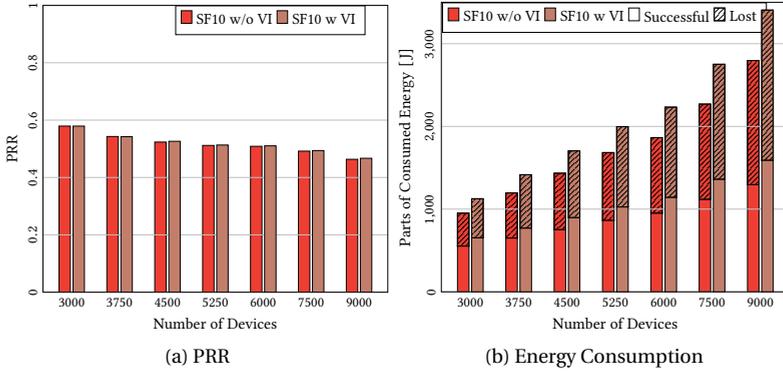


Figure 4.10: Comparison of  $np$ -CECADA with and without periodic device beaoning (VI) for increasing number of devices at SF10.

the results regarding energy consumption can be considered a worst-case scenario.

**Packet Reception Ratio.** Fig. 4.2 presents the performance of the protocols we simulated in terms of PRR. The improvement due to  $np$ -CECADA compared to LoRaMAC is manifolds, specifically from 16.03 times (SF7), to 9.30 times (SF10), to 4.78 times (SF12) when 3000 LoRa-devices are considered. Further, although BEB achieves higher PRR than LoRaMAC, it is also outperformed by  $np$ -CECADA by 1.96, 1.92, and 1.39 times correspondingly. The gain in scalability is obvious if we consider the number of devices served under the same PRR, e.g., for an average PRR-value of 60%,  $np$ -CECADA serves more than  $12\times$  devices than LoRaMAC in SF7.

**Fairness.** Although each device decides on its backoff value in a distributed manner based on its view of the traffic, the spread of individual PRR-values indicates the fairness of  $np$ -CECADA as a medium access protocol. Similar PRR-values assume a fair access policy, regardless of the different backoff values. As seen in Fig. 4.3, the binary exponential strategy of backoffs employed by BEB already manages a smaller variation of PRR-values than LoRaMAC, leading to a fairer distribution of transmissions. However, when  $np$ -CECADA is employed the PRR-variations become extremely small, especially in SF7, and SF10, almost converging to a single value. At the same time, the average PRR is increased, implying that  $np$ -CECADA improves the network's effectiveness collectively. Furthermore, outlying values are less than 3% in  $np$ -CECADA, indicating that the amount of devices under- or over-served is minimum.

**Energy Consumption.** It is obvious that the combination of mechanisms that guarantee the aforementioned increased performance in PRR have their price in energy terms, as devices in  $np$ -CECADA consume on average 1.3 times more energy compared to LoRaMAC, as seen in Fig. 4.4. However, let us observe how much of this energy is useful for the devices, i.e., leads to successfully transmitted frames. As seen in Fig. 4.4, devices in LoRaMAC waste almost all their consumed energy for transmissions that collide, especially at low SFs. The back-off policy employed in BEB offers better performance, but still much more than half of the total energy of devices goes for nothing in all cases. However,  $np$ -CECADA makes smart usage of the consumed energy, even in cases of many areas of

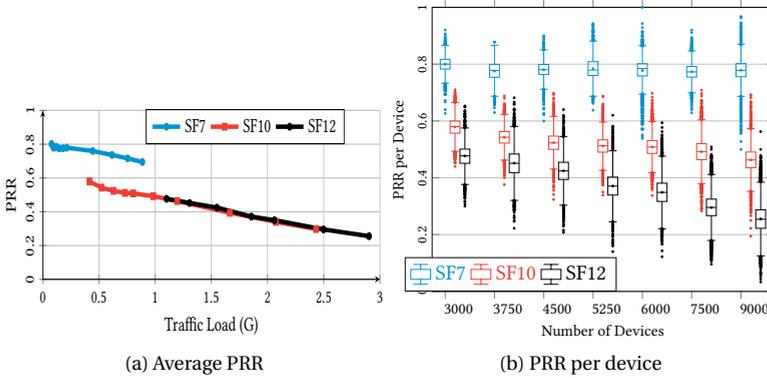


Figure 4.11: Mean values of PRR per device for  $np$ -CECADA with three gateways.

hidden devices like SF12. In particular,  $np$ -CECADA delegates from 31.50% to 85.23% of the consumed energy for successful transmissions. The better results of lower SFs like SF7 reconfirm the relatively higher effectiveness of  $np$ -CECADA in these SFs, see Fig. 4.2. Thus, although CAD is an overhead, as the number of deployed devices increases CAD pays off by reducing collisions and retransmissions. Furthermore, note that mechanisms like the periodic beaconing of regional information,  $R$ , can take place less often, leading to even lower values of energy consumption.

**Channel utilization.** As observed in Fig. 4.5, the application of CAD-CS boosts channel utilization due to better distribution of transmissions over time, especially when multiple thousands of devices are involved. In specific, for 3000 devices BEB improves over LoRaMAC by a factor of 3.18-8.38, and  $np$ -CECADA even more by 4-16.21, depending on the SF. Furthermore, when more than 1500 devices at SF7 are deployed, using  $np$ -CECADA allows us to stabilize the performance at a PRR of 0.8 only by increasing channel utilization, see Fig. 4.5c and Fig. 4.2c. This denotes the scalability potential of  $np$ -CECADA, especially in smart city scenarios, wherein thousands of LoRa-devices utilize low SFs to transmit [15].

Obviously, the same amount of transmitted bytes (20 B) does not incur the same ToA for all SFs. Therefore, for SF12, where ToA is 1.32 s the network is already saturated when more than 1000 devices are used. This happens when 2500 devices are employed at SF10. For SF7 the periodicity of transmissions should become very short in order to result in channel saturation since the ToA of each frame is 0.056 s.

**Comparison to the state-of-the-art.** Let us compare  $np$ -CECADA to  $p$ -CARMA [45] and LMAC [81]. Since  $np$ -CECADA utilizes a single-packet buffer, it is compared to the version of  $p$ -CARMA that uses a buffer. Further, since Gamage *et al.* utilize 16 combinations of two SFs and eight transmission channels, LMAC results are scaled down to one channel for a fair comparison.

First, we present the improvement in close-to-reality simulations that is offered by accounting the results from Capture Effect from real-world experiments. In Fig. 4.6a vanilla LoRaWAN (LoRaMAC) is simulated. Faded colors correspond to CE being accounted

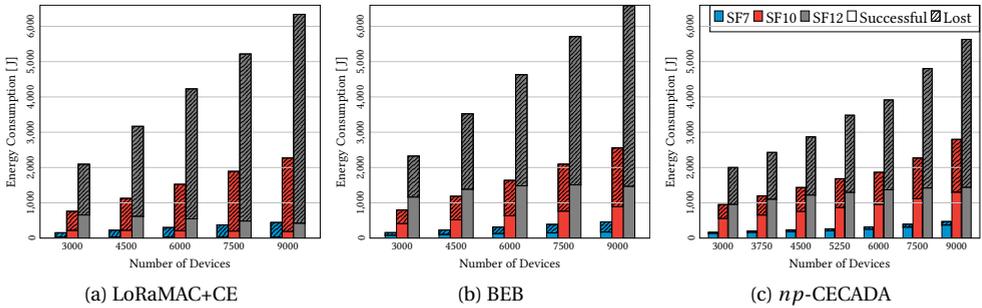


Figure 4.12: Energy consumed for successfully received and lost frames for three gateways.

while normal colors are taken from Fig. 3.6a of the  $p$ -CARMA paper (see Chapter 3). We compare only on SF10 and SF12 because in this chapter the devices are distributed in ring-bands on all the SFs, while in Chapter 3 the devices were distributed in circular disks on SF7 and in ring-bands on SF10 and SF12. As observed, accounting CE offers at least 35% improvement in PRR, i.e., 250 devices at SF10. At SF12, where a device can capture the channel easier—see Section 2.3.1—the improvement in PRR can even reach a factor of 30.58, i.e., 3000 devices.

Fig. 4.6b compares the effectiveness of  $np$ -CECADA,  $p$ -CARMA, and LMAC under different network traffic conditions. For LMAC, we use the results of 4000 bps in which the traffic load  $G = 1$ . For  $p$ -CARMA and  $np$ -CECADA the case of 3000 devices is used, which corresponds to different traffic loads depending on the SF.  $np$ -CECADA prevails by  $1.50$ - $6.31 \times$  compared to  $p$ -CARMA on 3000 devices and by a factor of  $5.76$  against LMAC.

Fig. 4.7 compares the protocols regarding energy consumption. Although  $np$ -CECADA spends more energy than  $p$ -CARMA at SF10 and SF12 due to the CAD and to listening for beacons, it clearly outperforms  $p$ -CARMA on the usefulness of energy consumption, as observed in Fig. 4.7a. In order to compare to LMAC, we present the per frame consumption in Fig. 4.7b. As seen, the energy consumed per transmission is not considerably affected by the increase in traffic. Even in the most inefficient case, i.e., transmission at SF12,  $np$ -CECADA spends around 10 mJ less than LMAC per frame.

In Fig. 4.8, we compare  $np$ -CECADA to  $p$ -CARMA in terms of useful normalized channel utilization under a different number of devices. The improvement in channel usage when  $np$ -CECADA is employed is multifold, confirming the effectiveness of  $np$ -CECADA even under congestion and its scalability potential. In specific,  $np$ -CECADA can lead up to a 6-fold and 11-fold improvement in effective channel use compared to  $p$ -CARMA. Note that this throughput improvement happens at the same or more traffic load (see the hatched bars of Fig. 4.8), since the smart adaptation of backoff values brought by  $np$ -CECADA leads to higher numbers of frames transmitted under the same number of nodes and transmission periodicities.

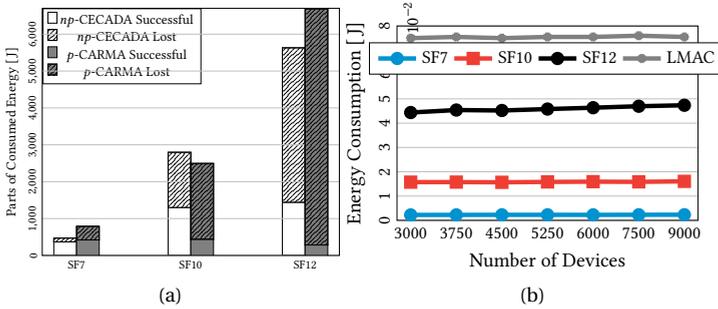


Figure 4.13: (a) Comparison of  $np$ -CECADA with  $p$ -CARMA and LMAC[81]. Energy consumption for successfully received and lost frames versus  $p$ -CARMA for 3000 devices per gateway; (b) Energy consumption per frame versus LMAC[81].

### 4.3.2. MULTIPLE GATEWAYS

The ability of LoRa-devices under  $np$ -CECADA to not only pick the right moment to transmit, but also adapt their transmission power for the benefit of the whole LoRa network is highlighted in scenarios involving three gateways. The gateways form an equilateral triangle with devices of the same SF (7, 10, or 12) distributed around them as seen in Fig. 4.9. Further, instead of transmitting only at 14 dBm like before, our devices employ the adaptive transmission power algorithm, explained in Section 4.1.3. Since the deployment of devices is not in ring-bands around one gateway anymore, in this case, devices that are relatively closer to a gateway can still achieve successful frame receptions by regulating their transmission power.

Furthermore, the mechanism that assists each device with providing regional information in terms of traffic –cf. periodic device beaconing in Section 4.1.4– becomes unnecessary, since regulating the transmission power already reduces the traffic from congested network areas. Fig. 4.10a indeed shows that under the current three-gateway scenarios the enhancement in PRR when the above mechanism is used –labeled as VI (Vicinity Information)– is trivial. At the same time a considerable amount of energy can be saved by not using periodic beaconing, as observed in Fig. 4.10b, i.e., from 174.43 J to 609.50 J in SF10.

**Packet Reception Ratio.** Fig. 4.11a presents the PRR-performance for three gateways and 1000-15000 devices per gateway. This corresponds to different traffic loads in the x-axis because packets of higher SFs incur more ToA. As observed, the presence of multiple gateways and many thousands of devices did not deteriorate the performance of  $np$ -CECADA. In specific, the PRR stabilizes between 70-80% in SF7, 30-60% in SF10, and 25-50% in SF12. Compared to  $p$ -CARMA using 9000 devices, i.e., the maximum for which  $p$ -CARMA is evaluated,  $np$ -CECADA outperforms it by  $1.5\times$  to  $5.13\times$  depending on the SF. Note that 9000 devices create a traffic load,  $G$ , of 0.21 in SF7 to 2.9 in SF12. For the same number of devices, vanilla-LoRaMAC is outperformed by  $3.87\times$  to  $15.74\times$  depending on the SF. Note that the improvement against both  $p$ -CARMA and LoRaMAC is similar to the single-gateway case. This reconfirms the scalability potential of  $np$ -CECADA to any number of gateways.

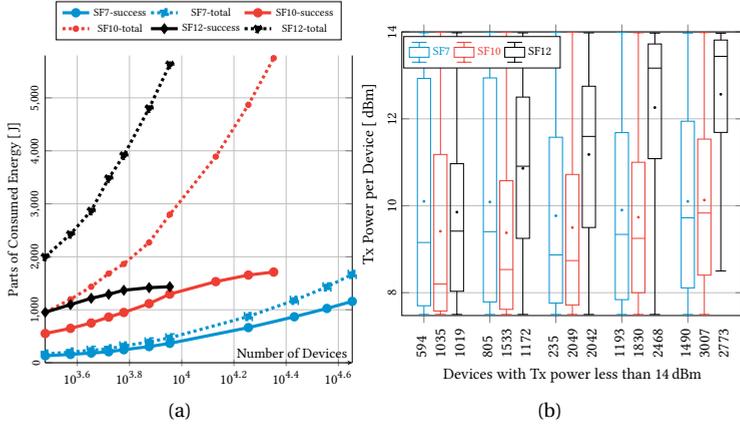


Figure 4.14: (a) Energy consumption for  $np$ -CECADA with three gateways for an increasing number of devices. Consumption for successfully received frames and total. (b) Distribution of transmission power values among devices that regulated their transmission power (from 3000 to 9000 devices with 1500 device-steps).

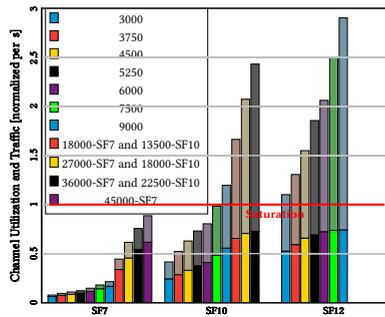


Figure 4.15: Normalized channel utilization (vivid colors) and incoming traffic (faded colors) for  $np$ -CECADA with three gateways.

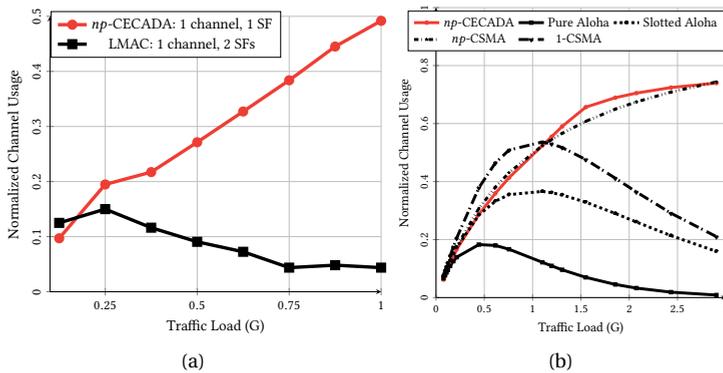


Figure 4.16: (a) Normalized channel utilization for  $np$ -CECADA and LMAC; (b) Throughput for  $np$ -CECADA and four main contention MAC schemes [30].

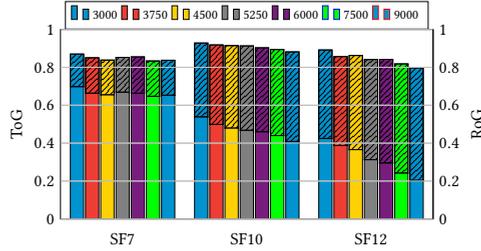


Figure 4.17: Average ToG and RoG values for an increasing number of devices for *np*-CECADA with three gateways. Dashed bars correspond to ToG values.

**Fairness.** As seen in Fig. 4.11b, the deviation of PRR is minimal. Box-plots infer that half of the devices in each case deviate utmost 0.025 from their mean; while, generally, the deviation-from-mean is less than 0.1. The outliers are merely up to 3.3%. *np*-CECADA allows equal chances for devices to access the medium. This confirms that the presence of multiple gateways does not affect fairness in service.

**Energy Consumption.** As observed in Fig. 4.12, regulating transmission power and avoiding transmissions that would collide saves energy. Especially in SF12, where the ToA of each transmission is relatively longer thus consumes more energy. However, *np*-CECADA consumes 30% less than LoRaMAC. In SF7 where less energy is needed per transmission, all protocols consume around the same energy. This is because CAD affects the overall consumption more drastically. Let us observe how much of this energy is useful (i.e., leading to successful transmission). As seen in Fig. 4.12a, devices in LoRaMAC waste almost all their energy for transmissions that collide, especially at low SFs. The backoff policy employed in BEB offers better performance, but still, more than half of the total energy goes for nothing in all cases, see Fig. 4.12b. However, *np*-CECADA makes smart usage of the energy, even in cases of many areas of hidden devices like SF12. In particular, *np*-CECADA uses 77.83%, 46.30%, and 25.49% energy corresponding to SF7, SF10, and SF12 for successful transmissions, while 8.06% and 37.01% are the highest that LoRaMAC and BEB can achieve respectively.

As observed in Fig. 4.13a, for 3000 devices per gateway only up to 53.17% of the total energy of *p*-CARMA is spent on successful transmissions in SF7. Even in this case *p*-CARMA is outperformed by  $1.5\times$  by *np*-CECADA. Further, as seen in Fig. 4.13b for the same payload LMAC consumes 75 mJ per frame while *np*-CECADA consumes at most 47 mJ (SF12).

Fig. 4.14a presents energy consumption for higher values of traffic when *np*-CECADA is utilized (dashed lines). Further, the parts of energy used for successful transmissions are also shown (solid lines). The amount of successfully used energy reaches a plateau at 7500 devices ( $G=2.5$ ) in SF12, showing that *np*-CECADA is unable to evade further frame collisions. In SF10 the plateau is also reached at around the same value of traffic, however, this corresponds to 22500 devices, since ToA is lower and more frames can be transmitted. SF7 has not yet reached a plateau even when 45000 devices are simulated, as the traffic increases slowly due to low ToA.

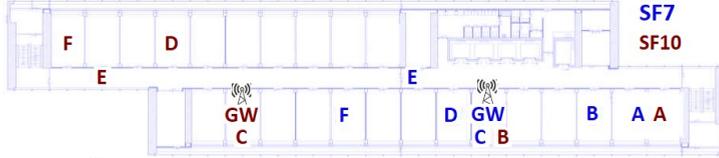


Figure 4.18: Thirty devices distributed per five devices per group in six groups. Floor-plan of campus building with group-positions according to their SF. Gateway positioned with group C. Group E is one floor below the other groups.

Group	Non-hidden	Group	Non-hidden	Group	Non-hidden
A	B, C	B	A, C, D	C and GW	A, B, D, E, F
D	B, C, E, F	E	C, D, F	F	C, D, E

Table 4.3: Groups of non-hidden devices for each group.

**Transmission power distribution.** Fig. 4.14b reconfirms the decrease in energy consumption when *np*-CECADA is used. In particular, around a third of devices in each case regulated their transmission power at lower levels. For example, around one-third of 9k devices in SF10 regulated their transmissions power to around 4.2dBm less. This corresponds to 1.4 mJ less per transmission and has a huge impact on consumption, since the 30% of the total number of transmissions refers to 58000 in the above case.

**Channel utilization.** As observed in Fig. 4.15, *np*-CECADA boosts channel utilization due to better distribution of transmissions over time. The increase in channel utilization (see vivid colors) versus normalized traffic,  $G$ , (see faint colors) exemplifies the scalability potential of *np*-CECADA, especially for SF7. This ratio stays between 80%-69% for  $G \in [0.08, 0.89]$  in SF7, 57.9%-29.7% for  $G \in [0.58, 2.43]$  in SF10, and 47.6%-25.5% for  $G \in [1.1, 2.9]$  in SF12. Notice that useful channel utilization stays stable in SF10 and SF12 even when the channel is well beyond saturation (red line); namely, utilization of 0.74 is achieved for both SFs even up to normalized traffic of 2.43 and 2.9 for SF10 and SF12, respectively. This illustrates the potential of adapting transmission time in high traffic conditions.

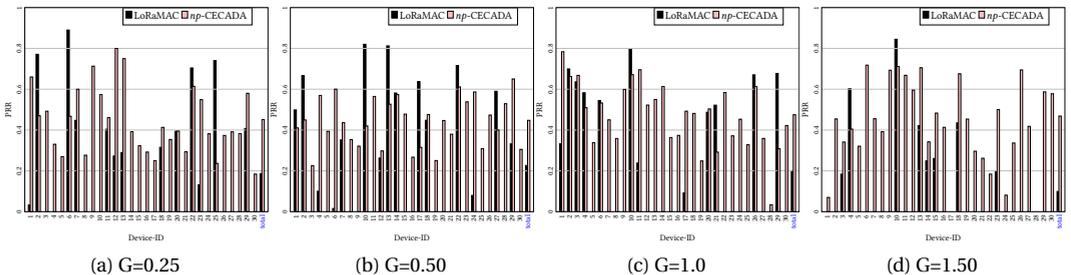


Figure 4.19: PRR per device: *np*-CECADA versus LoRaMAC and in total for different traffic loads at SF7.

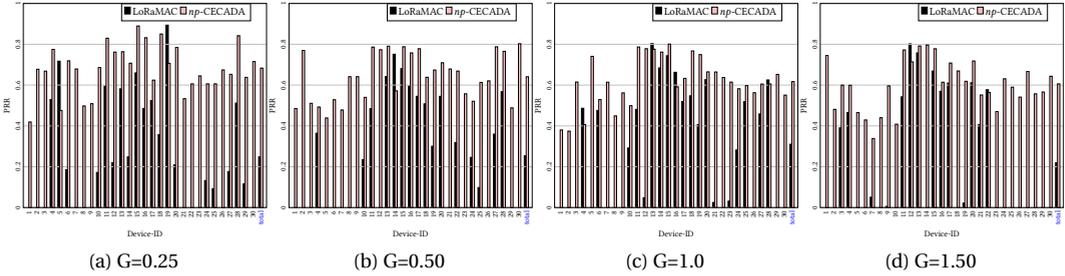


Figure 4.20: PRR per device: *np*-CECADA versus LoRaMAC and in total for different traffic loads at SF10.

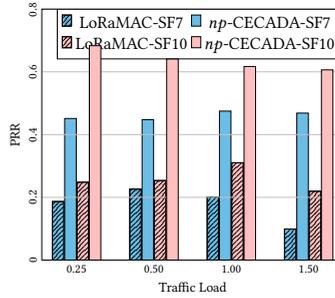


Figure 4.21: PRR for *np*-CECADA and LoRaMAC for different traffic loads and SFs.

In Fig. 4.16a, *np*-CECADA is compared to LMAC on a channel utilization basis for increasing levels of traffic. Since LMAC uses simultaneously 16 different combinations among 8 channels and 2 SFs, we downscale per channel. *np*-CECADA outperforms LMAC in any traffic condition. Specifically at  $G = 1$ , *np*-CECADA achieves 11-fold improvement.

The effectiveness in utilizing the medium by *np*-CECADA is highlighted in Fig. 4.16b, being compared even to the classic *p*-CSMA and *np*-CSMA protocols which use sophisticated feedback mechanisms (ACKs, RTS/CTS) and continuous sensing (*p*-CSMA). As observed, *np*-CECADA outperforms 1-CSMA above  $G=1$ . Further, *np*-CECADA behaves similarly to the classic *np*-CSMA while having minimum feedback from the gateway, with an inferior sensing mechanism (CAD), without using RTS/CTS, and while consuming minimum energy.

**Transmitted/Received over Generated frames.** Regarding transmitted over generated packets *np*-CECADA scores are always above 80% regardless of the traffic level. However, as seen in Fig. 4.17 for higher SFs and as traffic increases the actual percentage of those frames that manage to be received successfully decreases, obviously due to the increase in the traffic load. Nevertheless, *np*-CECADA outperforms *p*-CARMA by  $6.15\times$  to  $13.33\times$  depending on the SF for 9000 devices.

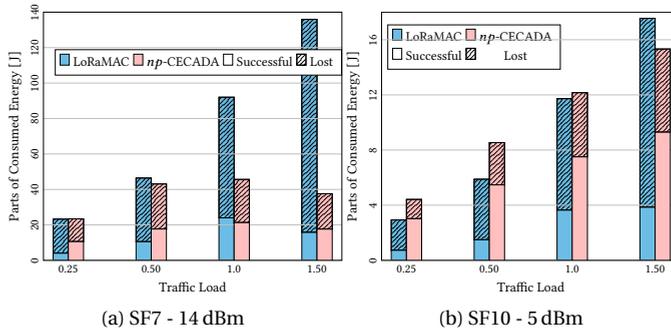


Figure 4.22: Comparison of energy consumption for *np-CECADA* and LoRaMAC with one gateway and thirty devices in-field for increasing values of traffic load. The parts of energy spent for successful and failed frames are depicted.

#### 4.4. PRACTICAL EVALUATION

It is hard to test *np-CECADA* in commercial LoRa deployments. Thus, we created a simple testbed involving 30 LoRa SX1261 *class A* devices (with STM32 Nucleo-F446RE MCU) and one gateway indoors emulating a LoRaWAN without interference from commercial operations. As shown in Fig. 4.18 the 30 devices are clustered into six groups of five devices deployed on two floors in our building. We first fixed the gateway and then we extensively measured the RSSI and successful reception of frames when transmitted from different locations. Then the locations of groups are carefully planned to mimic LoRaWAN with hidden and non-hidden terminals for both SF7 and SF10. Each group can see only a subset of other groups, e.g., group C devices can see all others, and group A devices see only groups B & C (See Table 4.3). The indoor environment offers higher fading because of walls. Hence, we observed higher signal attenuation. This setup mimicked an open-field LoRaWAN with devices positioned at large distances also involving hidden devices. To emulate large numbers of devices we increased the duty cycle (above 1%) and frame sizes so that the offered load,  $G$ , is close to what is seen by a gateway in practice to test scalability. We evaluated one hour of operation under *np-CECADA* and vanilla LoRaMAC for  $G \in [0.25, 0.5, 1.0, 1.5]$ . Multiple experiments were done with frames of 200 B and 40 B for SF7 and SF10, correspondingly. The transmission power used for the case of SF7 is 14 dBm, while for SF10 we used 5 dBm. The rest of the parameters remain unchanged. The PRR of each device is presented in Fig. 4.19 and Fig. 4.20 for SF7 and SF10 correspondingly. As observed, in SF7 the devices which perform better when *np-CECADA* is used vary from 24 to 28 out of 30, depending on the traffic. In SF10, 28 to 30 out of 30 perform better when *np-CECADA* is applied. More importantly, *np-CECADA* manages to distribute the transmissions much more fairly among the devices as shown by the lesser variations in PRR. On the contrary, the vast majority of devices have 0% PRR if LoRaMAC is used in both SF7 and SF10. Fig. 4.21 shows the overall PRR for SF7 and SF10. *np-CECADA* outperforms LoRaMAC by  $2\times$  to  $4.7\times$  in SF7 and by  $1.5\times$  to  $2.76\times$  in SF10 for  $G=[0.25, 0.5, 1.0, 1.5]$ .

In Fig. 4.22 the total energy consumption of the devices is presented, along with the parts of energy spent for successful transmissions and collided frames. *np-CECADA*

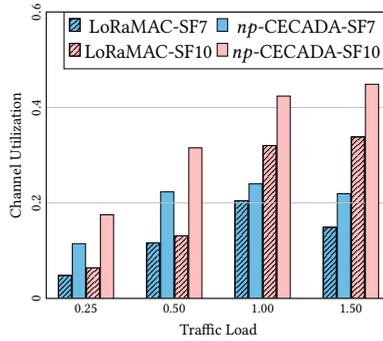


Figure 4.23: Channel Utilization for *np*-CECADA and LoRaMAC for different traffic loads and SFs.

spends energy not only for transmissions but further for the employment of CAD. As seen in Fig. 4.22a, in SF7 wherein the transmission power is at 14 dBm LoRaMAC spends up to 3.6 times more energy than *np*-CECADA for traffic load of 1.50, since the attempted transmissions are not regulated by the MAC layer. From this energy, only 11.6% results in successful frames; the rest is wasted. On the contrary, even at such a saturated network *np*-CECADA manages up to 46.8% useful energy consumption. At SF10, as observed in Fig. 4.22b, the overall consumption is heavily decreased for both the protocols since the transmission power is decreased at 5 dBm, i.e., transmission requires most of the energy spent. The difference in consumption between the two protocols is smaller because the energy-cost of transmissions is decreased, while the cost of CAD remains the same. However, *np*-CECADA still dominates LoRaMAC in terms of useful consumption, using 60-68% for successful transmissions while LoRaMAC achieves at most 31%.

Channel utilization is presented in Fig. 4.23. In SF7 *np*-CECADA manages more than 20% even in the heavy traffic of  $G=1.50$  while LoRaMAC shows signs of collapse since its channel utilization decreases drastically. In SF10 for the same traffic *np*-CECADA has 0.45 utilization which is  $\approx 30\%$  more compared to 0.33 for LoRaMAC. The channel utilization of SF10 is higher than SF7 in Fig. 4.23 because under the same traffic SF10 creates more robust transmissions than SF7 due to encoding each symbol with more chips. This leads to less collided frames, see Fig. 2.7. Note that higher than theoretical utilization of 0.18 for LoRaMAC is because of CE.

## 4.5. CONCLUSION

We proposed a novel MAC-layer protocol for LoRa networks involving thousands of devices and multiple gateways, called the non-persistent Capture Effect Channel Activity Detection Algorithm (*np*-CECADA), which shares principles with the classic *np*-CSMA protocol in its way of working, i.e., devices sensing the medium and backing off if it is not idle. *np*-CECADA optimizes LoRaWAN without changing the protocol and gateway code by adapting the backoff duration at each device independently and distributively using the imperfect CAD that senses the channel. *np*-CECADA leverages the capture effect phenomenon, which is part of any wireless physical layer, to reduce the transmission power of each device without compromising PRR, leading to less interference in the

network. The effectiveness and limitations of CAD and the influence of the capture effect in frame reception were evaluated using real-world experiments in Chapter 2.3. The observations drawn from these experiments were accounted to design a complete *ns-3* model to simulate *np*-CECADA for thousands of devices and multiple gateways. In addition, in-field experiments were performed using 30 LoRa-SX1261 devices to cross-validate the performance of *np*-CECADA for increasing values of traffic load. *np*-CECADA devices were shown to increase PRR while keeping energy consumption at the same or lower levels compared not only to vanilla LoRaMAC, but also to LMAC and *p*-CARMA.

*np*-CECADA improves PRR by 4 to 16 times depending on the SF compared to LoRaMAC in a 3 gateways network and 9000 devices and outperforms *p*-CARMA by 1.5 to 5.13 times. Further, the cost per transmission with *np*-CECADA varies between 2-47 mJ, while LMAC consumes 75 mJ per frame. Furthermore, the adaptive characteristic of *np*-CECADA allows the individual PRR-values of devices to not deviate higher than 0.1 from the mean PRR of the network, guaranteeing a fair service. *np*-CECADA offers high levels of throughput even at heavily saturated networks, i.e.,  $G > 1.0$ , wherein the total duration of transmissions is higher than the air-time offered by the channel, resembling the performance of the classic *np*-CSMA but with very rare feedback from gateways and with no continuous active sensing. As observed by the in-field evaluation, even for an offered normalized traffic load of  $G=1.5$  the average PRR was 47% and 61% for SF7 and SF10, respectively; while with LoRaMAC the majority of devices did not even manage a single successful transmission. Finally, *np*-CECADA can be directly implemented in the legacy LoRa network without deviating from the existing standard. After successfully creating MAC protocols for LoRaWAN inspired by the CSMA mechanisms, for the next chapter we turn our attention to Time Division Multiple Access (TDMA) approaches.



# 5

## VIRTUAL TDMA FOR LoRaWAN: SCHEDULING TRANSMISSIONS THROUGH DEDICATED CHANNELS

THE MAC layer protocols proposed until now are distributed and contention-based, i.e., LoRa-devices sense the channel and, based on the outcome, adapt their  $p$ -persistence or back-off time in an attempt to find the "proper" moment to transmit using heuristics. The gateway may assist by providing broadcast-feedback on its view of the traffic, but this assistance is secondary. However, regardless of the effectiveness of the heuristics in estimating the traffic, the performance of  $p$ -CARMA and  $np$ -CECADA is bounded by the myopic view of each device, sensing only devices deployed in its locality, i.e., non-hidden terminals.

The obvious step forward in improving scalability from the MAC layer is to provide direct information to devices about the traffic in sectors of the network that are hidden to them. This would require the use of a centralized protocol, in which the gateway can schedule transmission times according to its global view of the traffic using time division. Time Division Multiple Access (TDMA) approaches have been shown to increase the capacity in LoRaWAN up to a factor of three [15]. However, the use-cases they can address are limited due to the following constraints: (a) the gateway adheres to the same duty-cycle constraints as the end-devices, being unable to update the transmission slots; (b) class A devices are neither allowed to extra receiving windows for feedback nor possess the energy to sustain more receptions; and (c) TDMA is not compatible with the existing LoRaWAN standards [58], as there is no unicast channel for downlinks.

In order to take advantage of the increased capacity offered by scheduling protocols like TDMA, while accounting for the basic LoRa-characteristics, we propose Spreading Factor MAC (SFMAC) for LoRa networks. SFMAC is an innovative, distributed protocol for the MAC layer of LoRaWAN, that dedicates a low data-rate (i.e., high-SF) channel to perform only channel sensing, by using CAD. High-SF CAD reveals data transmissions

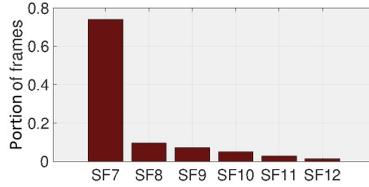


Figure 5.1: Portion of frames transmitted per SF based on 130,430 transmissions in the Antwerp testbed [102].

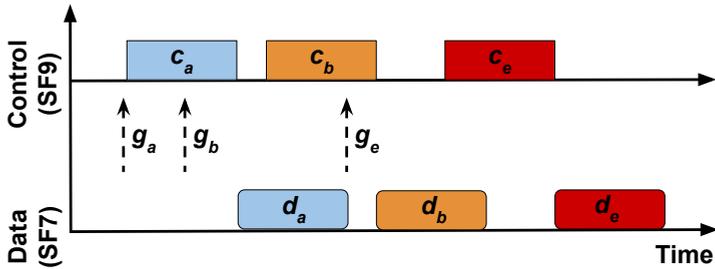


Figure 5.2: Time diagram of the operation of SFMAC with three devices.

5

from devices that would be considered hidden if low-SF CAD was to be used (see Section 2.3). Thus, the myopic view of the network for devices transmitting data at low SFs is extended significantly. Then, the devices can adapt their transmission times to evade packet collisions<sup>1</sup>. Devoting a low data rate channel to control purposes is reasoned with respect to the trade-off between the consumed energy and the provided channel capacity. Bor and Roedig have shown that configuring LoRa transmissions at high data rates, i.e., low SFs, is relatively more energy efficient [24]. Further, committing high-SF traffic to sense the channel comes without any considerable cost in terms of capacity, since a commonly used method for scaling in Smart City applications is to increase the number of gateways, which creates smaller cells of devices transmitting at low SFs [15]. The literature regarding LoRa deployment in urban environments confirms the above. Blenn and Kuipers analyzed the data from The Things Network (TTN), which is a large-scale open LoRa network [103] involving 229 gateways and 488 devices, and reported around 75% of LoRa transmissions at SF7 and SF8, with the vast majority of them being SF7 transmissions using 125 kHz of bandwidth [48]. In addition, in Fig. 5.1 we show the portion of frames transmitted per SF by a testbed of LoRa-devices and gateways deployed in Antwerp, Belgium, by Aernouts *et al.* [102]. As seen in Fig. 5.1, SF7 is used in more than 70% of the transmissions while the other SFs are underused. Furthermore, utilizing a high data-rate channel allows packets of longer payloads to be transmitted, i.e., 222-242 B of application-payload per packet for SF7-SF8. This enables a wider variety of applications.

**Principle behind SFMAC.** In SFMAC, pairs of *control-data* channels are utilized, employing SF9-SF11 for the transmission of control packets, i.e., frames of minimum preamble size, and SF7-SF8 for the transmission of data. The transmission of a data packet

<sup>1</sup>In SFMAC, apart from the dedicated high-SF channel, CAD also happens at the data-transmitting low-SF channel to reveal ongoing transmissions from non-hidden devices.

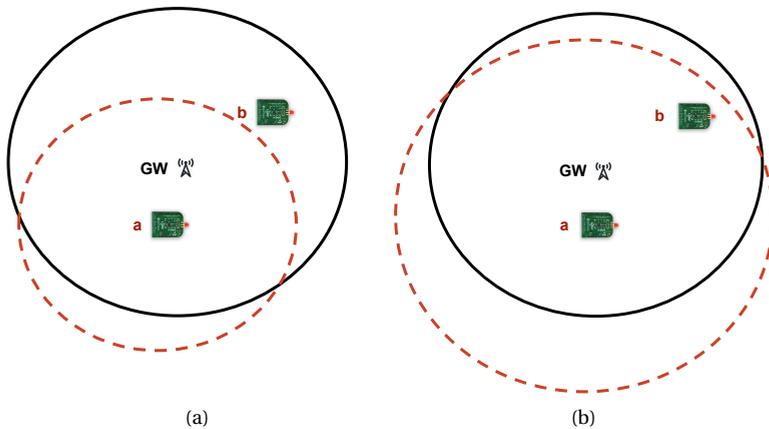


Figure 5.3: Channel sensing by CAD on the data channel (a) and the control channel (b). Gateway (GW) operates always on the data channel.

always takes place immediately after the transmission of a control packet. Therefore, any device that is ready to transmit in the territory covered by the control channel can be informed that there is an imminent transmission in the data channel. The time diagram of Fig. 5.2 depicts SFMAC's way of working. In this diagram,  $c_a$ ,  $d_a$ , and  $g_a$  represent the transmission of a control packet at the control channel, the transmission of a data packet at the data channel, and the generation of a packet, respectively, for device  $a$ . Further,  $a$ ,  $b$ , and  $e$  are three devices attempting to transmit their packets. Upon generating a new packet a device performs CAD at the control channel. If no control packet is detected, like in the case of device  $a$ , the device transmits at the control channel to inform as many devices as possible that it will proceed to data transmission ( $c_a$ ) and then switches to data channel to transmit ( $d_a$ ). Otherwise, like in the cases of devices  $b$  and  $e$ , the device backs off for an adaptable duration of time and tries sensing the data channel later, see for example the time difference between  $g_b$  and the beginning of  $c_b$ . When the data channel is found free, the transmissions of a control packet and a data packet take place in sequence without performing CAD on the control channel, see  $c_b$  and  $d_b$ , as all devices with pending transmissions are assumed to have already sensed the corresponding control packet previously, and switched to sensing on the data channel. Further, every extra switch between data channel and control channel adds complexity to the system and increases the energy consumption. The working principle of SFMAC, i.e., the sequential transmission of control and data packets, resolves packet collisions due to hidden devices in the way depicted in Fig. 5.3. When device  $a$  performs CAD at the low SF data channel, it is not able to sense any transmission from device  $b$  since it is a hidden device when low SF channels are used, see Fig. 5.3a. This would lead to frame collisions if  $a$  and  $b$  transmit simultaneously. With SFMAC, the device  $a$  first switches to a high SF control channel and performs CAD, as shown in Fig. 5.3b. Therefore, it can sense the transmission of control packets from device  $b$  and evade the imminent frame collision at the data channel.

Symbol	Justification	Value random Bw. $[t_1, t_2]$
$W_1$	Dependent on Time on Air (ToA) of the data-packets	[70, 90] ms
$W_2$	Allows adequate spread of chosen back-off times reducing collision probabilities	[200,400] ms

Table 5.1: Backoffs enforced in CADMAC.

**Contributions.** SFMAC increases the effectiveness of the transmissions at low SFs i.e., high Packet Reception Ratio (PRR), and improves the channel utilization at high traffic loads. Although a level of complexity is added to the system due to the need of switching between data/control channels, the overall capacity is increased through the effective utilization of the channel. To the best of our knowledge, SFMAC is the first MAC for LoRaWAN reserving underutilized, low data-rate channels for medium-access control. Our contributions are summarized as follows:

1. We design SFMAC, a distributed, energy-efficient, virtual TDMA protocol for LoRaWAN. **(a)** SFMAC utilizes a dedicated control channel and the CAD mechanism to inform each device regarding the traffic in network sectors that are out of the device's sensing range. **(b)** We construct and analyze the problem of choosing the Spreading Factor and length of control packets for the dedicated control channel. **(c)** SFMAC-devices can coexist with current LoRa-devices without requiring any change in the gateway/infrastructure.
2. We create an *ns-3* model of SFMAC to simulate scenarios with thousands of devices. Through our simulations, we offer critical insights on the performance of SFMAC against three state-of-the-art CS-protocols, *p*-CARMA [104], *np*-CECADA [46], and LMAC [81], and also against vanilla LoRaWAN regarding capacity, PRR, channel utilization, and energy consumption.
3. As a proof of concept, we evaluate SFMAC by utilizing realistic traffic from a monitoring application that uses cameras to preserve social distancing in public spaces, having filtered out the footage first. We show that by using a single gateway and wearables connected under LoRaWAN (instead of cameras), SFMAC can identify events where social distancing was not respected while being more efficient and without breaching privacy.
4. We applied SFMAC in our LoRa testbed with 30 devices, i.e., SX1261 LoRa modules. To compare at scale, we increased the offered load from  $G=0.5$  to  $G=2.0$  (normalized), showing that SFMAC improves PRR by  $3\times$  compared to LoRaMAC for high traffic values ( $G=1.5$  and above).

### 5.1. DESIGN OF MAC PROTOCOL

**CADMAC.** CADMAC is an end-node MAC protocol that relies on the LoRa radios being able to detect the transmissions of non-hidden devices through CAD. CADMAC will be used as a point of comparison between the simple vanilla LoRaWAN and the sophisticated SFMAC. The specific MAC actions that CADMAC enforces in the nodes are summarized in

these steps: (1) Upon generation of a packet, the node performs CAD in SF7. In case the channel is idle, the node transmits in SF7 and returns to sleep. If the channel is occupied, the node backs off for  $W_1$ . (2) After  $W_1$ , the node probes again the channel in SF7, leading to transmission in SF7 or backing-off for  $W_2$  if the channel is found occupied. (3) The last CAD attempt forces the node to transmit its frame with probability 1 either directly in case of free channel or after a  $W_1$  back-off in case of a busy channel. Table 5.1 summarizes the back-off values and the justification leading to them.

**SFMAC.** SFMAC extends the utilization of CAD for CS to a separate control channel, of low data-rate, dedicated strictly to the transmission of control packets. In this way, a LoRa-device can be informed regarding the state of the medium even by devices at relatively higher ranges than its neighbors, i.e., possible hidden terminals. Further, it allows transmissions of control-packets without affecting regular traffic, i.e., traffic in SF7. The control packet is a short sequence of downchirps with encoded value 0, i.e., only preamble symbols.

### 5.1.1. DESIGN OF CONTROL CHANNEL

#### OPTIMIZATION PROBLEM

SFMAC design can be formulated as an optimization problem when operating under different configuration parameters. We aim to find the optimal length and SF of control packets so that equilibrium among the following parameters is achieved: (i) minimizing energy consumption, (ii) minimizing the number of collisions, (iii) maximizing the range of channel sensing. The optimization was carried out using Gurobi 9.0.3 [105]. Most solvers offer optimization of linear objective functions. However, our objective function uses Gurobi solver since it minimizes non-convex quadratic objective functions. Our objective function is:

$$\min(\alpha[i, j](E[i, j]L[i, j] - D[i, j])) \quad \mathbf{s.t} \quad (5.2)(5.3)(5.4) \quad (5.1)$$

Eq. (5.1) minimizes the number of control-symbols, i.e., the length of the control sequence of the control packet,  $L[i, j]$ , and maximizes the range for reliable CAD,  $D[i, j]$ .  $E[i, j]$  is a ratio representing the energy cost of switching between SFs. In Eq. (5.1), index  $i$  represents SF7 (i.e., starting point of choice of control channel), and index  $j$  is any other candidate-SF for minimization.  $\alpha[i, j]$  is a score to qualify the improvement from SF $_i$  to SF $_j$ .  $\alpha[i, j]$  restricts the minimization problem since the sum of all the scores must be equal to 1, as stated in Eq. (5.2). For solving the minimization, the model must be feasible and we must define the constraints in Eq. (5.3) and Eq. (5.4).

$$\sum \alpha[i, j] = 1 \quad (5.2)$$

$$L > [2, 2, 2, 4, 4, 4] \quad (5.3)$$

$$D < \left[ \frac{D_{SF7}}{D_{SF7}}, \frac{D_{SF8}}{D_{SF7}}, (\dots), \frac{D_{SF12}}{D_{SF7}} \right] \quad (5.4)$$

The minimum values of variables  $L[i, j]$  are the number of symbols needed to perform CAD, derived from the data-sheet of Semtech on CAD [40].  $D_{SF7}$ - $D_{SF12}$  are obtained from

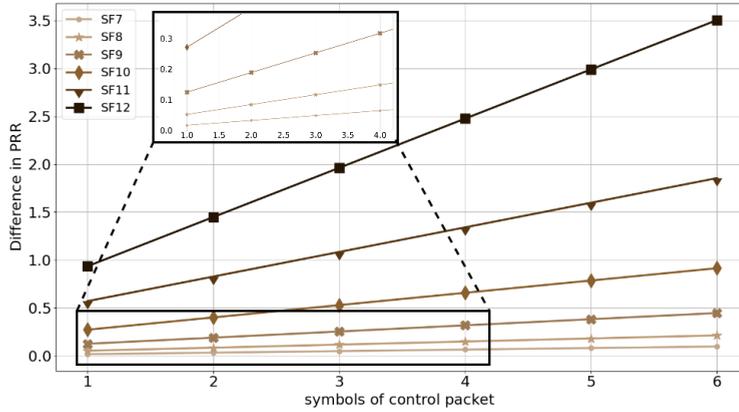


Figure 5.4: PRR improvement required per SF and length of control packet.

5

CAD experiments of subsection 2.3.2 and represent the maximum distance at which CAD is successful. Therefore, by using the ratio  $D_{SF12}/D_{SF7}$  for example, we obtain a measure of the improvement in reaching more nodes when choosing higher SFs. The optimization solver results in our variables being in the bounds defined in Eq. (5.2)-(5.4).

#### THROUGHPUT INCREASE BASED ON SF

The extra energy consumption when transmitting control packets in high SFs must be justified by the corresponding reduction in collisions, since for each collided packet the energy of the whole transmission is lost. Excluding the energy for data transmission, the remaining energy that is spent in a MAC protocol using a dedicated channel is denoted by  $E_1$ . Therefore,  $E_1$  comprises of the total energy spent in CAD at the control-channel and at the data-channel ( $E_{1,CAD}$ ) and the energy spent in transmitting the control-packets ( $E_{1,ct}$ ), as shown in Eq. (5.5). The energy of control-packets ( $E_{1,ct}$ ) equals the product of the electrical current needed per symbol ( $I_{ct}$ ), the Time on Air (ToA) of the dedicated SF,  $ToA_{ct}$ , and the symbol-length of the packet  $L_{ct}$ , seen in Eq. (5.6).  $E_2$  is the energy spent for CAD in a MAC protocol that does not use a dedicated channel, as seen in Eq. (5.7). For both of the above protocols, the energy consumed in data transmissions taking place at SF7 is  $E_T$ , as seen in Eq. (5.8); where  $ToA_7$  is the Time on Air of a LoRa-symbol in SF7,  $L_T$  is the number of symbols of payload, and  $I_T$  is the value of electrical current needed per symbol.

$$E_1 = E_{1,CAD} + E_{1,ct} \tag{5.5}$$

$$E_{1,ct} = I_{ct} L_{ct} ToA_{ct} \tag{5.6}$$

$$E_2 = E_{2,CAD} \tag{5.7}$$

$$E_T = I_T L_T ToA_7 \tag{5.8}$$

Our target is to define a relationship between: (i) the improvement in PRR that is accomplished by using control- and data-channel over using solely the data-channel, and (ii) the extra energy expense that is required when control- and data-channel are used. Thus, in

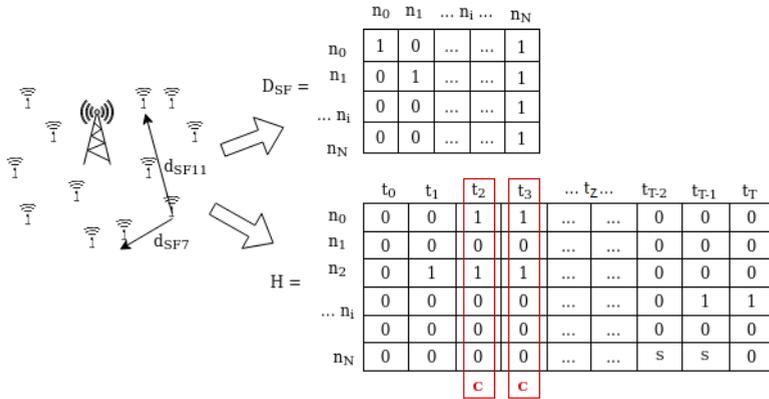


Figure 5.5: Representation of method to find the best parameters for the control channel. Values in matrices are an illustration of the information it contains.

Eq. (5.9) we equal the energy consumed by a MAC using control- and data-channel to a MAC using only data-channel, excluding the part of energy spent for successful packets in each case,  $E_T y_1$  and  $E_T y_2$ , where  $y_1$  and  $y_2$  are the PRR values in each case.

$$E_1 + E_T - E_T y_1 = E_2 + E_T - E_T y_2 \tag{5.9}$$

Simplifying the above expression, we obtain Eq. (5.10), which is a relationship between the energy expense and the PRR improvement,  $Q$ , that should be accomplished to match the equality of Eq. (5.9).

$$Q = y_1 - y_2 = (E_1 - E_2) / E_T \tag{5.10}$$

Situations wherein the  $Q$  is higher than 1 are not valid. Fig. 5.4 sets an upper limit on the collisions to be avoided once energy is spent in the control messages.

**Observation 1:** From Fig. 5.4 we observe that compensations in PRR are not achievable when utilizing SF10-SF12 for dedicated channels. Therefore, the next SF available, SF9, requires an increase of 13% in PRR when 1 symbol is used in the control packet. However, to increase the chances of detecting the control packet under CAD, 2 symbols are required for SF9 [40]. Consequently, the objective of increasing the PRR by 20% is chosen to compensate for the energy spent in transmissions of the control packet of 2-symbol duration in SF9.

**DETERMINATION OF OPTIMAL LENGTH AND SPREADING FACTOR OF CONTROL PACKETS**

The previous analysis relied on global statistics. However, the optimal length of the control packet and its SF also depends on the interarrival time of packets and topology. These dependencies of time and space have been modeled as shown in Fig. 5.5. The matrix representing distances  $D_{SF}$  summarizes the results of a device,  $a$ , sensing by using CAD the transmission of another device,  $b$ , according to our in-field CAD experiments. The matrix showing the state of the channel,  $H$ , represents the transmissions per node in

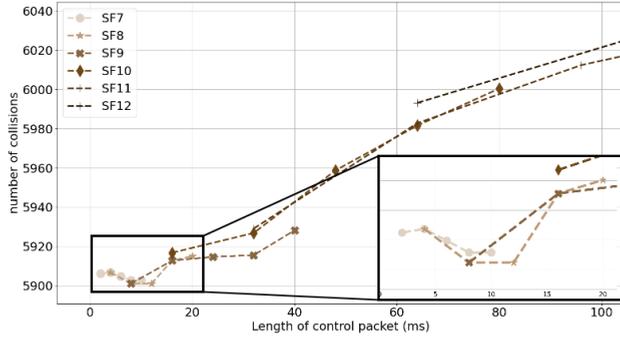


Figure 5.6: The number of collisions depending on the length of the control packet and the spreading factor. Average of 10 runs of the randomized process described above.

5

millisecond resolution. Therefore,  $H$ , will contain 1 in  $(a, b)$  if the node  $a$  is transmitting at that moment. Since a transmission length of 50 ms is considered (ToA of SF7), elements  $(a, b)$  to  $(a, b + 50)$  will contain 1 whenever node  $a$  transmits its LoRa packet starting at an instance  $z$  of time,  $t_z$ . In addition,  $H$  includes the control packet transmissions. To differentiate from the transmission of the payload, we label the presence of control packets in the medium with  $S$ . The number of elements filled with  $S$  depends on the length of the control packet. In order to characterize LoRaWAN under different lengths and SF of control-packets, we randomize and repeat the following process: (i) out of 1,000 nodes forming a ring around the gateway, 200 nodes are selected, ensuring heterogeneity of location of nodes, (ii) each node generates packets with an inter-arrival time defined by an exponential random variable. The scheduled transmissions of the nodes follow then a Poisson process. To find the optimal length of the control packet, we test from 2 up to 10 symbols. To reflect the effect of using a different SF, the matrix  $D_{SF}$  adds a 1 in locations  $D_{SF}[a, b]$  and  $D_{SF}[b, a]$  when nodes  $a, b$  can sense each other when increasing the SF. For every outcome of the setup, the elements of each column of matrix  $H$  are summed. As highlighted in Fig. 5.5 a collision ( $C$ ) occurs if the sum of a column in  $H$  is greater than 1. In Fig. 5.6, the average number of collisions out of ten runs of the above process is presented for each different configuration. As observed in Fig. 5.6, the benefit of reaching further nodes when using higher SFs gets limited by the number of nodes that listen concurrently to the control packet and thus can lead to collisions. In order to achieve balance among the aforementioned trade-offs, our final choice is SF9 and length of 2 symbols, i.e., 8.192 ms of the control packet. SF9 is preferred because its use represents less than 10% of all transmissions, as seen in Fig. 5.1. Further, orthogonality in LoRaWAN is imperfect, leading to collisions of packets of different SFs [106] and false detections of higher SF packets when performing CAD at lower SF [104]. Since SF9 is underutilized (see Fig. 5.1), there is less chance to falsely detect control packets as data packets.

### 5.1.2. SFMAC MODEL

The Markov model of the SFMAC mechanism is presented in Fig. 5.7. When a packet is generated,  $P_g$ , the device performs channel sensing on the control channel at state

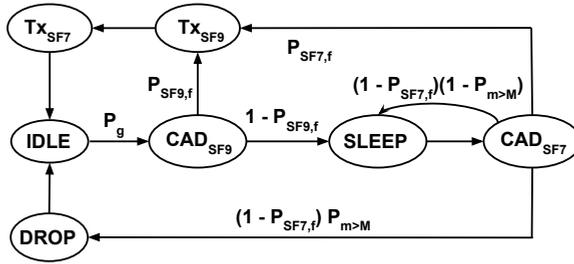


Figure 5.7: Markov model of SFMAC.

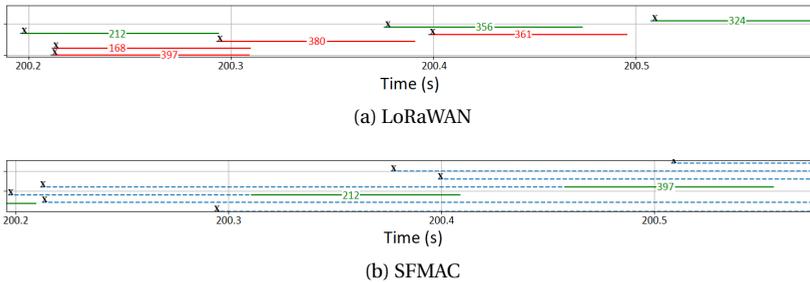


Figure 5.8: Timeline under vanilla LoRaWAN and SFMAC. The numbers on the lines indicate the identities of the devices that were transmitting. The crosses describe the generation of packets. Red bars represent collided packets. Green represent successfully received packets.

$CAD_{SF9}$ , to detect whether other nodes are announcing their imminent transmissions at lower spreading factors, i.e., data channel. In case of channel found free with probability  $P_{SF9,f}$ , the device transmits on the SF dedicated to controlling (SF9) its control packet, (state  $Tx_{SF9}$ ), and then transmits the data packet (state  $Tx_{SF7}$ ) at the data channel (SF7). If the channel is found occupied with probability  $1 - P_{SF9,f}$ , the device sleeps for a duration equal to payload-transmission at the data channel<sup>2</sup>. Upon waking up, the device contends for the channel in state  $CAD_{SF7}$ . In case of unsuccessful CAD at the data channel up to a specific number of attempts,  $M$ , the generated packet is dropped. SFMAC orders random arrivals of transmissions that coincide, thus operating in a TDMA-like manner. The model shown in Fig. 5.7 is analyzed under extensive simulations. Fig. 5.8 presents an example snapshot of the state of the data channel in the time domain of a simulation. As observed, under current LoRaWAN deployments, nodes will transmit as soon as they generate a packet (marked with a cross) resulting in several collisions, for example around the point 203.5 s. However, SFMAC can order the transmissions and spread them by allowing a structured contention period. The MAC actions that would be imposed in the nodes using SFMAC are summarized in Algorithm 3.

<sup>2</sup>The duration of payload-transmissions is either fixed or the average of previous transmissions, and is assumed to be known by the devices.

---

**Algorithm 3:** SFMAC

(CTRL: control-channel, DATA: data-channel)

---

```

1 Wait for packet generation
2 Perform CAD at CTRL
3 if CTRL found free then
4     Transmit packet in CTRL
5     Transmit packet in DATA
6     Idle until next packet generation
7 else
8     Sleep for  $s$ 
9     /* initial sleep-time  $s$  is equal to  $T_{oA}$  of the state  $T_{X_{SF7}}$  */
10    Perform CAD at DATA
11    if DATA found free then
12        Transmit packet in CTRL
13        Transmit packet in DATA
14        Idle until next packet generation
15    else
16        if  $m \leq M$  then
17            /* current CAD attempt at the DATA,  $m$ , not exceeding
18            maximum CAD attempts,  $M$ . */
19            Update duration of sleep-time  $s$ 
20            Go To line 9
21        else
22            Drop packet
23            Idle until next packet generation
24        end
25    end
26 end

```

---

Parameter name	Symbol	Value
Duration transmission payload	$D$	97.5 ms
Multiplication parameter	$N$	12
Duration initial listening time	$X$	$X = (N + 1) D$
Duration transmission control packet	$d$	$d = D/N$
Duration sleep time	$s$	$s = (N - 2) d$
Number of attempts allowed	$M$	-
Current attempt value	$m$	-
Min. contention window size	$C_W^{\min}$	-
Max. contention window size	$C_W^{\max}$	-
Contention window update policy	-	-
Contention window reset policy	-	-

Table 5.2: SF-MAC parameters.

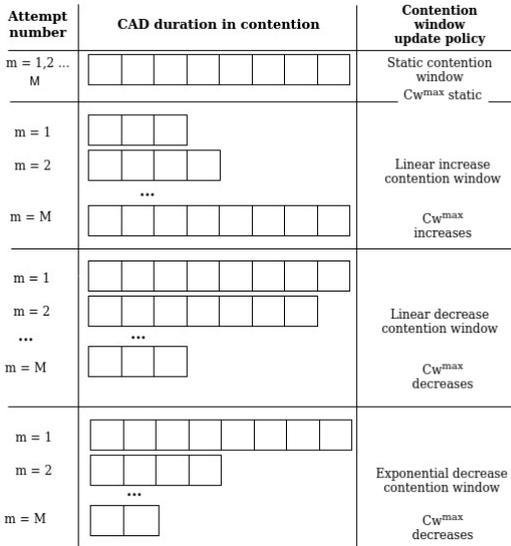


Figure 5.9: Contention window update policies.

### 5.1.3. OPTIMIZATION OF SFMAC PARAMETERS

The increase of the PRR of LoRa devices under SFMAC depends on the choice of the most appropriate values of the parameters presented in Table 5.2. The multiplication parameter,  $N$ , is a constant related to the duration of transmission in the data channel,  $D$ , and the duration of transmission in the control channel. In addition, the current number of attempts wherein the nodes are contending for the channel is labeled as  $m$ .

After detection of a control packet and the corresponding sleeping time, the contention period starts with the choice of a random listening duration, depending on the size of the contention window. The initial size of the contention window,  $C_W$ , and its increase/decrease per attempt are explained in Fig. 5.9. The contention window reset policy refers to the strategy the node follows to reset the current attempt number  $m$  either per generation of a packet, or per drop, or based on the idleness of channel observed over a reasonable amount of time.

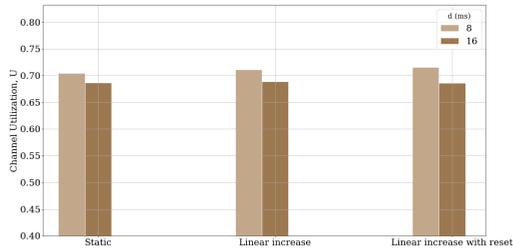


Figure 5.10: Channel utilization based on length of control packet.

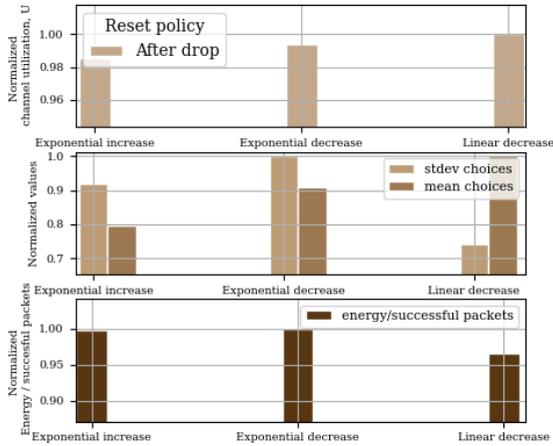


Figure 5.11: Evaluation of the update policies. The standard deviation, mean choices, collided, and dropped packets are normalized with respect to the highest value per metric considered. Therefore, the value of 1 is the highest value of that metric.

### DURATION OF TRANSMISSION CONTROL PACKET

According to our analysis, the minimum number of collisions occurred when 2 - 4 symbols were used for the control packet at SF9. Therefore, our protocol is implemented and tested under both 2 and 4 symbols of control packet. In Fig. 5.10, better channel utilization is achieved under shorter control packets. Two symbols of control packet ( $d = 8$  ms) can guarantee correct CAD while not delaying the transition to SF7 for data transmitting.

### CONTENTION WINDOW SIZE, UPDATE POLICY, AND NUMBER OF ATTEMPTS

Following the description of Fig. 5.9, we fixed the length of the control packet to 8 ms. Also, we chose our reset policy ( $m = 0$ ) after dropping, and the maximum value of  $m$  at 4. As observed in Fig. 5.11, the choice between exponential update and linear update is evaluated. The static contention window is not analyzed since it did not report better results than linear decrease (see Fig. 5.10). In the case of linear decrease, we set  $C_W^{\max} = 10$

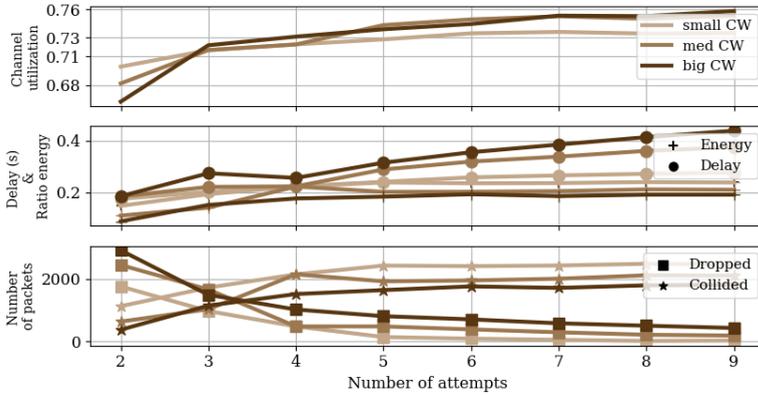


Figure 5.12: Channel utilization based on number of attempts and window size. Small, medium and big window size is respectively defined as  $C_W^{\max} = 5$ ,  $C_W^{\max} = 10$  and  $C_W^{\max} = 15$ .

and  $C_W^{\min} = 4$  to obtain the same result at  $m = 2$  (i.e., the most common value of  $m$  for both exponential decrease and linear decrease; both policies share the same window size of 8). Since the linear decrease starts with a smaller  $C_W^{\max}$ , the normalized standard deviation is the lowest for linear decrease (less values to choose from compared to exponential decrease). Regarding the change in window size, when the window reduces its size at a slower rate fewer packets collide since nodes can still choose from a considerable pool of values. Consequently, our study continues with a linear decrease update policy. Further, as observed in Fig. 5.11, with linear decrease we achieve 4% lower energy consumption compared to the exponential decrease update policy because linear decrease achieves better distribution of waiting times between the CAD attempts, leading to successful receptions with fewer CAD attempts. In Fig. 5.12 we show the impact of changing the maximum number of attempts under linear update policy. Although more attempts allow more opportunities to access the channel, they also allow a higher number of devices contending for the channel at the same time. Therefore, there is a trade-off between the chances (i.e., attempts) offered to each device in accessing the channel and the number of devices involved in the contention. This trade-off is reflected in Fig. 5.12, where there is no considerable gain in channel use when the number of attempts exceeds 5. Thus, we choose 5 attempts as they guarantee high channel utilization with low delay and energy consumption. Furthermore, based on Fig. 5.13 we choose the medium size of contention window and reconfirm 5 attempts and linear decrease with reset as the most suitable policy since the above combination achieves the highest channel utilization.

## 5.2. PARAMETERS OF SIMULATION AND EVALUATION

After having described the mechanism that guides the nodes to access the channel and transmit their frames, simulations of SFMAC have been performed in *ns-3*. Table 5.3 summarizes the simulation parameters. An exponential random variable of mean  $\lambda$  with a sufficiently low packet inter-arrival rate ensures a realistic LoRaWAN deployment. The

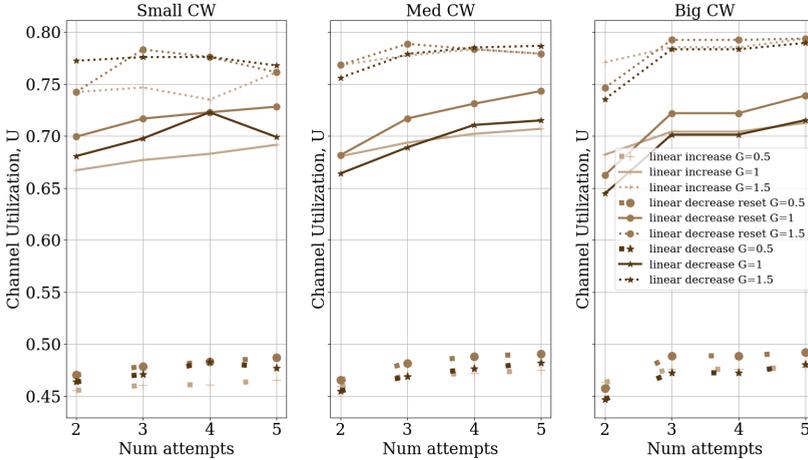


Figure 5.13: Channel utilization based on different traffic loads.

reduction of collisions achieved under Poisson traffic is transferable to other types of traffic. The end devices are located homogeneously around the gateway in a circle. The radius selected ensures that no hidden terminals are present in the deployment in order to evaluate the highest potential of both CADMAC and SFMAC.

### 5.2.1. EVALUATION METRICS

The LoRaWAN network performance is assessed using the following global and per-node metrics:

- Channel Utilization:** It refers to the time the channel was used for correctly received frames over traffic, i.e., the total time the channel was used for transmissions. This metric assesses how effectively the devices utilize the channel.
- Goodput:** It is defined as the number of correctly received units of information (bits) per second in our network. To obtain the goodput of our network, we calculate the ratio of total correctly received bits in the gateway and total observation time.
- PRR per node:** PRR reflects the ratio between packets successfully received by the gateway and the total number of transmitted packets by a specific node.
- Effective energy:** It is the product of Packet Error Rate (PER), defined as  $PER = 1 - PRR$  multiplied by the energy consumed on average per packet. The effective energy is a score to weigh the energy consumed based on the outcome of packet reception.
- Energy per transmission per node:** This metric shows the energy consumed per transmission per node. It is a measure to evaluate the cost of implementing a more complex

Parameter name	Value
Frequency of channel (MHz)	868.1
Number of devices, N	500
Number of gateways	1
Topology	Circle
Radius (m)	500
Packet size (B)	40
Coding rate, CR	4/5
SF of transmissions	7
SF of control channel	9
ToA (s)	0.0975
Offered load	$G$
Poisson rate (1/s)	$\lambda = G/\text{ToA}$
Transmission rate per node (1/s)	$r = \lambda/N$
Start time (s)	$\in [0, 100]$

Table 5.3: Parameters of Simulation.

protocol than ALOHA. We also provide results regarding the portion of the above energy that is consumed for successful/collided packets.

- **Packet Transmittance Ratio (PTR):** Since we impose a dropping policy in SFMAC, we evaluate the ratio of transmitted packets over generated packets.

Apart from the aforementioned metrics, we also specifically compute the number of collided and correctly received packets under different MAC protocols and the corresponding energy that is consumed.

### 5.3. SIMULATION-BASED EVALUATION

We add our observations of CE to LoRaWAN, CADMAC, and SFMAC to obtain a closer representation of collisions in a LoRa network benefiting from the capture effect (see subsection 2.3.1 for CE evaluation).

**Performance of network-wide metrics.** Fig. 5.14 presents the results of the simulations in  $ns$ -3 carried out for the normalized traffic values of  $G=0.5, 1, 1.5,$  and  $2$ . This corresponds to the generation of up to 64,800 packets per hour (for  $G=2$ ). Regarding channel usage, SFMAC improves over LoRaWAN by  $2.08\times$  under a heavily loaded network. For  $G=2$ , SFMAC achieves an average PRR of  $2.3\times$  higher, compared to LoRaWAN. LoRaWAN keeps a stable channel utilization for  $G=1, 1.5, 2$ , while CADMAC slowly decreases after reaching 0.6 at  $G=1$ . The reason for the stabilizing of LoRaWAN regardless of the increment of  $G$  is explained by the outliers of PRR in Fig. 5.14c. Since some *privileged* nodes are located close to the gateway, as observed in Fig. 5.15, their frames have higher chances to capture the channel against interference, reaching  $\text{PRR}\approx 1$ . In addition, the values of channel utilization observed in Fig. 5.14 help in determining when nodes should change from CADMAC to SFMAC. Since under  $G=0.5$ , these differences are not considerable and the energy spent in CADMAC per node per transmission is  $0.8\times$  the energy in SFMAC, CADMAC should be applied for these traffic values. However, for increased traffic, SFMAC outperforms CADMAC by up to  $1.78\times$  and must be preferred. The stability

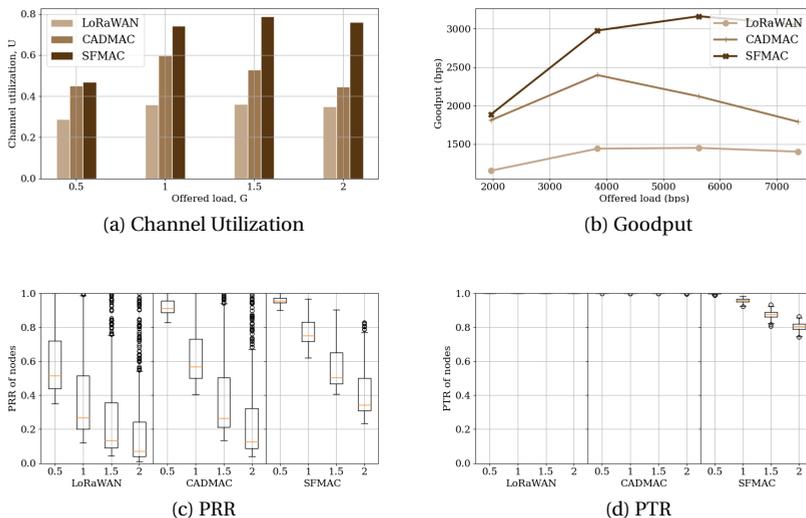


Figure 5.14: Comparison between MAC protocols including the capture effect.

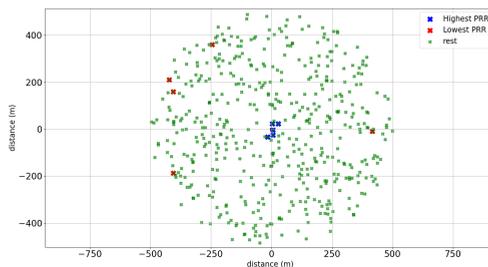


Figure 5.15: Location of nodes with best and worst PRRs, showing unfairness under LoRaWAN deployments due to the capture effect. Offered traffic considered is  $G=2$ .

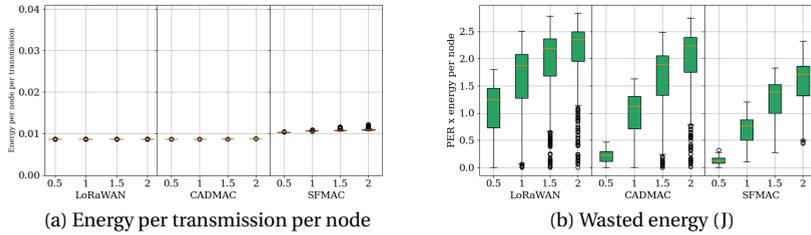


Figure 5.16: Comparison between existing MAC protocols considering the capture effect.

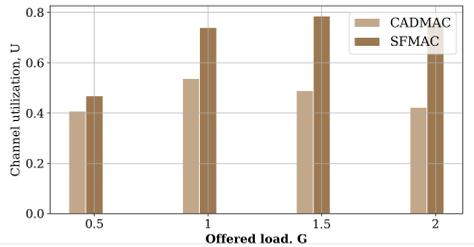


Figure 5.17: Impact of reducing 33.3% sensing range of CADMAC.

in the channel utilization of SFMAC is achieved after  $G=1.5$  since the value of channel utilization remains the same.

**Performance of individual metrics.** Fig. 5.14c reflects the fairness of the MAC protocol used in terms of the number of packets that are received successfully. In LoRaWAN and CADMAC, since there is no dropping policy, the nodes closer to the gateway manage a PRR of 1, compared to 0.1 reported by nodes further away. On the contrary, SFMAC clearly outperforms LoRaWAN and CADMAC by factors of 3 and 1.27 at  $G=1$ , respectively. Further, SFMAC maintains the PRR of nodes at least at 0.22 under  $G=2$ , which is the worst-case scenario.

**Energy overhead.** SFMAC reports in Fig. 5.16a a 37% increase in energy compared to LoRaWAN per transmission under the highest traffic scenario. However, this does not reflect the amount of energy spent for correct receptions or wasted for collided packets. Therefore, in Fig. 5.16b we calculated the amount of energy consumed for frames that did not reach the gateway during the observation time under the different MAC protocols. Fig. 5.16b shows that SFMAC decreases the energy spent in collisions per node from  $6.25\times$  at  $G=0.5$  to  $1.35\times$  at  $G=2$ .

**Impact of hidden terminals.** The results of Fig. 5.14 and Fig. 5.16 are obtained considering that CAD in all SFs has the same performance. However, as shown by our in-field analysis of CAD, the sensing range of CAD under higher spreading factors is higher (see subsection 2.3.2). In order to show the impact of hidden terminals and CAD performance when comparing SFMAC and CADMAC, the sensing range of nodes in CADMAC deployment is reduced by 33.3% in Fig. 5.17. Compared to  $G=0.5$  in Fig. 5.14a, CADMAC has reduced its performance in the best-case scenario by 9.5%. Further, as the traffic

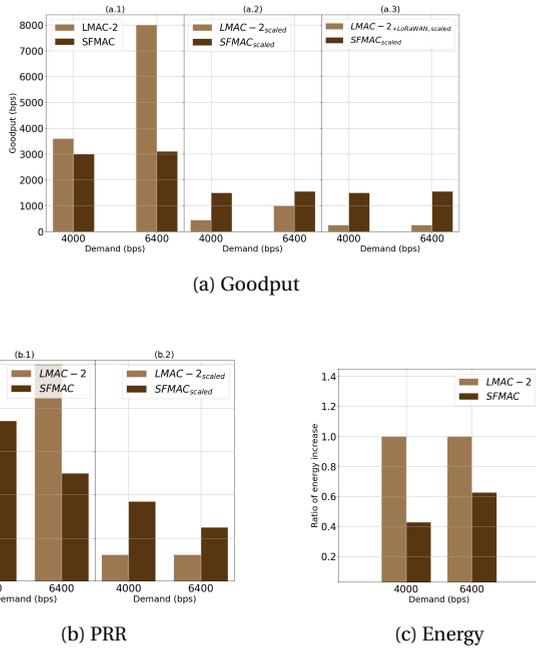


Figure 5.18: Comparison between SFMAC and LMAC.

loads increase, SFMAC outperforms CADMAC by higher factors, improving almost  $2\times$  the channel utilization for  $G=2$ .

### EFFECT OF IMPOSING FAILURES IN CAD ON THE PERFORMANCE OF SFMAC

Till this point, the success probability of the CAD mechanism in our simulations was taken to be 1. However, in a real LoRa deployment, the success rate of CAD deteriorates as distance increases per SF, as shown in subsection 2.3.2, due to attenuation caused by path loss. Further, the pseudo-orthogonality between different SFs and bandwidths leads to a non-negligible amount of false detections. Therefore in Fig. 5.19 we analyze the impact of inducing CAD errors, leading to a 6.75% downgrade of performance compared to the perfect scenario. Generally, imposing 5% and 10% CAD errors lead to a degradation of performance in channel utilization between 3.5%-4.5% and 6-7% depending on the traffic load.

#### 5.3.1. COMPARISON WITH STATE-OF-THE-ART

In this section, SFMAC is compared to three state-of-the-art CS protocols:  $p$ -CARMA [104],  $np$ -CECADA [46], and LMAC [81]. LMAC [81] proposes several versions of a MAC protocol for LoRa with CAD as a channel sensing mechanism. We will compare to LMAC-2 since it reports the best performance for class A devices. Whereas LMAC uses 16 combinations of 8 channels and 2 SFs, we use SF7, SF9, and one frequency. While LMAC uses 16 B transmissions, 50 nodes and could use in extreme cases 450 CAD attempts per device,

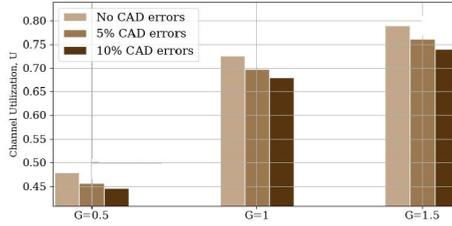
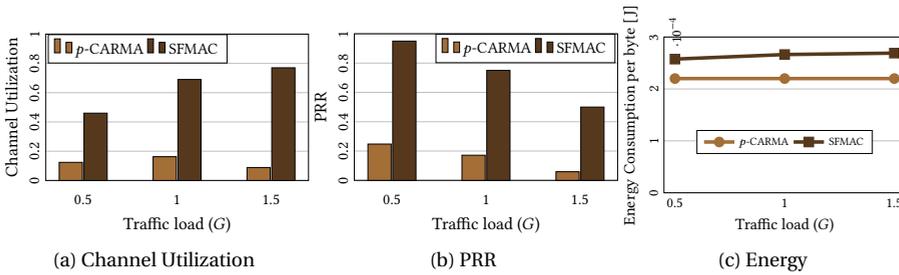


Figure 5.19: Performance of SFMAC under CAD errors.

Figure 5.20: Comparison between SFMAC and  $p$ -CARMA.

SFMAC uses 40 B payload, 500 nodes, and up to 5 CAD attempts.  $p$ -CARMA [104] and  $np$ -CECADA [46] differ from SFMAC, apart from the use of a dedicated control channel, in that they need a convergence time to reach optimal  $p$ -values and back-off times, respectively.

### COMPARISON WITH LMAC

**Goodput.** Fig. 5.18a presents a goodput comparison between LMAC-2 and SFMAC. Since LMAC uses 16 combinations of SFs and transmission channels, we scaled it down to compare to SFMAC. In Fig. 5.18 (a.3) LMAC-2 is implemented under the restrictions of current LoRaWAN specification (guard times, two receive windows, etc.). As observed, the goodput offered by SFMAC at high traffic outperforms LMAC-2. Specifically, SFMAC achieves 6.25 $\times$  higher goodput at 4 kbps. The increase in goodput highlights the potential of using a control channel under dense traffic and minimum resources.

**PRR.** Compared to LMAC-2, scaling down to consider the PRR per channel and SF, SFMAC reports an improvement of 3 $\times$  under the demand of 4000 bps, see Fig. 5.18 (b.2).

**Energy consumption.** The ratio of energy consumed with LMAC or SFMAC protocol and LoRaWAN (ALOHA) is presented in 5.18c. The overhead of SFMAC is 0.42 - 0.62 times the energy of LoRaWAN, while LMAC-2's overhead is 1.0 for the same traffic.

### COMPARISON WITH $p$ -CARMA

**Channel Utilization.** As seen in Fig. 5.20a, SFMAC offers 8.7 $\times$  increase in channel use compared to  $p$ -CARMA at a normalized traffic load of  $G = 1.5$ . The above confirms the advantage of direct channel assessment given by the control-channel than the indirect, probabilistic adaptation of  $p$ -value that takes place in  $p$ -CARMA.

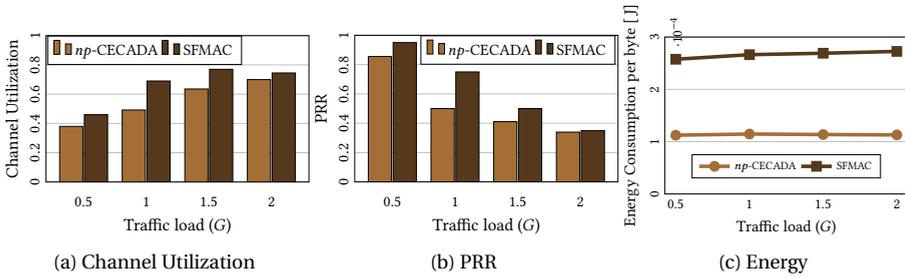


Figure 5.21: Comparison between SFMAC and *np*-CECADA.

**PRR.** Fig. 5.20b shows that SFMAC increases PRR by 8.5× at  $G = 1.5$  compared to *p*-CARMA thanks to the spread in time of backoffs and the dropping policy.

**Energy consumption.** The switch between control and data channel costs a bit more to SFMAC, but this increase in energy consumption is negligible. As observed in Fig. 5.20c, SFMAC consumes at most only 1.17× the energy of *p*-CARMA per byte, i.e., 37.5 μJ more than *p*-CARMA.

#### COMPARISON WITH *np*-CECADA

**Channel Utilization.** SFMAC outperforms *np*-CECADA too in terms of channel usage. Fig. 5.21a presents a 1.4× increase compared to *p*-CARMA at  $G = 1$ , while even at the highest normalized traffic that we evaluated,  $G = 2$ , SFMAC slightly outperforms *np*-CECADA (1.07×). The above results show that the use of a dedicated control channel outperforms the heuristic rules under which the back-off values of devices are adapted in *np*-CECADA.

**PRR.** In Fig. 5.21b, SFMAC outperforms *np*-CECADA in PRR regardless of the traffic load, with the highest increase shown at  $G = 1$ , wherein the PRR of SFMAC is 1.5× the PRR achieved by *np*-CECADA.

**Energy consumption.** *np*-CECADA, due to its back-off policy and the transmission power reduction strategy, is more economical than both SFMAC and *p*-CARMA. However, its energy savings remain negligible considering the improvement offered by SFMAC in channel utilization. Specifically, in Fig. 5.21c, it is seen that for each byte in SFMAC we consume 143 μJ of energy more than in *np*-CECADA.

#### 5.3.2. MULTIPLE GATEWAY SCENARIO

Extensive data collection of LoRa transmissions in different gateways spread in the city of Antwerp was performed by Aernouts *et al.* [102]. In order to test our algorithm in a reality deployment, we assessed the performance of SFMAC in multiple gateway scenarios, using the position of gateways in the city of Antwerp, taken from Aernouts *et al.* [102]. As expected, in topology 1 (the real locations of gateways in Antwerp) we obtain very similar results as the previous sections since there is no overlap in the circles of a radius of 500 m that is the distribution of nodes in our simulations. In order to get a closer result to what LoRaWAN deployments in smart cities will look like, where the population of gateways will increase to serve the increase of end devices, topology 2 in Fig. 5.22 halves

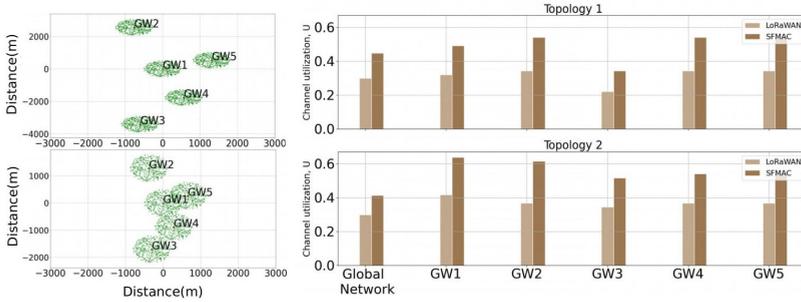


Figure 5.22: Performance of SFMAC in multiple gateway scenario.

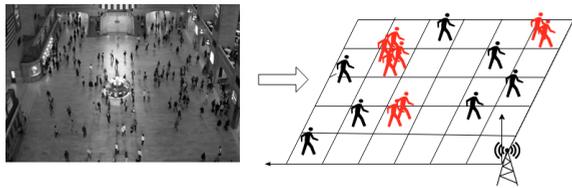


Figure 5.23: Diagram of monitoring social distancing using LoRaWAN. Red icons represent areas where social distancing is not being preserved. In those areas, pedestrians using LoRa-enabled wearable devices will transmit a frame to the closest gateway. Example of image processed [107].

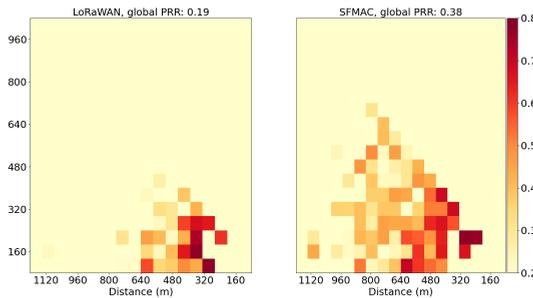


Figure 5.24: PRR per area. Most of the color map corresponds to PRR = 0 due to non-existent distance violations in the footage from the NYC Grand Central terminal.



Figure 5.25: Thirty devices distributed per five devices per group in six groups. Floor-plan of campus building with group-positions according to their SF Gateway positioned with group C.

5

the distances between gateways of topology 1. As noticed in Fig. 5.22, gateways that are not in contact with others still preserve the same channel utilization. On the other hand, networks that are closer to each other report higher traffic than usual. SFMAC still offers a 0.58 channel utilization even under the interference from close deployments.

### 5.3.3. SFMAC EVALUATION IN A PRACTICAL APPLICATION

In this real-world scenario, we build on top of an existing project [108] that has the objective of monitoring with a camera if social distancing is preserved in public spaces to avoid the spread of COVID-19. Yang *et al.* [108] use several datasets of images to perform image processing and detect violations in social distances. We will take as an example of application one of the datasets used which contains footage from NYC Grand Central Terminal. The footage was introduced by Zhou *et al.* [107]. Using the outcome of these processed images and distance calculations, we can model traffic and distribution of nodes in our scenario. Our scenario would consider instead of cameras monitoring, wearable devices tracking whether the distance between individuals has been preserved in a public space. These devices report to a gateway whenever a social distance violation is detected. We evaluate whether SFMAC can handle all the generated transmissions. In order to simulate how SFMAC would handle all the transmissions, both violations of distance per time and space have been adapted. We have gathered the number of violations happening per time unit and we have simplified the space to cover with a grid the available space. In order to use the potential of performing social distancing monitoring with LoRa, we increase the distance covered by the footage until 1 km in one dimension and we sparse the data obtained from the image processing of Fig. 5.23. Compared to deploying cameras and dealing with privacy issues because of recording in public spaces, using long-range-transmitting wearable devices could be a more efficient solution to raise awareness to keep social distance. As observed in Fig. 5.24 by using a LoRa network with SFMAC, we could cover a monitoring area of 1 km x 1 km and obtain 80% correct packet deliveries in some areas with a single gateway.

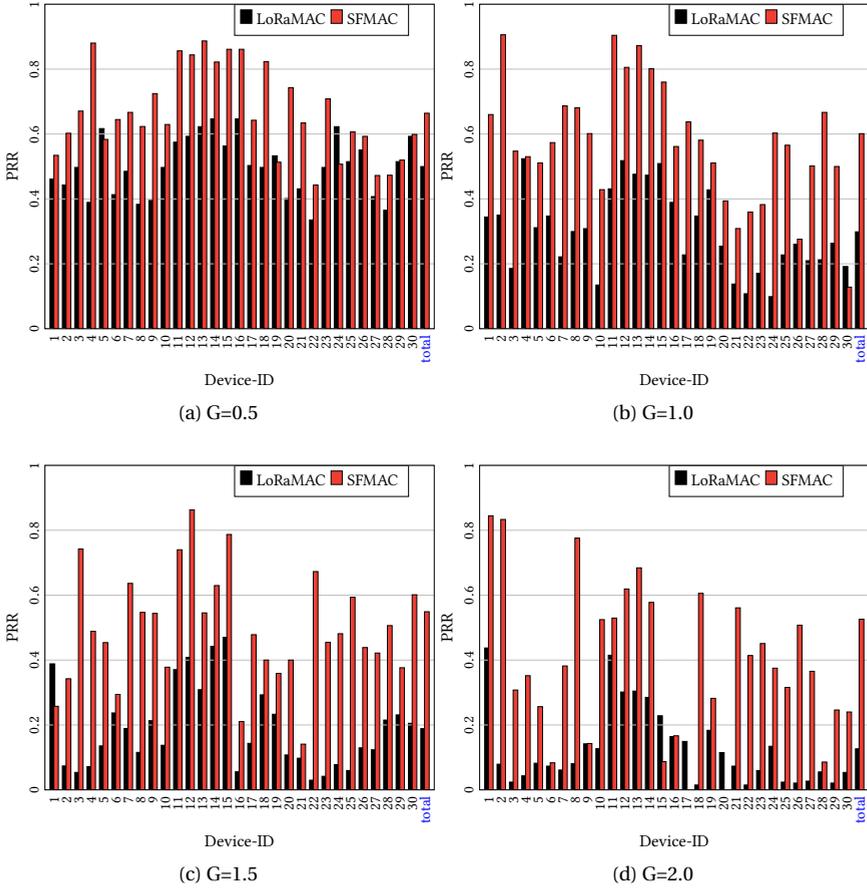


Figure 5.26: PRR per device: SFMAC versus LoRaMAC and in total for different traffic loads.

Group	Non-hidden	Group	Non-hidden	Group	Non-hidden
A	B, C	B	A, C, D	C and GW	A, B, D, E, F
D	B, C, E, F	E	C, D, F	F	C, D, E

Table 5.4: Groups of non-hidden devices for each group regarding SF7.

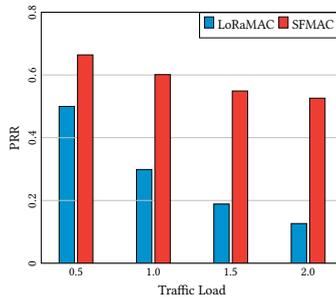


Figure 5.27: Overall PRR comparison between SFMAC and LoRaMAC.

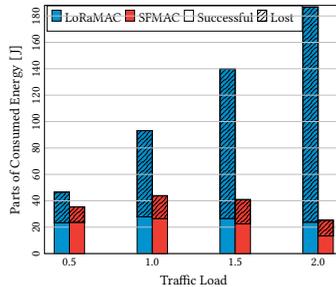


Figure 5.28: Energy consumption for SFMAC and LoRaMAC for different traffic loads. The parts of energy spent for successful and failed frames are depicted.

### 5.4. PRACTICAL EVALUATION

To evaluate SFMAC in actual LoRa deployments we used our testbed of 30 LoRa SX1261 *class A* devices (with STM32 Nucleo-F446RE MCU) and one gateway in an indoor environment, emulating a LoRaWAN without interference from commercial operations. As shown in Fig. 5.25 the 30 devices are deployed in six clusters of five devices in a campus-building. We fixed the gateway and extensively measured the RSSI and successful reception of frames as they were transmitted from the different locations, mimicking a LoRa network with hidden and non-hidden terminals. Each group is visible only by a subset of other groups in terms of SF7-CAD, see Table 5.4, but at the same time visible by every other group regarding SF9-CAD as it works as the control channel. The above environment, due to its indoor nature, introduces higher fading because of walls and signal attenuation, like an open field LoRaWAN with (non-)hidden devices positioned at large distances. To emulate the high traffic values used in our simulations we increased the duty cycle (above 1%) and frame sizes (i.e., Time on Air). We tested one hour of operation under SFMAC and vanilla LoRaWAN for  $G \in [0.5, 1.0, 1.5, 2.0]$ . Several experiments took place with frames of 220 B and transmission power of 14 dBm for SF7. The rest of the parameters remain unchanged. The PRR of each device is presented in Fig. 5.26. As observed, when SFMAC is used, at least 28 devices out of 30 for each traffic load report better results regarding PRR. Further, SFMAC distributes the transmissions fairly among the devices as shown by the lesser variations in PRR. On the contrary, the vast majority of devices employing

vanilla LoRaWAN (or LoRaMAC) stay below 40% PRR, and at high traffic loads, i.e.,  $G=2$ , half of them report almost 0% PRR. The overall PRR is presented in Fig. 5.27. SFMAC outperforms LoRaMAC by  $2\times$  for  $G=1.0$  and by  $4\times$  for  $G=2.0$ .

In Fig. 5.28 the energy consumption of the devices is depicted, showing also the parts of energy spent for successful transmissions and collided frames. SFMAC consumes energy not only to transmit but also for the employment of CAD both at the control and the data channel. In particular, LoRaMAC spends up to 4 times more energy than SFMAC for a traffic load of  $G=2.0$ , since the attempted transmissions are not regulated by the MAC layer. From this energy, the portion that is consumed for successfully received frames is 50% for  $G=0.5$  and 11.4% for  $G=2.0$ ; the rest is wasted. On the other hand, SFMAC manages 66.4% and 52.6% useful energy consumption for  $G=0.5$  and  $G=2.0$ , respectively.

## 5.5. CONCLUSIONS

By relying on the channel sensing operations to a higher spreading factor, SFMAC is the first MAC-layer protocol for LoRa networks that reveals the traffic created by hidden terminals while being distributed. Through the enforcement of structured actions – either *listen-transmit* or *listen-sleep-drop*– overlapping LoRa transmissions get aligned, generating a virtual schedule of nodes. This virtual TDMA approach on the MAC layer allows structuring the transmissions seamlessly, without requiring any change in the gateway/infrastructure.

SFMAC not only outperforms the simple CADMAC protocol, but also improves over the state-of-the-art, i.e., LMAC [81],  $p$ -CARMA [104], and  $np$ -CECADA [46], in terms of goodput, channel utilization, and PRR. SFMAC achieves two times the channel utilization of CADMAC at a traffic load of  $G=2$ , while reducing the energy wasted for collided packets by  $1.35\times$ . Further, SFMAC improves the channel utilization and the PRR over  $p$ -CARMA by up to  $8.7\times$  and  $8.5\times$ , respectively. Regarding the same metrics,  $np$ -CECADA is also outperformed by SFMAC by factors of 1.4 and 1.5, correspondingly. In addition, SFMAC achieves  $6.25\times$  higher goodput than LMAC at 4 kbps. Finally, in real-world experiments with our LoRa nodes involving traffic loads up to  $G=2.0$ , SFMAC outperforms vanilla LoRaWAN regarding PRR by  $4\times$  at  $G=2.0$ , while vanilla LoRaWAN consumes 4 times more energy.

With SFMAC, we conclude our contribution towards the improvement of scalability of LoRa networks through the MAC layer. In the following chapter, we delve into the application layer, wherein we will propose coding mechanisms that improve LoRaWAN's scalability by recovering information from corrupted and lost LoRa-frames.



# 6

## APPLICATION LAYER CODING TECHNIQUES FOR THE RECOVERY OF DATA FROM LOST AND CORRUPTED LoRa FRAMES

SINCE the communication range of LoRaWAN is very high and the power limited because of the use of unlicensed spectrum, the corrupting effect of multipath propagation, shadowing, and scattering is predominant. In Non-Line of Sight (NLoS) conditions urban clutter intensifies these phenomena. The above lead to a very low Received Signal Strength Indicator (RSSI) for the LoRa frames, *e.g.*, -115 dBm. Further, by operating at the ISM bands LoRaWAN shares spectrum with other contending networks, which interfere with LoRa-frames, and their transmissions corrupt LoRa-symbols. Also, the Medium Access Control (MAC) layer of LoRaWAN is unslotted and Aloha-like. This aggravates further the issue of corrupted frames since collisions are increased due to the absence of channel sensing. Therefore, invariably LoRa transmissions often suffer from symbol corruption including bursts of symbols. This leads either to frame losses or renders the received LoRa-frames useless.

LoRaWAN targets to serve up to 15,000 devices per gateway. Acknowledgment (ACK) mechanism is an option in the LoRaWAN protocol [73, 66]. However, ACK requires downlink communication and also introduces communication overheads. With a large number of devices in a LoRa network, the transmission of ACKs for each packet seems improbable due to the duty-cycle limits (1% for gateways in Europe). Also, the gateways are typically half-duplex, which will lead to frame losses in the uplink communication while ACKs are transmitted [55]. Data from The Things Network (TTN) [103] confirms that the utilization of ACKs is evaded. Specifically, out of the total number of received frames, only 3.47% were received using ACKs, while the rest of frame-receptions resulted from single unacknowledged transmissions [109]. The above data confirm the study

performed by Augustin *et al.* [110] on the properties of LoRa and LoRaWAN, which states that acknowledged messages significantly reduce the successful throughput in a LoRa network and should, therefore, be avoided. Further, even if the ACK-mechanism was more actively employed the devices would retransmit if ACKs were not received, increasing the traffic in the network, and leading to increased interference, which in turn would result in even more corrupted frames. Accounting for the above observations and expecting a large portion of the projected 41 billion IoT devices by 2027 [4] to use LoRaWAN, recovering LoRa-frames is highly critical –not only to preserve the battery life of the already restricted LoRa-devices, but also for the efficient utilization of the spectrum. To address the previous point, we design coding schemes for efficient data recovery, i.e., increasing data reception while trading off energy consumption. Our schemes must not require any change in the gateways since such a solution is expensive for existing deployments, but should operate at the application layer such that the solution can work with already deployed LoRa networks. Also, any introduced byte redundancy due to encoding should be minimal, especially for high SFs, because the allowed payload size is limited (i.e., 51 B for SF10-SF12).

The preamble-part of a frame is more important because if it is corrupted the frame is completely lost [65]. Therefore, LoRa channels are generally of erasure nature, i.e., frames are either received correctly or lost. However, Rahmadhani *et al.* reported 32% of transmitted SF8-frames being received corrupted [34]. Thus we conducted preliminary research regarding both frame loss and frame corruption. Specifically, we found that up to 53% of the frames were lost at distances of 6 km between devices and GWs [51]. However, the vast majority of the frames were transmitted at SF12 (see subsection 6.2.2). Regarding frame corruption, we found even 50% of transmitted frames at SF8 being corrupted when they were received under specific SNR-ranges in real-world LoRa deployments [52]. Therefore, we delved deep into the characterization of frame corruption under various scenarios in order to understand which receiving conditions favor corruption and which lead to frames being lost. We observed up to 49.72% and 29.06% of frames being received corrupted at SF8 and SF10, respectively (more details in subsection 6.3.1). To recover the vast amount of data lost/corrupted, we designed coding mechanisms for data recovery for both cases: (i) Data Recovery (DaRe) for frame loss (see Section 6.2) and (ii) Divide & Code (DC) for frame corruption (see Section 6.3). Our mechanisms operate on the application layer of LoRaWAN.

## 6.1. RELATED WORK

Before we proceed, we present important works on the fields of block codes (fountain codes) and convolutional codes, since both DaRe and DC utilize their mathematical frameworks and operational mechanisms which we will briefly mention in this section, and analyze in detail in their corresponding sections (see Section 6.2 and Section 6.3). Then, we discuss notable coding schemes for LoRaWAN. There are two types of channels regarding frame reception: (i) channels in which the corrupted frames are lost, i.e., erasure channels and (ii) channels in which the corrupted frames are received with their corrupted symbols included, i.e., error channels. In LoRaWAN, both cases exist, having more corrupted frames in low SFs and/or at specific SNR-ranges, while observing most frames being lost in high SFs and/or below these SNR-ranges, see Section 6.3. DaRe

focuses on erasure LoRa-channels [51], while DC operates on error LoRa-channels.

In communication theory, majorly, two methods exist to deal with frame loss/corruption: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) [111]. The former employs error detection schemes in communication but relies on frame retransmission to correct errors. In FEC, redundant data is transmitted that helps in detecting and correcting a limited number of errors. Erasure/frame loss coding deals with frame loss channels. We also find frame loss coding in distributed data storage, where information is stored redundantly over different disks or servers. This allows the loss of storage media while preventing data loss. In applications communicating over packet-switched networks, like media streaming [112], frame loss coding is employed to handle packet loss without data loss. The earliest codes used for erasure channels are Reed-Solomon codes [113, 114], introduced in 1960. Reed-Solomon (RS) codes are, for instance, found on CD-ROMs, to withstand physical damage to the disk. RS-codes require complex computations and are resource expensive [115]. In 1962, Gallager developed the concept of Low-Density Parity-Check codes (LDPC codes) [116]. With LDPC codes, data blocks are supplemented with parity bits before transmission to enable error/frame loss detection and correction. LDPC codes are, for instance, used in the current DVB-S2 standard [117]. RS-codes and LDPC codes provide error/frame loss detection and correction.

**Fountain Codes.** In Fountain codes, error detection is presumed given. Thus, the coding scheme is designed entirely for erasure correction. In fountain codes, codewords are calculated as linear combinations of data fragments [118]. The principle of a fountain code is that any random combination of sufficient codewords can be used to decode the information [115]. Fountain codes were introduced by Luby [50]. Examples of fountain codes are LT-codes [119], Raptor codes [120] and Online codes [121]. In fountain codes, the idea of producing an infinite number of code words, instead of a fixed number, was introduced, i.e., rateless codes. Fountain codes are block codes, like LDPC codes. The mathematical principles of fountain codes can be used to recover data loss in LoRaWAN. The content of frames can be viewed as data fragments. Linear combinations of these data fragments can be transmitted as well to provide redundant information. The redundant information needs to be spread over multiple LoRaWAN frames.

**Convolutional Codes.** An alternative to block codes are convolutional codes, which encode with a sliding window. This means the dataset used to calculate codewords changes over time, while for a block code the dataset is constant. An example of a convolutional code is the class of Turbo codes [122]. Turbo codes are iterative, with messages being sent along with parity bits computed in a recursive manner. The principle of a sliding window can be applied to fountain codes, making a convolutional-fountain code. There are already some works on combining fountain codes and convolutional codes [123, 124]. Also, there has been research done on applying fountain codes in a windowed manner [125]. These existing studies do not apply fountain codes on a sliding window in a convolutional manner.

**LoRaWAN coding schemes.** Montejo-Sánchez *et al.* transmit the encoded redundancy in independent frames, whose number is decided based on range, configuration, and QoS requirements [126]. Elshabrawy and Robert proposed non-binary Single Parity Check (SPC) codes with soft-decision decoding, trading off the increase in coding gains with increased Time on Air (ToA). Optimal application of SPC code rate enhances the capacity

of LoRa networks up to 65% [63]. Further, they showed reduction in Bit Error Rate (BER) by applying Bit Interleaved Coded Modulation (BICM) considering Rayleigh fading and AWGN channels and gained up to 8 dB in BER [127]. Coutaud *et al.* design LoRaFFEC wherein frames are encoded using pseudo-random linear combinations of already sent data. Combined with data-fragmentation –to cater to the variety of LoRa-frame sizes– LoRaFFEC manages a data delivery ratio of 98% for channels with a 0.4 probability of frame error [128]. Further, Coutaud and Tourancheau propose CCARR, which encodes frames using RS-coding, adapting the size of the added redundancy dynamically [129]. Borkotoky *et al.* suggest windowed and selective coding at the application layer of delay-intolerant LoRaWANs with minimum feedback. Windowed encoding accounts for all the non-delivered and non-expired transmitted symbols, while selective mechanism chooses a few among them according to feedback [130]. Wang *et al.* focus on convolutionally encoded frames received on high SNR values. They design algorithms assisted by the outcome of CRC: (i) Partial Iterative Decoding-Detecting (P-IDD) retrieves possible errors on bits with the highest log-likelihood of being erroneous, (ii) Soft-Decision Syndrome Decoding (SDSD) technique identifies patterns of errors in frames [131]. Sant’Ana *et al.* propose a hybrid coding scheme, comprised of packet replication and the use of linear XOR operations for the extension of battery life [132]. Chen *et al.* employ Luby codes (LT) on multiple versions of the same frame received by several gateways to recover the correct parts of the frame [133]. Angelopoulos *et al.* apply rateless encoding on frames of  $k$ -symbols before transmission. Upon reception, they evaluate the algebraic consistency of frames by applying the Algebraic Consistency Rule (ACR) on  $k + 1$  symbols. Any corrupted symbols found are recovered through iterative decoding assisted by CRC [134].

## 6.2. MEMORY-BASED CODING FOR DATA RECOVERY FROM LOST LoRa FRAMES

In this section, we focus on frame loss. To recover data from lost frames, we propose a novel application layer coding technique called *Data Recovery*, DaRe, based on convolutional and fountain codes. Our contributions are the following:

- To study and characterize frame loss, we perform real-world measurements with a LoRa network. We have collected data extensively (around 23,000 frames) over several days in stationary and mobile scenarios. We characterize both spatial (i.e., frame loss over distance) and temporal (i.e., burstiness of frame loss) properties of the channel using the datasets. With the collected data, we observe that there is a significant amount of frame loss that occurs as the end-device moves farther away from a gateway. We find that the channel can be bursty even when the end-device is stationary.
- We design DaRe, a coding technique that does not intend to recover the lost frames, but it enables the recovery of the data from lost frames using FEC at the application layer. DaRe bases its mechanism on the application of AND and XOR operations among previous frames in order to produce the redundancy that is added to the currently transmitted frame. A sliding window dictates the number of previous frames to be considered for the encoding.

- We compare DaRe to classic LoRaWAN repetition coding and to Luby Transform (LT) codes, which are widely used fountain codes. When a bursty channel is used, the Data Reception Rate (DRR) of DaRe is reduced at most 1.4%, which is significantly less than the 18% of the loss added when classic LoRa-FEC is applied. Compared to LT codes, DaRe improves DRR up to 2.29 times.
- We develop an implementation of DaRe for LoRaWAN. DaRe can recover up to 99% of the data when frame loss is up to 40% with a code rate of 1/2. Compared to a naive repetition coding method, DaRe reduces energy requirement up to a factor of 0.42.

### 6.2.1. SETUP AND DATA COLLECTION

Before proceeding to the design and the evaluation of coding schemes we must characterize the LoRa-channel in terms of frame loss. In this part, we describe our data collection setup, scenarios, method, and the datasets collected for the analysis of the LoRaWAN communication channel.

#### MEASUREMENT SETUP

**Network** While the measurement devices adhered to the duty cycle limits, they were exempted from any usage limits on the number of messages. We use Thingpark Wireless Logger [135] to log the LoRaWAN frames received by the network server. Wireless Logger stores the payload and metadata of each received frame. Multiple gateways can receive a single frame. However, only the three gateways that received the LoRaWAN frame with the most reliable RSSI can be identified in the logs. This alleviates false negatives (assuming a gateway did not receive a frame, while in reality, it did), as the strongest three receiving gateways per dataset are used. All the gateways in the LoRaWAN network are situated at an average height of 27 m. The antennas have a gain of 11 dBi. The gateways are positioned an average of 8 km apart. Furthermore, the maximum distance between an end-device and its closest gateway is 7.5 km as the gateways are carefully positioned, as shown in Fig. 6.1.

**End-Device** The end-device used for data acquisition is a Sodaq Mbil Rev. 4 [136]. The antenna has a gain of 2 dBi<sup>1</sup>. The device operates in *class A*. To characterize the LoRaWAN communication channel spatiotemporally, measurements were done at multiple locations. The final version of the device included a GPS module, so the coordinates were sent in the frame payload itself. The sub-band and bandwidth were set to comply with ETSI standards [137]. The bandwidth was 125 kHz for all transmissions. Furthermore, we set the transmission power to the maximum allowed value of +14 dBm. The SF is controlled by the ADR mechanism since the device does not support a fixed value. However, the vast majority of the frames (~95%) were transmitted using SF12. Only these frames have been used for data analysis. The SF and the coordinates of the unsuccessful frames were interpolated on a straight line between the neighboring received frames. The SF for the missing frames was taken to be the lowest SF of the neighboring received frames.

<sup>1</sup>A gain of 2 dBi antenna is within the European regulations, which states that the maximum effective radiated power should be 14 dBm [137, 138]

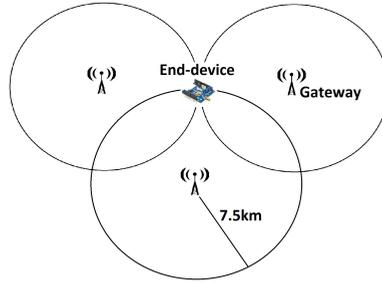


Figure 6.1: Gateways are deployed such that the maximum distance for an end-device from its closest gateway is 7.5 km.

### DATA COLLECTION SCENARIOS

We identify two scenarios for our data collection: data from stationary locations and data when the end-device is moving (mobile data). All the data was collected with the end-devices placed next to a window or outdoors.

**Stationary data** The stationary datasets were generated with the end-devices transmitting mostly at every 10 minutes or 15 minutes. A small part of the dataset contains frames transmitting at 15 s interval. During measurements, the devices were in a fixed position and orientation. The stationary dataset (~18,000 frames) was collected at different houses (locations) in the city of Delft, the Netherlands, which forms a typical urban scenario. While the gateways were positioned to be for LoS coverage, the urban clutter cannot guarantee LoS transmissions at all locations. The set of rich data captures the complex propagation environment in urban scenarios, including NLoS transmissions.

**Mobile data** The mobile datasets were collected positioning the end-device on a bicycle and a car of an average speed of 22 km/h and 80 km/h, respectively, driven for approximately 300 km and 350 km in total, respectively. The selected terrain was flat. Among the 5,000 LoRaWAN frames contained in the mobile data, approximately 80% were transmitted from rural areas and the rest were sent from an urban area which included NLoS transmissions.

### 6.2.2. FRAME LOSS CHARACTERISATION

Before we present our coding scheme, we first provide an overview of the analysis of the datasets in order to characterize the frame losses. Note that the characterization presented here also holds for *class B* and *C* end-devices for the uplink channel.

#### FRAME LOSS OVER DISTANCE

We use the bike dataset to determine the parameters of the path-loss channel model with shadowing [139] between the sender (end-device) and the receiver (gateway) since the dataset provides measurement points at different distances. The Doppler effect is negligible because of the relatively low speed. Since the bike data is collected from all different routes, we can negate the effect of permanent shadowing and other effects due to surroundings.

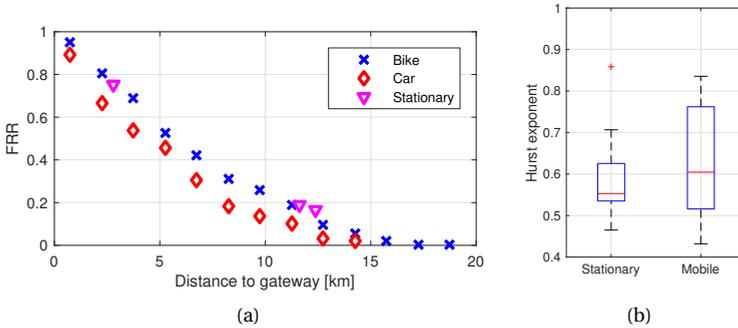


Figure 6.2: Frame loss characterization: (a) Frame Reception Ratio (FRR) as a function of the distance to a gateway for bicycle, car, and stationary datasets. (b) Box plot quantifying the burstiness in the two different dataset types using the Hurst exponent.

Through analyzing the dataset, we estimate the path-loss exponent to be 2.71, and the shadowing follows the log-normal distribution with a mean of 0.56 and a standard deviation of 7.11. With these parameters and the receiver sensitivity of the gateway ( $-137$  dBm), we can estimate the cell outage probability [139]. The outage probability for the farthest distance of 7.5 km is found to be 0.004, indicating that the coverage of the current deployment is sufficient. While the frame loss characteristics would be typical as reported here, the path-loss exponent and the shadowing distribution may be affected by several factors including gateway location, gateway height, antenna gains, and terrain. A method to determine optimal deployment using satellite imagery has been proposed by Demetri *et al.* [140].

To characterize the frame loss or erasure of frames, due to the channel effects, we analyze the data from all our datasets. We first consider frame loss as a function of the distance between the end-device and the gateway. To account for location estimation inaccuracies, we consider bins of 1.5 km in which we calculate the average Frame Reception Ratio (FRR). Fig. 6.2a shows the FRR with respect to the distance of the end-device from the gateway. It is evident from the figure that more frames are lost as this distance increases. While the outage probability is quite low at 7.5 km, the frame loss is significant at that distance: almost 70% of the frames are lost. Although this seems counter-intuitive as the coverage was found to be adequate theoretically, the theoretical calculations were based on a simple path-loss model without considering the complex propagation environment. We can observe another interesting aspect in this figure because the datasets are collected at different moving speeds. The dynamics of the channel are pronounced when the end-devices are mobile due to varying channel fading and multipath fading, which is the main reason for the increased losses as the end-devices move at higher speeds, especially for higher SFs (due to longer airtimes).

**Finding #1:** Frame loss is quite significant in LoRaWAN despite an almost collision-free channel.

**Finding #2:** Frame losses are higher due to the dynamics of the channel when the end-devices move at high speeds.

### BURSTINESS

Burstiness is a temporal property of the channel. The channel shifts between poor and good states with a correlation between the frame delivery events. This results in frame losses being closer to each other for a bursty data stream compared to a non-bursty data stream. While burstiness in wireless links has been studied extensively, a metric to express burstiness is not standardized. We take a simple and effective measure to quantify the burstiness, using the *Hurst exponent*, or the self-similarity parameter,  $H$  [141].

This exponent is used to reveal self-similar streams of data, i.e., data patterns over smaller intervals, which are also present over longer intervals. Considering a binary time-series, data sets without burstiness will show  $H \sim 0.5$ . The burstiness increases with an increase in the Hurst exponent, and if  $H > 0.6$ , a dataset is characterized as bursty. In Fig. 6.2b, a box plot of the calculated values for the Hurst exponent for the different types of datasets (stationary and mobile) is shown. We observe that the mobile data sets show more burstiness on average ( $\overline{H_{bike}} = 0.6$ ) and higher deviation than the stationary data sets ( $\overline{H_{stat}} = 0.56$ ). Burstiness in the mobile dataset is due to mobility, while burstiness in stationary datasets is the result of channel effects.

**Finding #3:** LoRaWAN channel is also prone to lose frames in bursts due to channel effects and mobility.

For the design of the data recovery method, it is essential to know if burstiness occurs. If burstiness is present, the data recovery method should spread redundant information relatively more over time. Around half of our datasets show uniformly distributed frame loss. We use the Bernoulli channel model with frame loss probability  $p_e$  for these datasets. For the datasets that show burstiness, we use the Gilbert Elliot model [142, 143, 144]. The channel model has an average frame loss probability  $p_e=0.46$ , which was found empirically.

### 6.2.3. DARE: DATA RECOVERY IN LoRaWAN

DaRe is an application layer coding technique, which is a combination of both convolutional and fountain codes. To the best of our knowledge, DaRe is the first such coding technique. DaRe can be applied to any application transmitting over a lossy transmission medium, however, with some fine-tuning. DaRe has been tuned for the unique properties of LoRaWAN. The main constraints are due to the packet size limitations imposed by the SFs (specifically for the high SFs, 10, 11, and 12), and the regulations on the duty cycle. Most applications of LoRaWAN use the *class A* mode in which an end-device transmits its sensor data periodically but infrequently using an ALOHA-like protocol. We refer to the sensor data that needs to be transmitted as a *data unit*. Since frame loss leads to data loss, frame loss must be minimized.

For data recovery, an existing set of data units should be extended with redundant information, such that the original set of data units can be recovered even if only a subset of the transmitted data units is received. LoRaWAN operates on a frame loss channel, i.e., frames are either received or wholly lost. Thus, DaRe must spread the redundant information from the data in one frame across other frames, so a lost frame can be recovered using redundant information from other frames. The redundant information included in a frame is a parity check of randomly selected previous data units. A parity check is a vector of parity bits for each bit position in the data units. Traditional fountain

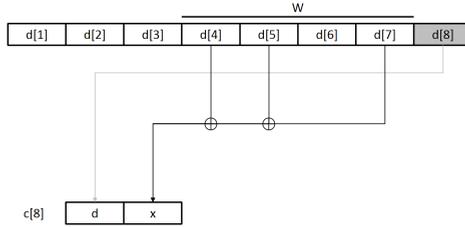


Figure 6.3: A schematic explanation of DaRe. In this example,  $R=1/2$ ,  $W=4$ ,  $\Delta=0.75$ . The code words for the frame at time instance  $t = 8$  are calculated by concatenating the data units from  $t = 8$  and a parity check of previous data units from  $t = 4, 5, 7$ .

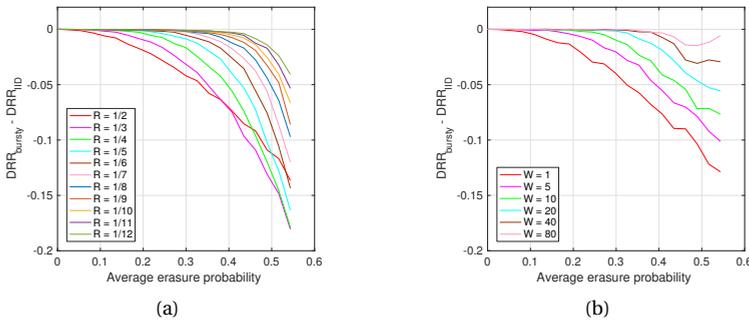


Figure 6.4: Difference in DRR over a Gilbert Elliot channel model compared to an IID channel, with the Gilbert Elliot model parameters  $(p_{GB}, p_{BG}) = (0.25, 0.21)$  varying  $p_{loss}$  and plotted for  $p_e$ . (a) For the benchmark. (b) Using DaRe with different values for  $W$  and  $R=1/2$ .

codes perform the coding over a data block. But since we want the redundant information to be calculated in a convolutional manner, we use a sliding window approach with a finite window. Since we intend to keep the complexity low for embedded devices, we work only with Galois Field 2 (GF(2)). This implies that the multiplications and additions are bit-wise ‘AND’ and ‘XOR’ operations, respectively. A schematic explanation of determining the frame payload using DaRe is shown in Fig. 6.3.

We will not delve into a detailed presentation of the coding parameters and the mathematical framework of DaRe. For more information, see the earlier work of Marcelis *et al.* [49]. There are three parameters in the context of DaRe: code rate ( $R$ ), window size ( $W$ ) and degree ( $\Delta$ ).

**Code rate** The code rate ( $R$ ) is the ratio between the size of original data and the size of the data actually transmitted. It expresses the amount of redundant information added in transmission.

**Window Size** The window size ( $W$ ) expresses the number of previous data units to consider for calculating the redundant information.

**Degree** The degree ( $\Delta$ ) expresses the relative number of previous data units from the selected window to include in the parity check. For example, if  $W=10$  and  $\Delta=0.5$ , there will be 5 data units –randomly chosen– included in a parity check.

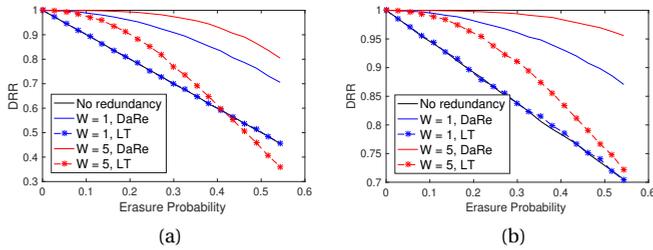


Figure 6.5: Comparison of DaRe with LT codes. (a) For a regular channel. (b) For a Gilbert channel.

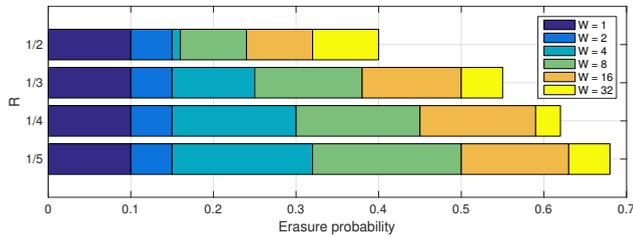


Figure 6.6: For a given code rate  $R$  the window size needed to recover 99% of the data for a certain frame loss probability.

**BENCHMARK: CONVENTIONAL REPETITION CODING AND LT CODES**

A simple form of frame redundancy is repetition, i.e., to append previous data units to a frame. This method provides some redundancy and allows for recovery, but has a lot of overhead. We use this coding method as a benchmark for DaRe as it provides a performance reference. Additionally, we also compare the performance with Luby Transform (LT) codes [50], which are widely used fountain codes. The LT-codes have been adapted to operate with finite windows for a fair comparison.

**6.2.4. NUMERICAL RESULTS**

We perform a numerical evaluation with the mathematical framework in MATLAB to compare DaRe with repetition coding algorithms, used with LoRaWAN. We observe in Fig. 6.4a that burstiness impacts the coding method used in the benchmark quite significantly, with an additional loss of up to 18% for higher values for the code rate. Compared to DaRe, shown in Fig. 6.4b, DaRe offers much better resilience. For a window size of  $W=80$ , the maximum performance reduction is 1.4%. Therefore, we can conclude that DaRe can handle both bursty and non-bursty frame losses equally well.

In order to compare, we chose Luby Transform Codes [119], which is one of the most famous fountain codes, and modified to operate over a given window size ( $W=1$  and  $W=5$ ). The method of implementation of these modified LT-codes is as follows: The LoRaWAN nodes send sensor data at a pre-determined periodicity. This data is also stored in a buffer of size  $W$ . After the window is full, LT-encoding is performed. For the next  $W$  packets, a block of encoded data is sent along with the sensor data. The blocks are collected at the

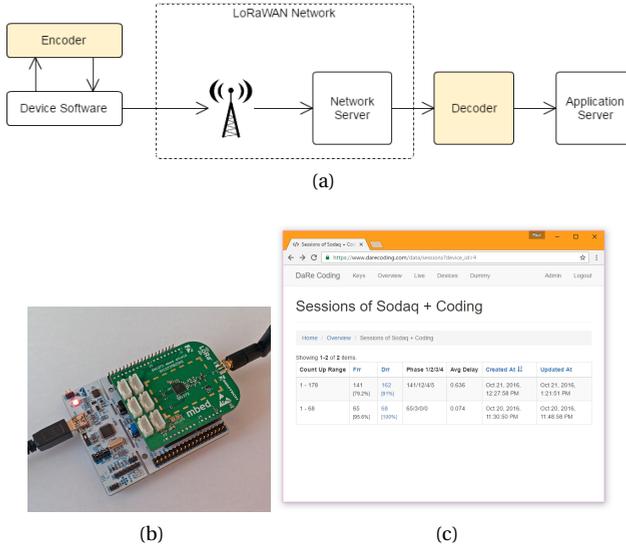


Figure 6.7: Implementation. (a) Software architecture. The encoder is implemented as a library that can be included in the device software. The decoder is a module between the Network Server and the Application Server that receives frames and outputs data units. (b) A second device on which the encoder was tested. (c) A screenshot of the web-based decoder. In the screenshot, two sessions of a device using DaRe are shown. The DRR is higher than the FRR as a result of the data recovery by DaRe.

receiver and then are passed on to the decoder. If the decoding is successful, then all the packets from the previous window can be recovered. The results of the evaluation for regular and Gilbert channels are shown in Fig. 6.5a and Fig. 6.5b. We observe that DaRe outperforms the convolutional LT codes significantly, especially for high erasure probabilities, wherein DaRe improves DRR up to 2.29 times for the Gilbert channel.

### 6.2.5. PRACTICAL EVALUATION

We will pass quickly through the implementation details of the DaRe encoder/decoder on end-devices and application server, as it is based on the previous work of Marcellis *et al.* [49]. Instead, we will focus on the evaluation results that we found based on this implementation.

#### IMPLEMENTATION

The two main parameters of DaRe,  $R$  and  $W$ , should be chosen, and  $\Delta$  is pre-computed with the chosen value for  $W$ .  $R$  and  $W$  are limited to the constraints set by the payload size and end-device memory respectively. By using the results in Fig. 6.4, the parameter settings providing a desired DRR for the expected frame loss probability and burstiness can be determined. In order to make it easy for interpretation, we present the results in Fig. 6.6. Using the expected frame loss, Fig. 6.6 can be used to pick the values for  $R$  and  $W$  that recover the expected frame loss up to 99%. As observed DaRe recovers 99% of the data with an erasure probability of 40% with a code rate  $R$  of 1/2. A static choice of parameters may offer varied data recovery performance depending on the long-term changes in

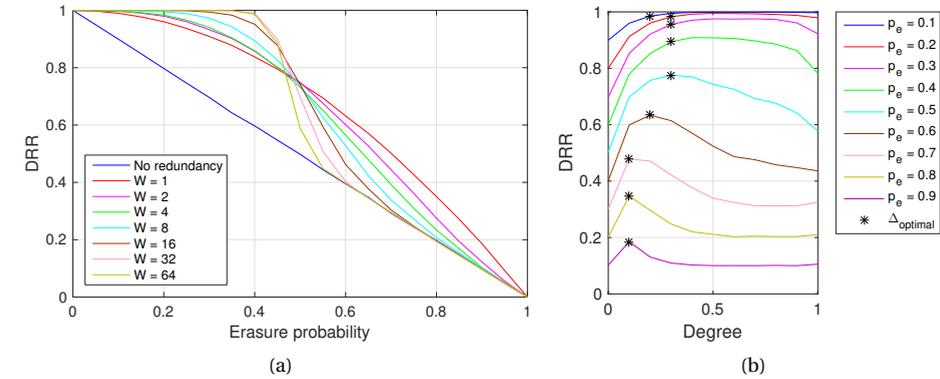


Figure 6.8: Data recovery ratio (DRR) in emulation results over an IID channel. (a) DRR for emulation of the decoding algorithm for different window size  $W$  and  $R=1/2$ . (b) Data recovery ratio (DRR) for  $R=1/2$  and  $W=10$  for different frame loss probabilities when varying  $\Delta$ . The black crosses mark the values for the optimal degree  $\Delta_{\text{optimal}}(p_e)$  that gives maximal recovery rate.

the frame loss of the channel. One way to adapt these parameters is to use a downlink LoRaWAN frame to notify the end-device about the average channel condition in the recent past. Typical LoRaWAN deployments also know the device locations. The downlink frame may contain the observed average frame losses from the device and nearby devices in the recent past allowing the end-device to dynamically adapt the parameters in runtime to provide optimal performance for the new circumstances.

The encoder and decoder can be implemented in respectively a LoRaWAN end-device and a LoRaWAN application server. In Fig. 6.7a a schematic overview of the software architecture for the system integration is given.

The **encoder** is implemented as a C++ library that is included in the device code. The inputs for the encoder function are the coding parameters  $R$  and  $W$ , and the sequence of data units to transmit. To test the library, it has been implemented on two devices: the same device used for data collection (see subsection 6.2.1.1) and another device with the same specifications, which is shown in Fig. 6.7b.

The **decoder** is implemented as a parser between the network server and the application server. The decoder takes frames with DaRe payload and converts them into data units. For emulation, the decoder has been implemented as C++ code, but for the final deliverable, the decoder has also been implemented in a web server. The web-based decoder is designed to work for multiple devices. The network server only needs to be configured to forward all the frames to the HTTPS endpoint of the decoder, and it will decode for all devices. A screenshot of the interface of the web-based decoder is given in Fig. 6.7c.

### EVALUATION RESULTS

We evaluate our implementation of DaRe in the following ways: (i) emulating an IID channel between the end-device and the application server and (ii) using our measurements from applying DaRe in-field.

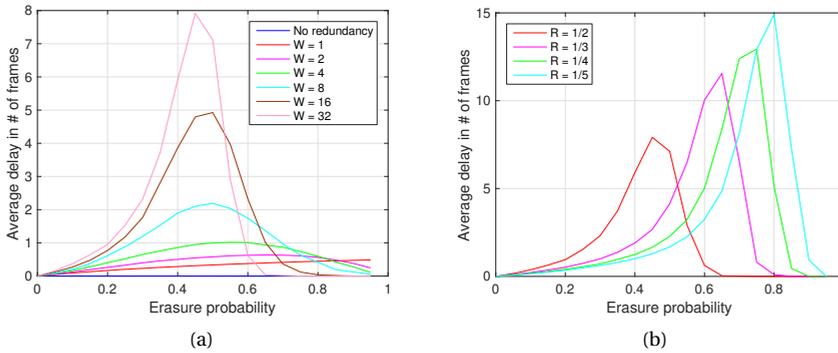


Figure 6.9: Average recovery delay in emulation results of an IID channel. (a) Recovery delay for different window size  $W$  and  $R=1/2$ . The maximum average delay is 7.9. (b) Recovery delay for different code rates  $R$  and  $W=32$ .

**Data Recovery Ratio** Fig. 6.8a shows the DRR for various window sizes for code rate  $R=1/2$ . It can be seen that the DRR for  $W=64$  is always lower than that for  $W=32$ . Thus  $W=64$  does not offer any advantage over  $W=32$ . Therefore,  $W=32$  is set to be the maximal window size. 99% of the data units can be recovered for channels with frame loss of up to 40%. For larger values of  $W$  the parity checks contain a larger absolute number of data units, due to the determined degree,  $\Delta$ . With a larger number of data units in a parity check, the chance of having non-reducible parity checks is higher for increasing frame loss probability, resulting in lower data recovery. This explains the curves crossing over each other in Fig. 6.8a. To reduce the chance for non-reducible parity checks, the degree  $\Delta$  should decrease. As can be seen in Fig. 6.8b, higher DRR is reached for higher frame loss probabilities when using a lower degree than 0.4. Since it is more important for the DRR for lower frame loss probabilities to be close to 100% –as these  $p_e$  values are often found in real-world scenarios– lower degrees for higher frame loss probabilities (above  $p_e=0.5$ ) are not relevant.

**Data Recovery Delay** The data recovery delay is defined as the number of additional frames needed to be received before a data unit is recovered. Delay could be a factor for a LoRaWAN application, requiring the coding parameters to be adapted to minimize the delay. While larger window sizes result in higher DRR as shown in Fig. 6.8a, the average delay increases as well, as can be seen in Fig 6.9a. A larger frame loss probability introduces a longer delay. However, at the point that the DRR starts to decrease rapidly the delay decreases as well. This is due to the fact that some parity checks are left unsolved, reducing the DRR but also the delay. The maximum average delay difference between  $W=16$  and 32 is 3.1 frames. For lower code rates, the top of the average delay graph increases and the maximum average delay increases as well, as can be seen in Fig. 6.9b, since lower code rates can lead to high DRR even for higher erasure probabilities (see Fig. 6.6), i.e., more time is needed before starting to leave parity checks unresolved.

## MEASUREMENT RESULTS

To evaluate DaRe for real-life datasets, we have (a) applied DaRe to the previously collected data, and (b) performed some measurements with the end-device running DaRe. We

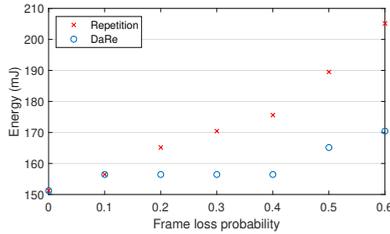


Figure 6.10: Comparison of energy consumption of repetition coding and DaRe for achieving 99% DRR for SF7 on an SX1276 radio.

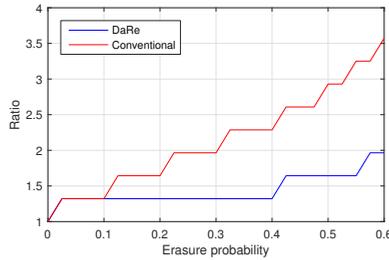


Figure 6.11: Transmission time increase for a 10-byte data unit for the desired DRR of 0.99, relative to the transmission time for uncoded data transmission. DaRe provides significant transmission time reduction.

focus on energy consumption by DaRe compared to conventional coding.

Adding redundancy in communication, like with DaRe, requires transmitting more bytes. The largest contributor to energy consumption on an end-device is frame transmission. Sending more bytes leads to significantly more energy consumption than additional computations. The impact of DaRe on the energy consumption of the end-device can be determined by calculating the additional transmission time needed. The airtime of a LoRaWAN frame can be calculated using the formulae given by Semtech [145], and the power consumption of SX1276 is obtained from Sodaq Mbili [136]. Multiplying the power consumption and the airtime yields the energy consumed for the transmission. A detailed energy consumption calculation method is outlined in the work of Bouguera *et al.* [146].

Fig 6.10 shows the energy consumption for both repetition coding and DaRe for SF7 on an SX1276 radio with standard parameters (13-byte header and coding rate of 4/5) for recovering 99% of the data. The energy consumption calculation includes energy consumed for all the phases from the radio waking up to the radio being turned off, assuming one reception window. For a frame loss probability of 0.6, DaRe consumes only 83% of the energy of repetition coding to achieve the same results. Note that when higher SFs are used the transmissions require more energy to take place due to larger air times. Therefore, the energy savings provided by DaRe will be higher. While Fig. 6.10 shows energy consumption specifically for SF7, we provide an SF-agnostic comparison in Fig. 6.11.

Fig. 6.11 shows the ratio of transmission time for DaRe and repetition coding when 99% data recovery is desired, compared to the transmission time when sending data

without coding. DaRe reduces the additional transmission time compared to repetition coding up to 42% for a data unit size of 10 bytes. When no coding is used for frame loss probability of 0, both DaRe and repetition coding takes the same amount of airtime. However, when the frame loss probability increases, redundancy must be added. When redundancy is added, either by repetition or DaRe coding, data can be recovered for a range of frame loss probabilities. For instance, with  $R=1/2$ , and  $W = 32$ , DaRe can recover 99% of data for up to 40% frame loss as shown in Fig. 6.6. Note that these parameters have been rightly chosen ( $R=1/2$ ,  $W=32$ , and  $\Delta=32$ ). This results in the constant transmission time up to 0.4 in Fig. 6.11. Similarly, for repetition coding, the redundancy introduced will be able to recover data for a range of erasure probabilities, which turns out to have smaller flat lines as compared to DaRe. The parameters were adapted ( $R=1/3$ ) when the frame loss probability is more than 0.4 to maintain the data recovery ratio at 99%. This increases the overhead and thus results in the corresponding increase in the ratio for DaRe. Larger data unit sizes will lead to an even higher reduction in transmission time.

### 6.3. MEMORYLESS CODING FOR EFFICIENT AND REAL-TIME DATA RECOVERY FROM CORRUPTED LoRa FRAMES

In this section, we focus on the efficient recovery of corrupted LoRa-frames in real-time, beyond the built-in error correction. Apart from the classic constraints regarding adding short redundancy, keeping the LoRa-infrastructure unchanged, and not using ACK-mechanisms, we introduce two more constraints: (i) The encoding mechanism should be memoryless, i.e., block codes to ensure the freshness of data. Approaches like DaRe [51] require the correct reception of subsequent frames to recover previously lost frames. Thus, they are unsuitable for time-critical applications. (ii) The decoding should be fast enough to provide real-time reception without depending on future frames.

To this end, we introduce DC, Divide & Code<sup>2</sup>—a novel coding scheme at the application layer of LoRaWAN. DC uses lightweight block codes to pre-encode the payload of a LoRa-frame and adds a limited number of encoded bytes before LoRa-FEC is applied, to provide robustness. The decoder of DC: (i) limits the decoding time to acceptable values; (ii) prioritizes the decoding of certain bytes in a frame based on their probability of being corrupted.

Our contributions are the following:

- We perform a realistic evaluation of frame-corruption in LoRa networks and draw insights on the patterns in which (bursts of) errors occur and the correlation among errors in the same frame (see subsection 6.3.1).
- We design a novel, memoryless coding scheme, called DC, that can be introduced at the application layer of LoRaWAN, being independent of the LoRa standard. DC recovers payload efficiently and in real-time well beyond the built-in error-correcting capability (see subsection 6.3.2).
- We define specific decoding algorithms depending on frame size and using the characteristics of frame corruption in LoRaWAN (see subsection 6.3.2.3).
- Through simulations and real-world experiments using our testbed of LoRa modules, we compare DC not only to the built-in error-correcting scheme of LoRaWAN but also to Reed-Solomon codes and the recent scheme ReDCoS [52]. Furthermore, we dictate the ratio among data bytes and error-correcting bytes that optimizes consumption and decoding ratio (see subsection 6.3.3).

#### 6.3.1. CHARACTERIZATION OF ERROR PATTERN

Before we develop more effective, faster, and energy-efficient data recovery algorithms, we first characterize frame corruption in LoRaWAN. We find conditions under which the majority of frame corruption is observed. Further, we evaluate which parts of LoRa-frames are more prone to corruption, and analyze various aspects of symbol corruption and understand how bursts of errors occur.

**Data Collection.** We conduct experiments on our testbed using SX1261 LoRaWAN modules as sender and receiver on our testbed. We vary the transmitter-receiver distance

<sup>2</sup>This is similar to the Divide & Conquer principle.

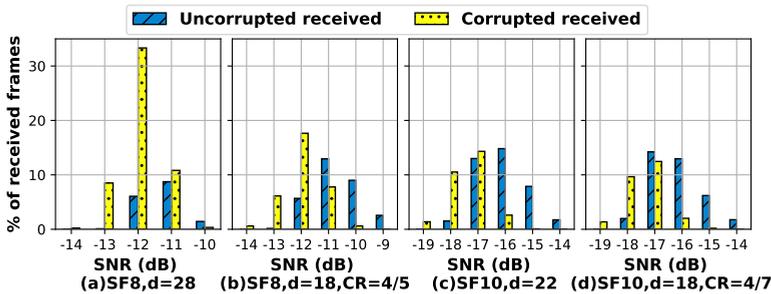


Figure 6.12: The portion of corrupted and uncorrupted received frames out of 5000 transmitted frames ( $d$  is payload size in byte).

between 100 – 300 m, using low transmission powers (0 dBm) under (N)LoS conditions, employing different types of isotropic antennas (0 – 2 dBi gain). The devices operate at 868.1 MHz, using the bandwidth of 125 kHz at SF8 and SF10. The following four cases are studied: (SF8,  $d = 28$ , no CR), (SF8,  $d = 18$ , CR = 4/5), (SF10,  $d = 22$ , no CR), and (SF10,  $d = 18$ , CR = 4/7) where  $d$  is the number of bytes within the payload. 5,000 frames with random data are transmitted for each case. Note that we used relatively small frames for our in-field experiments (Fig. 6.12) considering the data from TTN, which shows 75.3% of the frames have at most 30 B payload [109].

**Frame corruption.** As seen in Fig. 6.12, the frames received with SNR values between  $[-10, -13]$  dB and  $[-16, -19]$  dB for SF8 and SF10, respectively, have high probabilities of being corrupted. These values correspond to RSSI of around  $-115$  dBm. Frames received with SNR values above this range are correctly decoded and below this range are lost completely. For example, 33.32% of the 5,000 frames (i.e., 1,675) that were transmitted on SF8 with  $d = 28$  were received corrupted when SNR is  $-12$  dB. At high SFs, the total number of corrupted frames is reduced due to more robust encoding, i.e., 29.06% of frames were corrupted for SF10 compared to 49.72% for SF8 shown in Fig. 6.12(c) and Fig. 6.12(a), respectively. We did not focus on SF11 and SF12 because frame loss is mostly observed instead of corruption, as we confirmed in Section 6.2 [51]. With inbuilt redundancy using CR, part of the corrupted bytes could be retrieved as seen in Fig. 6.12(b) and Fig. 6.12(d). However, the sheer amount of corrupted (bursts of) bytes render the simple LoRa-FEC unable to correct them, still leading to 32.82% and 25.78% of frames being corrupted at SF8 and SF10, respectively.

**Byte error probability vis-à-vis position.** Fig. 6.13(a) depicts the probability of a byte being corrupted based on its position in the payload for the above cases. For SF8, the first byte has the lowest probability of being corrupted. For SF10, the first and the second bytes have the lowest probability of being corrupted. At SF8, except for the first byte, there is a pattern where 3 bytes have an average probability of being corrupted ( $\sim 18\%$  for  $d = 28$  and  $\sim 14\%$  for  $d = 18$ ) and the fourth byte has a higher probability ( $\sim 22\%$  for  $d = 28$  and  $\sim 18\%$  for  $d = 18$ ). This pattern repeats per 4 bytes. At SF10 except for the first and the second bytes, there is a similar pattern where there are 4 bytes with an average probability of being corrupted ( $\sim 12\%$ ) and 1 byte with a higher probability ( $\sim 15\%$ ). This pattern repeats per 5 bytes.

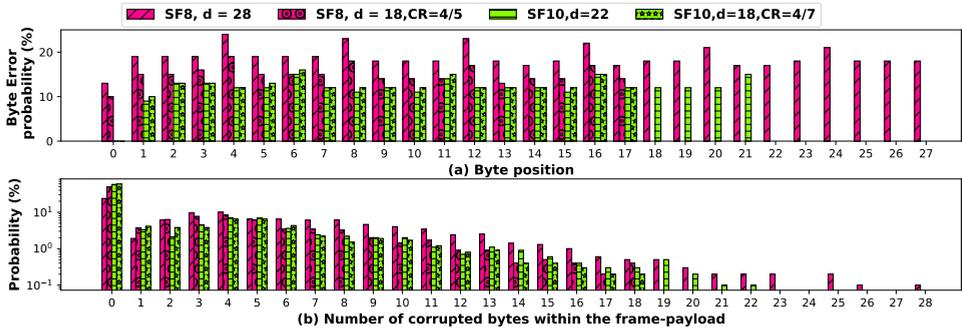


Figure 6.13: (a) Byte Error Rate ( $\beta$ ) versus byte position, (b) Probability for different numbers of corrupted bytes.

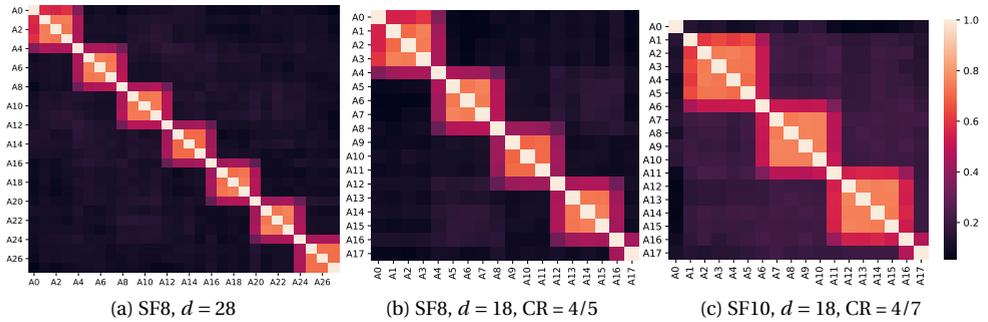


Figure 6.14: Corruption correlation heat-map for: (a) SF8,  $d = 28$ , (b) SF8,  $d = 18$ ,  $CR = 4/5$ , (c) SF10,  $d = 18$ ,  $CR = 4/7$ .

**The number of corrupted bytes.** Fig. 6.13(b) shows the probability of receiving a payload with a different number of corrupted bytes for the previous 4 cases. Apart from the case of no corruption (0 bytes), a peak is observed for having 3, 4, and 4, 5 corrupted bytes per received payload in SF8 and SF10, respectively, confirming our observations regarding bursts of errors in Fig. 6.13(a).

**Correlation among corrupted bytes.** We arrange received data bytes (of corrupted frame) as a row having  $d$  columns corresponding to a byte. Each element can take 0/1; 1 if the associated byte is corrupted, and 0 otherwise. We define  $A_i$  as the  $i$ -th column associated with the  $i$ -th byte of the received payload where  $i \in \{0, 1, \dots, d-1\}$ . The Pearson correlation coefficient between any  $A_i$  and  $A_j$  depicts the strength of linear correlation between the given two columns. It ranges from  $-1$  to  $1$  where correlation coefficients of  $0$  and  $\pm 1$  depict no correlation and a high correlation, respectively. Fig. 6.14 visualizes the correlation matrix as a heatmap for (SF8,  $d = 28$ ), (SF8,  $d = 18$ ,  $CR = 4/5$ ), and (SF10,  $d = 18$ ,  $CR = 4/7$ ). As depicted in Fig. 6.14, there is a pattern for each SF. Considering SF8, for any  $\alpha \in \mathbb{Z}$  and

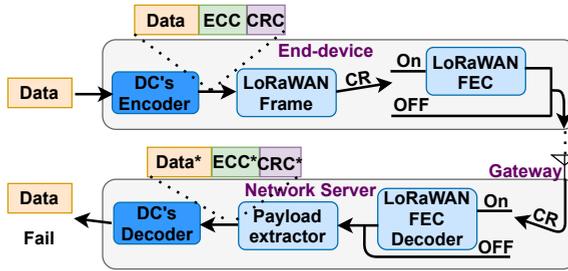


Figure 6.15: The encoder and corrupted frame decoder.

$\beta \in \mathbb{Z}$ , considering  $\alpha' = (\alpha \oplus \theta)$  and  $\xi = (\alpha - \alpha' + \gamma)$  we have:

$$\rho(A_\alpha, A_\beta) \begin{cases} \geq 0.3 & \text{if } \alpha' = \gamma \ \& \ \beta \in [\alpha - \theta, \alpha + \theta] \\ \geq 0.3 & \text{if } \alpha' \neq \gamma \ \& \ \beta \in [\xi, \xi + \theta] \\ < 0.3 & \text{otherwise,} \end{cases} \quad (6.1)$$

where  $(\theta = 4, \gamma = 0)$  for SF8,  $(\theta = 5, \gamma = 1)$  for SF10, and  $\oplus$  is the modulo operation. For SF10,  $A_0$  is an exemption as the correlation coefficient between  $A_0$  and any other byte is less than 0.3. The previously observed patterns are only dependent on the SF and not the used CR or payload size.

We verified this pattern in various environments and also at different distances for different SFs as well as with different devices, which are not shown in the figures due to paucity of space. We believe that these patterns are due to the mechanisms and steps involved in decoding the incoming LoRa messages on SX1261. It may be possible that for every 4th or 5th byte the hardware buffer gets refreshed or due to internal noise that is pulling the RSSI down further. The decoding schemes on SX1261 at the hardware level are not made available to the public by SemTech. While these explanations are merely hypothetical, we ensure that external noise has no contribution towards this pattern as the same pattern was repeated during experimentation at various locations.

### 6.3.2. DC: DIVIDE & CODE

DC is a coding scheme at the application layer of LoRaWAN that pre-encodes the payload of frames using lightweight schemes, before LoRa-FEC which is added afterward. DC uses the characteristics of frame-corruption in LoRaWAN to improve the speed and accuracy of decoding.

#### ENCODER

DC can use different CRC checks and ECCs, but in this work, we focus on CRC-32 and Reed-Solomon codes (RS) [114]. RS codes are a family of linear ECCs. In particular, we use systematic  $RS_{256}(n, k)$  over the finite field  $2^8$  with  $n$  and  $k$  being the byte-size of the codeword and the message, respectively. We define  $t$  and  $l$  as the number of error-correcting bytes and CRC check bytes, respectively, where  $n = k + t$ . The encoder generates the encoded data by attaching to the original data  $t$  error-correcting bytes which are linear combinations of the original data. Further, it calculates a CRC of length  $l = 4$  bytes over

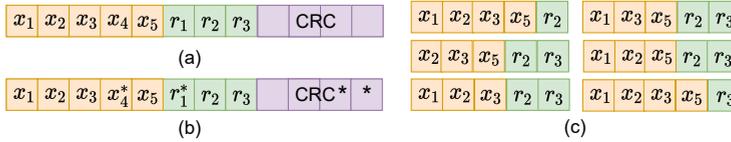


Figure 6.16: Data retrieval example for Case 3: (a) Transmitted payload, (b) Received payload (corrupted bytes marked with a star), (c) All  $k$ -combinations generating correct CRC.

the encoded data. The encoded data and the CRC form the payload of the LoRaWAN frame. Then if CR is on, the LoRa's FEC is added to the frame before transmitting (see Fig. 6.15).

**DECODER**

The DC's decoder is located at the network server as depicted in Fig. 6.15 but it could be placed at GW too if it is computationally powerful. Potentially corrupted encoded data and the CRC are parsed at the DC's decoder. The DC's decoder recovers the data if one of the following cases is true:

**Case 1:** The received encoded data and the received CRC are both uncorrupted.

**Example 1** Let us consider  $k = 5, t = 3, H = 2$ . Then, the data bytes  $\{x_1, x_2, x_3, x_4, x_5\}$  are mapped to the encoded data  $\{x_1, x_2, x_3, x_4, x_5, r_1, r_2, r_3\}$ , and transmitted. Obviously if all the received bytes are uncorrupted, data is retrieved.

**Case 2:** The received CRC is uncorrupted and there is a minimum of  $k$  correct bytes out of the  $k + t$  bytes of the received encoded data.

**Example 2** Let us consider again  $k = 5, t = 3, H = 2$ . However, let us assume the gateway receives  $\{x_1, x_2^*, x_3, x_4, x_5^*, r_1^*, r_2, r_3\}$  where a corrupted byte is shown with a star and CRC is uncorrupted. Then, the decoder tries different  $k$ -combinations. Once, it selects the uncorrupted bytes  $\{x_1, x_3, x_4, r_2, r_3\}$ , it generates the correct encoded data and the correct CRC, which is verified using the received CRC. Hence, data is retrieved (Case 2).

**Case 3:** A minimum of  $H$  bytes of the received CRC are uncorrupted (to verify the recovered CRC) and a minimum of  $k + 1$  bytes of the received encoded data are uncorrupted.  $H$  is determined by the user (usually  $H \geq 2$ ), depending on the size of CRC, and determines the strictness of CRC verification. The decoder tries all the  $k$ -combinations from the given set of  $k + t$  elements to retrieve the data. The total number of  $k$ -combinations is equal to  $C(k + t, k) = \frac{(k+t)!}{k!t!}$ . For each  $k$ -combination, the decoder calculates the remaining  $t$  bytes of the encoded data using the decoding and encoding function of the applied RS code and calculates the associated CRC.

**Example 3** Once more, we consider  $k = 5, t = 3, H = 2$ . Now we assume 2 bytes of the received CRC are corrupted and the received encoded data is  $\{x_1, x_2, x_3, x_4^*, x_5, r_1^*, r_2, r_3\}$  as shown in Fig. 6.16. Then, every time the decoder selects a  $k$ -combination from the set of the correct bytes  $\{x_1, x_2, x_3, x_5, r_2, r_3\}$ , the same correct CRC is generated, i.e.,  $C(6, 5) = 6 = k + 1$  times, see Fig. 6.16 (c). This CRC has  $H = 2$  bytes which are equal to their counterparts from the received CRC and hence is verified. Thus, the data can be retrieved.

Data set (subsection 6.3.1)	Data set (subsection 6.3.3)
(SF8, $d = 28$ )	(SF8, $k = 20, t = 4$ )
(SF8, $d = 18, CR = 4/5$ )	(SF8, $k = 10, t = 4, CR = 4/5$ )
(SF10, $d = 22$ )	(SF10, $k = 10, t = 8$ )
(SF10, $d = 18, CR = 4/7$ )	(SF10, $k = 10, t = 4, CR = 4/7$ )

Table 6.1: Mapping of the Data Sets

### PROPOSED SCHEMES FOR REAL-TIME DECODER

We propose three different schemes to decode in real-time as described below:

**Utilizing the Pattern of CRCs Repetitions (PCR).** Regarding the aforementioned Case 3, instead of repeating the correct CRC at least  $C(k + 1, k) = k + 1$  times in order to retrieve the payload, we observe that the repetition of the same CRC for 2 times is enough to guarantee that it will be repeated for a minimum of  $k + 1$  times because the combinations that generate the same CRC result from correctly encoded data; corrupted data combinations produce a different CRC each time. Thus, we stop the procedure earlier, saving time and computations.

**Utilizing the Pattern of Errors (PE).** We utilize the insights from subsection 6.3.1 regarding the error patterns in corrupted LoRa-frames to select the bytes with a higher probability of being corrupted. Eq. 6.1 represents the correlation between corrupted bytes. As observed in Fig. 6.14, adjacent bytes have a higher probability of being corrupted. Therefore, the decoder should prioritize them. For example, in SF8 if  $A_3$  is corrupted, there is a relatively higher probability that  $A_0, A_1, \dots$ , and/or  $A_4$  are corrupted as well. Afterward, if data is not decoded yet, the decoder tries all the remaining combinations in reverse lexicographical order. To select the few  $k$ -combinations, initially, the  $t$  rightmost elements are set to zero. To generate the next  $k$ -combination, the Boolean vector is rotated to the left by 1 byte. The process stops once all the leftmost elements are ones, i.e., after  $k$  rotations. This technique is called PE, hereafter. Accordingly, PE checks initially only  $k + 1$  of the  $k$ -combinations, which is much less than the total number of  $k$ -combinations. For instance, considering  $k = 20$  and  $t = 4$ , the PE checks only 21  $k$ -combinations initially, while there is a total of 10,626 of them.

**Setting Time Threshold (TT).** There is a considerable portion of large frames that cannot be decoded due to lots of corrupted bytes. The time needed for the decoder to dictate that decoding is not possible would be immense as all the different  $C(k + t, k)$  of  $k$ -combinations had to be tried. Since using the PE most payloads are decoded in a short time, we define a time threshold (T) discarding any payload not decoded before this specified boundary. We call this technique TT.

### 6.3.3. PERFORMANCE EVALUATION

**Data set & Set up.** We used the data collected on our testbed as described in subsection 6.3.1 for our evaluations. Each data set corresponds to a given  $k$  and  $t$  value which is depicted in Table. 6.1. Note that  $d = k + t + 4$ , where 4 bytes are used for the CRC. For all our experiments, we used  $RS_{256}(k + t, k)$  as the ECC in the DC algorithm with random data in the payload. The encoder and decoder are implemented as a C++ library. The transmitted LoRaWAN frames contain 8 and 2 LoRa bytes for preamble and physical header, respectively. The decoding is done on a Core i7-7820HQ, 2.90GHz network server

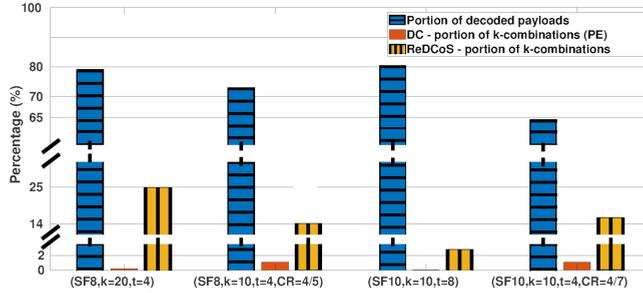


Figure 6.17: The portions of decoded payloads and tried  $k$ -combinations using PE feature of DC, compared to ReDCoS.

using a single thread. We consider  $H = 2$  and a large time threshold enough to process all the payloads unless otherwise stated.

**Schemes for comparison.** We compare our technique with No-Added-ECC and RS-ECC, considering Reed-Solomon code,  $RS_{256}(k + t, k)$ , as an ECC which recovers the data if (i) CRC is uncorrupted and (ii) up to and including  $\lfloor \frac{t}{2} \rfloor$  bytes of the encoded data are corrupted.

**Effect of PE.** As mentioned above, PE initially tries a set of few  $k$ -combinations with the highest probability of decoding the data. PE's portion of  $k$ -combinations is equal to  $\frac{k+1}{C(k+t,k)}$ . Fig. 6.17 shows the ratio of decoded payloads by PE over the total number of decodable payloads, considering corrupted payloads. For instance, for SF10,  $k = 10$ ,  $t = 8$ , PE decodes 80.46% of the corrupted payloads received which are decodable by trying only up to 0.03% of all the  $k$ -combinations, and the remaining 19.54% can be decoded using the rest 99.97%  $k$ -combinations. Further, DC outperforms ReDCoS regarding the number of  $k$ -combinations needed to achieve the same percentage of decoding. For example, as seen in Fig. 6.17 ReDCoS needs to try 125x more  $k$ -combinations than DC to decode 80% of payloads at (SF8,  $k = 20$ ,  $t = 4$ ).

**Effect of TT.** Fig. 6.18 compares the decoding ratio of DC using different time thresholds against ReDCoS, RS-ECC, and No-Added-ECC. We used a similar approach as TT for ReDCoS (we call it ReDCoS(STT)) for a fair comparison. For each time threshold, decoding ratio and average processing time of DC and ReDCoS are calculated. Note that the time thresholds are not shown. By increasing the time threshold, the decoding ratio increases at the expense of increasing the average processing time. After a certain point, increasing the time threshold does not affect critically anymore the decoding ratio as the payloads are either decoded or failed to decode. It is assumed that RS-ECC and No-Added-ECC process the payload instantly. The initial sharp increase in decoding ratio for DC and ReDCoS for all cases corresponds to the uncorrupted payloads which can be decoded quickly. Besides, we observe another sharp increase in decoding ratio for DC. This is due to trying a few  $k$ -combinations with the highest probability of containing the correct data (utilizing PE). DC for SF8,  $k = 20$ ,  $t = 4$  outperforms RS-ECC and No-Added-ECC after average processing time of as low as 0.27 ms and 0.083 ms, correspondingly. Furthermore,

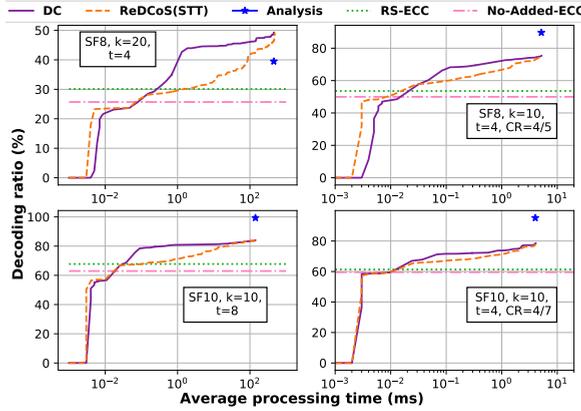


Figure 6.18: Decoding ratio per processing time using TT.

the average processing time can be reduced from 458 ms to 1.91 ms at the expense of only 5.4% lesser decoding ratio considering DC for SF8,  $k = 20$ ,  $t = 4$ . However, for ReDCoS, the same reduction in terms of processing time requires to decrease decoding ratio by 19.7%. According to Fig. 6.18, the decoding ratio deviates at most 16% from the experimental results. For the analysis, we consider the average Byte Error Rate,  $\beta$ .

**Finding the Optimum Point.** Given a fixed  $k + t$ , we evaluate the optimum value for  $\frac{k}{k+t}$ , i.e., which part of an encoded payload should refer to information bytes and which to added redundancy in order to achieve a considerable decoding ratio while keeping energy consumption as low as possible. Note that for No-Added-ECC  $k + t$  changes only by changing  $k$ . For a fair comparison, No-Added-ECC's results are shown for the same  $k$  value as in DC and RS-ECC. Fig. 6.19 shows the decoding ratio, transmission energy per bit, and the total number of  $k$ -combinations utilized. For each case, the in-field experiments are performed for a specific  $k$ , see vertical dashed lines, and the rest of the values are calculated accordingly. Decoding ratio increases with decreasing  $\frac{k}{k+t}$  because the portion of error-correcting bytes increases for RS-ECC and DC. RS-ECC corrects up to and including  $\lfloor \frac{t}{2} \rfloor$  corrupted bytes. Thus, its decoding ratio increases by adding two bytes to the error-correcting bytes. Considering No-Added-ECC, decoding ratio increases with decreasing  $\frac{k}{k+t}$ , which is equivalent to decreasing  $k$ , as the number of bytes per payload decreases and the probability of having a minimum of one corrupted byte per payload decreases. For SF8,  $d = 28$ , DC provides up to  $1.84\times$  and  $2.49\times$  better decoding ratio compared to RS-ECC and No-Added-ECC, respectively.

The main source of energy consumption in an end device is data transmission. Energy per transmission can be calculated by multiplying the transmission power by the frame time-on-air which depends on the used SF, CR, and bandwidth [54]. Transmission energy,  $E$  (in  $\mu$ J), in Fig. 6.19 is measured per bit as follows,

$$E = \frac{X10^{-6}N_T}{8kN_D}, \tag{6.2}$$

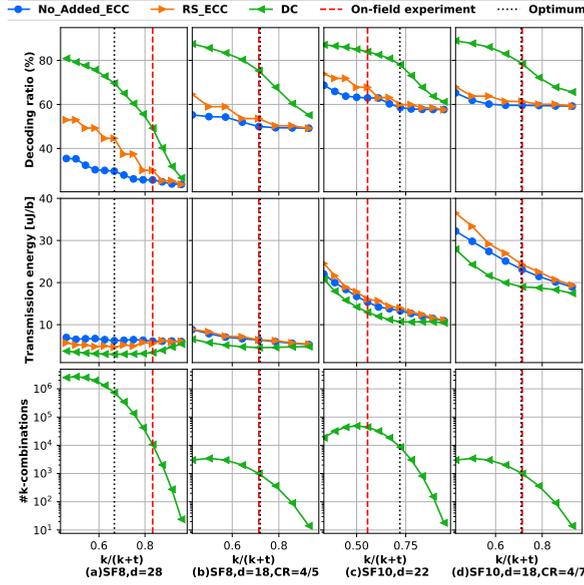


Figure 6.19: Finding the optimum value for  $\frac{k}{k+t}$  considering a fixed  $k+t$ . As  $k+t$  is constant, the payload size for RS-ECC and DC is constant and equal to  $d = l + k + t$  for each case. The payload size for No-Added-ECC changes by changing  $k$  as  $t = 0$ .

6

where  $X$  the energy consumed per transmission,  $N_T$  the number of transmitted frames, and  $N_D$  the number of decoded payloads. Eq. 6.2 indicates the average energy consumed to receive one *data* bit, correctly, by the decoder. If a frame is not correctly received, it should be retransmitted. To consider the effect of frame loss, Eq. 6.2 includes  $\frac{N_T}{N_D}$ . DC and RS-ECC include error-correcting bytes, which lead to extra energy consumed compared to No-Added-ECC for a given  $k$ . Furthermore, the  $8 \cdot k$  data bits in a frame carry the main information. For SF8,  $d = 28$  and SF8,  $d = 18$ , CR = 4/5, with decreasing  $\frac{k}{k+t}$ , transmission energy initially decreases as the probability of decoding increases. Subsequently, it increases as  $k$  decreases while there is a fixed overhead per frame coming from error-correcting bits, preamble, and physical header. The latter one is the dominant factor for SF10,  $d = 22$  and SF10,  $d = 18$ , CR = 4/7. For SF8,  $d = 18$ , CR = 4/5, DC consumes  $1.49\times$  and  $1.99\times$  less energy for correctly transmitting each data bit compared to RS-ECC and No-Added-ECC, respectively.

Fig. 6.19 also shows the total number of  $k$ -combinations ( $= C(k+t, k)$ ) per case. This is an indication of the throughput of the decoder. The total number of  $k$ -combinations peaks at  $k = t$ , i.e.,  $\frac{k}{k+t} = 0.5$ . Overall, to minimize the transmission energy while keeping a relatively high decoding ratio,  $\frac{k}{k+t}$  should be chosen close to 0.7 (with  $k+t$  fixed). Therefore, for a fixed  $d$  while  $l = 4$  bytes are allocated to the CRC, roughly 70% of the remained bytes should be data and 30% error-correcting bytes to minimize the transmission energy and keep a high decoding ratio.

## 6.4. CONCLUSIONS

In this chapter, we focused on frame loss and frame corruption in LoRaWAN. Conventional wireless techniques against frame loss, such as using an ACK for every transmitted frame, are omitted in LoRaWAN in order to provide scalability, to save transmission time at the gateways, and to save energy at the end-devices. Further, the built-in Hamming code schemes –correcting at most single errors– cannot repair most corrupted frames, which makes the frames useless leading to data loss and frequent (re)transmissions, i.e., increased energy consumption. Therefore, we turned to the application (APP) layer to guarantee and increase the data reception using mechanisms that do not deviate from the LoRaWAN standard and/or require a change of infrastructure. First, we characterized frame loss and frame corruption in LoRaWAN through real-world experiments using our LoRa-testbed. Then we designed two coding mechanisms at the APP layer of LoRaWAN: (i) Data Recovery (DaRe), a non real-time, memory-based coding scheme that recovers data from lost frames, and (ii) Divide & Code (DC), a real-time, memory-less coding scheme that recovers data from corrupted frames.

We characterized frame loss in terms of spatial and temporal properties by performing large-scale in-field experiments in scenarios involving mobile and stationary devices. Looking into the acquired datasets we found that frame loss in LoRaWAN is significant and noted that it can be also bursty in nature, even for stationary end-devices. We introduced a novel erasure coding method, DaRe, that reduces data loss in LoRaWAN significantly. DaRe is based on applying fountain codes on a sliding window. We evaluated DaRe both with emulations and by implementing it on end-devices. We achieved a significant recovery of 99% with a code rate  $R=1/2$  when channel erasure probability was 0.4. Further, on erasure probability of 0.55, DaRe outperformed LT-codes by  $1.33\times$  and  $2.29\times$  at Gilbert channels and regular channels, respectively.

We used our LoRa testbed, comprising of LoRa-SX1261 modules, to characterize frame corruption. We observed large portions of LoRa-frames being received corrupted under challenging conditions. For example, when  $\text{SNR} \in [-16, -19]$  dB one-third of the LoRa-frames of 22 B at SF10 had errors. Further, our tests showed corruptions in bursts of up to 5 or 6 bytes, with 70%-80% probability for a byte to be erroneous when located next to a corrupted byte. We introduced Divide & Code (DC), a novel application layer coding scheme for LoRaWAN. DC encodes independently and proactively the LoRa-payloads before the Cyclic Redundancy Check (CRC) of LoRa-FEC is applied, increasing the robustness of transmitted frames. We compared with RS-coding and vanilla LoRaWAN, and showed that DC boosts the decoding ratio by  $\approx 2\times$  and  $2.5\times$ , while consuming  $1.5\times$  and  $\approx 2\times$  less energy per correctly transmitted data-bit.



# 7

## CONCLUSION

*A person's success in life is determined by having a high minimum, not a high maximum.*

Donald Knuth

The proliferation of networks of energy-constrained IoT-devices with long-range communication requirements led to the growth of Low Power Wide Area Networks (LPWAN), which were fueled by the recent progress in RF technology. LPWANs cover the needs of such IoT-networks by providing robust encoding at the PHY layer and simple, lightweight protocols at the MAC layer.

Among popular LPWANs, like Sigfox and NB-IoT, Long Range (LoRa)WAN has been the most successful in providing an easily accessible and inexpensive LPWAN, operating at the ISM spectrum. Due to the chirp spread spectrum modulation, LoRa-PHY establishes long links that are robust to fading and noise. Using a star topology LoRa-frames are directly transmitted to the gateway(s) in a single-hop over several kilometers, and then forwarded to a network server, which supports energy-efficient devices that sleep most of the time.

However, the characteristics of LoRa-networks and LoRa-devices challenge LoRaWAN in large-scale deployments involving thousands of devices, questioning its potential to enable the future Smart Cities and Industry 4.0 applications. LoRaWAN's unslotted Aloha protocol at the MAC layer causes interference leading to frame collisions. The constraints of LoRa-devices in energy and computational capabilities hinder the adoption of more sophisticated mechanisms for channel access. Furthermore, due to the long-range communication multi-path fading corrupts LoRa-frames despite their robust design. These factors limit the scalability of LoRaWAN when massive deployments are considered, as the throughput requirements of the respective applications cannot be met. In addition, the existing LoRa-deployments already involve multitudes of devices operating under the vanilla LoRaWAN protocol, so any proposal to address the aforementioned challenges should not require any change in the LoRaWAN standard. Therefore, the goal of this work is to improve the scalability of LoRaWAN while being energy-efficient and backward-compatible with the current LoRaWAN specifications. This is achieved by developing

novel protocols and mechanisms for the MAC and APP layer of LoRaWAN, respectively.

## 7.1. RECAPITULATION

**Real-world evaluation of CE and CAD in LoRaWAN.** We started with an in-field characterization of Capture Effect (CE) and Channel Activity Detection (CAD) for the following reasons: (i) CE can be leveraged to reduce network interference, by having each device decrease its transmission power without compromising frame reception, and (ii) CAD can be utilized as an energy-efficient means of channel sensing for each device to avoid collisions with frames of non-hidden devices. In terms of CE, we conducted experiments involving up to three LoRa-transmitters and one LoRa-receiver. The transmitted frames were fixed to reach the receiving device at slightly different timestamps (millisecond-differences) and/or with different signal strengths. We showed that if an ongoing reception is interfered at its preamble the "stronger" signal is more probable to capture the channel, but if the preamble is already received the rest of the frame is more likely to be received too. Regarding CAD, we performed experiments at different fading environments with (N)LoS conditions and evaluated its success rate for increasing distances between transmitter and receivers and for different levels of elevation of receivers. Using our findings from the characterization of CE and CAD we derived a probability of success for every device attempting to capture a LoRa-channel and/or detect the transmission of LoRa-frames, depending on the receiving times and SNR-values of frames, the distance/position of devices, and the fading environment. We incorporated the rules of deriving the above probabilities into the LoRaWAN-MAC protocols we designed.

**Novel MAC layer protocols for LoRa networks.** Utilizing CAD can evade collisions with (most) frames generated by neighboring devices. Further, a LoRa-device will reduce its transmission power "politely" to leverage the capture effect as long as its probability of capturing the channel is above 50%. Although the above methods improve over vanilla LoRaWAN in terms of packet reception ratio, they fall short when more than half of the channel's "airtime" is occupied, i.e.,  $G > 0.5$  (normalized traffic per second), showing that shuffling transmission times blindly is not enough, especially when the number of transmissions from hidden devices increases. To this point, we adopted the principles of persistent and non-persistent CSMA, and designed localized and distributed algorithms based on the view of traffic from the perspective of an individual device.  $p$ -CARMA and  $np$ -CECADA are used by every LoRa-device to adapt its probability of transmission and the duration of its back-off, respectively, according to heuristics in order to spot the "right time" to transmit. Furthermore, noticing that the high Spreading Factors (9-12) are underutilized in smart city scenarios due to their low data rates, we utilize them for the transmission of short control frames notifying devices that a data-transmission –which took place with a low SF– is over. We call this indirect way of transmission-scheduling SFMAC and it shares principles with time-division protocols as it spreads transmissions in time, reducing the occurrences of overlapping transmissions.

To put the performance of our protocols into perspective we compare them to the best-performing state-of-the-art protocols (see Section 2.2, Table 2.1). Note that the effectiveness of around 70% for  $np$ -CECADA and SFMAC is achieved at normalized traffic of  $G=1$ , showing the high reception ratio achieved due to our algorithms even under network saturation. Compared to state-of-the-art CS-approach LMAC [81], our

MAC protocol	Effectiveness (PRR $\times$ G)	Consumption per byte [mJ]	Overhead	Medium Access	Complexity	Gateway Dependence	Evaluation
MoT [75]	0.41	N/A	ACKs and scheduling-beacons	Aloha (request)/ TDMA (data)	Medium	Medium	S
S-MAC [72]	0.12	N/A	extra field in UL and ADR messages	FDMA	High	High	R
RT-LoRa [77]	0.06	0.049-0.201	$\sim$ 10% communication time for beacons/ACKs	TDMA/ slotted Aloha	Medium	Medium	S
LMAC [81]	0.13	3.02	periodic beacons of 49 B (class-B)	channel-hopping using CS	Medium	Low	R
<i>p</i> -CARMA [45]	0.13	0.127-4.923	+3 B in UL and +1 B in ADR	<i>p</i> -CSMA	Low	Low	R,S
<i>np</i> -CECADA [46]	0.73	0.116-2.369	+1 B in UL/ADR	<i>np</i> -CSMA	Low	Low	R,S
SFMAC	0.69	0.35	one SF (9-11) dedicated to signaling	time-division using CS	Medium	Low	R,S

Table 7.1: Comparison of state-of-the-art MAC protocols for LoRaWAN to *p*-CARMA, *np*-CECADA, and SFMAC (G: normalized traffic R: real-world experiments, S: simulation)

CSMA approach *np*-CECADA improves effectiveness by  $5.62\times$ , while our TDMA approach SFMAC outperforms the corresponding MoT [75] by  $1.68\times$ . At the same time, the energy consumed by all our protocols never surpasses 5 mJ per byte, while usually standing below 1 mJ, respecting the energy constraints of LoRa-devices. At the same time, RT-LoRa [77] that scores similarly in energy terms is  $12.17\times$  less effective than *np*-CECADA. These results are reached without the requirement of any change in LoRaWAN infrastructure and/or deviation from the LoRaWAN standard. Lastly, our protocols are evaluated in real-world scenarios involving a testbed of 30 LoRa-SX1261 devices reaching traffic values of even  $G=2$  (normalized) and outperforming vanilla LoRaWAN by multi-folds.

#### Novel coding mechanisms for data recovery at the application layer of LoRaWAN.

Multi-path fading in LoRa networks leads to frame loss and frame corruption. These constitute a critical obstacle in the scalability of LoRaWAN since corrupted and lost frames reduce the useful channel utilization. Often such frames need to be retransmitted, increasing the network overhead and energy consumption. By conducting in-field experiments we showed the following: (i) the fraction of lost (bursts of) frames can reach even half of the total transmissions as the transmission range increases. (ii) the probability of having corrupted (bursts of) symbols in LoRa-frames increases for specific symbol positions and specific ranges of SNR-values at reception, especially in low SFs.

To recover data from lost frames we introduce DaRe, which reconstructs pieces of information of a lost frame by decoding a window of received frames that follow it. DaRe outperforms not only the repetition coding schemes of vanilla LoRaWAN, but also the well-known LT fountain codes. To recover in real-time data from frames that are received but include corrupted (bursts of) symbols we propose DC, which pre-encodes LoRa-payloads using lightweight coding. The decoder of DC improves the decoding ratio and reduces decoding time, by exploiting our findings regarding the positions of corrupted symbols in LoRa-frames. DC outperforms both vanilla-LoRaWAN and Reed-Solomon coding in the aforementioned aspects. Both DaRe and DC save up considerable amounts of energy, otherwise spent for retransmissions, while adding a small overhead per LoRa-frame, as at most 30% of the symbols of any encoded frame refer to error-correcting data.

To sum up, LoRaWAN must scale to support the requirements of future IoT-applications by becoming able to maintain a high level of throughput while traffic increases. Compared to the current LoRaWAN, not only we increased the maximum throughput by more than

4×, but this was achieved at 6× higher traffic load and while spending less energy per transmission. Further, we made LoRaWAN 2.5× more effective in decoding corrupted frames, saving energy from retransmissions. The MAC protocols and coding mechanisms we created are backward-compatible with the existing LoRa deployments, targeting their gradual evolution and not their exclusion from usage. Therefore, this dissertation made a step forward in improving LoRaWAN's scalability while respecting the standard.

## 7.2. FUTURE WORK

Despite the achievements of this dissertation regarding the scalability of LoRaWAN, there are still open problems towards the realization of ubiquitous LoRa networks.

Both the simulation-based and real-world experiments of our MAC protocols took place using stationary devices. The next step would be to test the ability of the devices to estimate the traffic in their vicinity in mobile scenarios. The above concept can be included more generally in the domain of neighbor discovery, which should be further investigated in LoRaWAN. Neighbor discovery can happen indirectly by using heuristics as in *p*-CARMA and *np*-CECADA, or directly by utilizing beacons as in SFMAC. SFMAC should be further applied on *class B* devices since this class utilizes extra listening windows for beacons, which will minimize the time required for neighbor discovery. Beaconing on predetermined time-slots can be further applied by the LoRa-devices that operate under *p*-CARMA or *np*-CECADA and experience high levels of traffic in order to inform their neighbors about it, sharing principles with the busy-tone mechanism [147]. Energy harvesting from ambient sources, although intermittent, can be used in the same scope but in random time-slots, i.e., when enough energy for a beacon is scavenged. In addition, the application of our protocols with no extra requirements in the standard and infrastructure of LoRaWAN will likely lead to networks comprised of devices operating under different MAC protocols. Obviously, the most "polite" devices among them, i.e., those with sensing and power-adaptation capabilities, will compromise their performance to reduce interference and evade collisions with the devices using vanilla LoRaWAN. The decrease in the collective performance of the network and in the fairness in service among devices due to the coexistence of devices of different MAC protocols should be further evaluated.

Our findings regarding frame corruption and loss should be applied in gateway positioning to guarantee that signals are not received in ranges of SNR-values that are detrimental for their SF. Furthermore, adding coding overhead trades off frame robustness with the amount of information/data that can be transmitted, which is of utmost importance for protocols with bounded duty-cycle like LoRaWAN. Therefore, the next step should be to incorporate our findings regarding the conditions that induce frame corruption and loss into the Adaptive Data Rate (ADR) mechanism of LoRaWAN, in order for the network to dynamically adjust the coding overhead for DC and the degree and window-size for DaRe. Thus, the highest amount of information will be always transmitted for each designated data retrieval ratio.



# BIBLIOGRAPHY

- [1] L. Barbierato, A. Estebsari, E. Pons, M. Pau, F. Salassa, M. Ghirardi, and E. Patti. “A Distributed IoT Infrastructure to Test and Deploy Real-Time Demand Response in Smart Grids”. In: *IEEE Internet of Things Journal* 6.1 (2019), pp. 1136–1146.
- [2] N. Kouvelas and R. V. Prasad. “Efficient Allocation of Harvested Energy at the Edge by Building a Tangible Micro-Grid—The Texas Case”. In: *IEEE Transactions on Green Communications and Networking* 5.1 (2021), pp. 94–105. DOI: [10.1109/TGCN.2020.3047432](https://doi.org/10.1109/TGCN.2020.3047432).
- [3] Akshay Uttama Nambi S. N., Venkatesha Prasad R, Antonio Reyes Lua, and Luis Gonzalez. “PEAT, How Much Am i Burning?” In: *Proceedings of the 9th ACM Multimedia Systems Conference*. MMSys '18. Amsterdam, Netherlands: Association for Computing Machinery, 2018, pp. 316–327. ISBN: 9781450351928. DOI: [10.1145/3204949.3204951](https://doi.org/10.1145/3204949.3204951).
- [4] Bosch. *Market size and connected devices: Where's the future of IoT?* <https://blog.bosch-si.com/internetofthings/market-size-and-connected-devices-wheres-the-future-of-iot/>. [Online; accessed: 2020-09-16].
- [5] International Data Corporation. *IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC*. <https://www.idc.com/getdoc.jsp?containerId=prAP46737220>. [Online; accessed: 2021-04-02].
- [6] Global X ETFs. *The IoT is at the Center of Disruptive Technologies*. <https://www.globalxetfs.com/iot-center-of-disruptive-technologies/>. [Online; accessed: 2021-07-23].
- [7] Matt Leonard. *Declining price of IoT sensors means greater use in manufacturing*. <https://www.supplychaindive.com/news/declining-price-iot-sensors-manufacturing/564980/>. [Online; accessed: 2020-03-19].
- [8] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. “A survey on LPWA technology: LoRa and NB-IoT”. In: *ICT Express* 3.1 (2017), pp. 14–21. ISSN: 2405-9595. DOI: <https://doi.org/10.1016/j.icte.2017.03.004>. URL: <https://www.sciencedirect.com/science/article/pii/S2405959517300061>.
- [9] Tae Rim Park, Tae Hyun Kim, Jae Young Choi, Sunghyun Choi, and Wook Hyun Kwon. “Throughput and energy consumption analysis of IEEE 802.15. 4 slotted CSMA/CA”. In: *Electronics Letters* 41.18 (2005), pp. 1017–1019.
- [10] R. Bruno, M. Conti, and E. Gregori. “Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs”. In: *IEEE Transactions on Mobile Computing* 1.1 (2002), pp. 10–31. DOI: [10.1109/TMC.2002.1011056](https://doi.org/10.1109/TMC.2002.1011056).

- [11] Nikolaos Kouvelas, Ajay K. Keshava, Sujay Narayana, and R. Venkatesha Prasad. “Pushing the Boundaries of IoT: Building and Testing Self-powered Batteryless Switch”. In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019, pp. 231–236. DOI: [10.1109/WF-IoT.2019.8767185](https://doi.org/10.1109/WF-IoT.2019.8767185).
- [12] Milan Saliya, Nikolaos Kouvelas, R. Venkatesha Prasad, and Niels Hokke. “Characterizing and Optimizing Piezo Harvesters for Train Interiors”. In: *2020 IEEE SENSORS*. 2020, pp. 1–4. DOI: [10.1109/SENSORS47125.2020.9278637](https://doi.org/10.1109/SENSORS47125.2020.9278637).
- [13] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi. “Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios”. In: *IEEE Wireless Communications* 23.5 (Oct. 2016), pp. 60–67.
- [14] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. “A comparative study of LPWAN technologies for large-scale IoT deployment”. In: *ICT Express* 5.1 (2019), pp. 1–7. ISSN: 2405-9595. DOI: <https://doi.org/10.1016/j.icte.2017.12.005>. URL: <https://www.sciencedirect.com/science/article/pii/S2405959517302953>.
- [15] Branden Ghena, Joshua Adkins, Longfei Shanguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. “Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity”. In: *The 25th Annual International Conference on Mobile Computing and Networking*. MobiCom '19. Association for Computing Machinery, 2019. ISBN: 9781450361699. DOI: [10.1145/3300061.3345444](https://doi.org/10.1145/3300061.3345444).
- [16] Mark Weiser. “The Computer for the 21 st Century”. In: *Scientific American* 265.3 (1991), pp. 94–105. ISSN: 00368733, 19467087.
- [17] Xufei Mao, Xin Miao, Yuan He, Xiang-Yang Li, and Yunhao Liu. “CitySee: Urban CO2 monitoring with sensors”. In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 1611–1619. DOI: [10.1109/INFOCOM.2012.6195530](https://doi.org/10.1109/INFOCOM.2012.6195530).
- [18] Bigbelly Inc. 2019. *New York City's Times Square Efficiently Manages 26,056 Gallons of Waste and Recycling Each Day with Bigbelly*. <http://info.bigbelly.com/case-study/times-square-new-york-city>. [Online; accessed: 2021-04-05].
- [19] ITU-R. 2017. *Guidelines for evaluation of radio interface technologies for IMT-2020*.
- [20] ITU-R. 2017. *Minimum requirements related to technical performance for IMT-2020 radio interface(s)*.
- [21] ericsson. *Cellular networks for Massive IoT*. <https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-networks-for-massive-iot--enabling-low-power-wide-area-applications>. [Online; accessed: 2021-09-09].
- [22] omdia. *5G and Massive IoT: legacy technologies will bridge the gap for now*. <https://omdia.tech.informa.com/OM004695/5G-and-Massive-IoT-legacy-technologies-will-bridge-the-gap-for-now>. [Online; accessed: 2021-10-03].
- [23] B. Reynders, W. Meert, and S. Pollin. “Power and spreading factor control in low power wide area networks”. In: *2017 IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6. DOI: [10.1109/ICC.2017.7996380](https://doi.org/10.1109/ICC.2017.7996380).

- [24] M. Bor and U. Roedig. “LoRa Transmission Parameter Selection”. In: *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. June 2017, pp. 27–34. DOI: [10.1109/DCOSS.2017.10](https://doi.org/10.1109/DCOSS.2017.10).
- [25] 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)*. 2015, pp. 55–59. DOI: [10.1109/ITST.2015.7377400](https://doi.org/10.1109/ITST.2015.7377400).
- [26] Martin Bor, John Vidler, and Utz Roedig. “LoRa for the Internet of Things”. In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. EWSN '16. Graz, Austria: Junction Publishing, 2016, pp. 361–366.
- [27] M. Aref and A. Sikora. “Free space range measurements with Semtech LoRa technology”. In: *2014 (IDAACS-SWS)*. Sept. 2014, pp. 19–23. DOI: [10.1109/IDAACS-SWS.2014.6954616](https://doi.org/10.1109/IDAACS-SWS.2014.6954616).
- [28] 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. Association for Computing Machinery, 2016, pp. 59–67. DOI: [10.1145/2988287.2989163](https://doi.org/10.1145/2988287.2989163).
- [29] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. “Understanding the Limits of LoRaWAN”. In: *IEEE Communications Magazine* 55.9 (2017), pp. 34–40. DOI: [10.1109/MCOM.2017.1600613](https://doi.org/10.1109/MCOM.2017.1600613).
- [30] L. Kleinrock and F. Tobagi. “Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics”. In: *IEEE Transactions on Communications* 23.12 (1975), pp. 1400–1416. DOI: [10.1109/TCOM.1975.1092768](https://doi.org/10.1109/TCOM.1975.1092768).
- [31] F. Tobagi and L. Kleinrock. “Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution”. In: *IEEE Transactions on Communications* 23.12 (1975), pp. 1417–1433. DOI: [10.1109/TCOM.1975.1092767](https://doi.org/10.1109/TCOM.1975.1092767).
- [32] D. Magrin, M. Centenaro, and L. Vangelista. “Performance evaluation of LoRa networks in a smart city scenario”. In: *2017 IEEE International Conference on Communications (ICC)*. May 2017, pp. 1–7. DOI: [10.1109/ICC.2017.7996384](https://doi.org/10.1109/ICC.2017.7996384).
- [33] N. Abramson. “The ALOHA System-Another Alternative for Computer Communications”. In: *Managing Requirements Knowledge, International Workshop on*. Vol. 1. IEEE Computer Society, Nov. 1970, p. 281.
- [34] Andri Rahmadhani and Fernando Kuipers. “When LoRaWAN Frames Collide”. In: *Proceedings of the 12th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. WiNTECH '18. Association for Computing Machinery, 2018, pp. 89–97. ISBN: 9781450359306. DOI: [10.1145/3267204.3267212](https://doi.org/10.1145/3267204.3267212).

- [35] F. Tobagi. “Multiaccess Protocols in Packet Communication Systems”. In: *IEEE Transactions on Communications* 28.4 (1980), pp. 468–488. DOI: [10.1109/TCOM.1980.1094698](https://doi.org/10.1109/TCOM.1980.1094698).
- [36] I. Rubin. “Access-control disciplines for multi-access communication channels: Reservation and TDMA schemes”. In: *IEEE Transactions on Information Theory* 25.5 (1979), pp. 516–536. DOI: [10.1109/TIT.1979.1056095](https://doi.org/10.1109/TIT.1979.1056095).
- [37] Connected Things. *LoRaWAN Gateways*. <https://connectedthings.store/gb/lorawan-gateways/>. [Online; accessed: 2021-10-23].
- [38] Semtech. *LoRa® wireless RF products*. <https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1261>. [Online; accessed: 2021-04-22].
- [39] Semtech. *LoRa® wireless RF products*. <https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276>. [Online; accessed: 2021-04-23].
- [40] *LoRa Channel Activity Detection (CAD) with SX126x*. <https://semtech.my.salesforce.com/sfc/p/#E0000000Je1G/a/2R000000Q1EX/pKB1DjG1B0AF0cdgLeDqC1Jur0hnVU1LcUH3brd0504>. [Online; accessed: 2020-08-14].
- [41] Jeongkeun Lee, Wonho Kim, Sung-Ju Lee, Daehyung Jo, Jiho Ryu, Taekyoung Kwon, and Yanghee Choi. “An Experimental Study on the Capture Effect in 802.11a Networks”. In: *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*. WinTECH '07. Association for Computing Machinery, 2007, pp. 19–26. DOI: [10.1145/1287767.1287772](https://doi.org/10.1145/1287767.1287772). URL: <https://doi.org/10.1145/1287767.1287772>.
- [42] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. “Exploiting the capture effect for collision detection and recovery”. In: *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II*. 2005, pp. 45–52.
- [43] Chun-Hao Liao, Guibing Zhu, Daiki Kuwabara, Makoto Suzuki, and Hiroyuki Morikawa. “Multi-Hop LoRa Networks Enabled by Concurrent Transmission”. In: *IEEE Access* 5 (2017), pp. 21430–21446. DOI: [10.1109/ACCESS.2017.2755858](https://doi.org/10.1109/ACCESS.2017.2755858).
- [44] *ns-3*. <https://www.nsnam.org/>. [Online; accessed 11-May-2019].
- [45] Nikolaos Kouvelas, Vijay S Rao, R. Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. “p-CARMA: Politely Scaling LoRaWAN”. In: *EWSN '20*. Lyon, France: Junction Publishing, 2020, pp. 25–36.
- [46] Nikolaos Kouvelas, R Venkatesha Prasad, Niloofar Yazdani, and Daniel E Lucani. “np-CECADA: Enhancing Ubiquitous Connectivity of LoRa Networks”. In: *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 374–382.
- [47] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. “LMAC: Efficient Carrier-Sense Multiple Access for LoRa”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. Association for Computing Machinery, 2020. ISBN: 9781450370851. DOI: [10.1145/3372224.3419200](https://doi.org/10.1145/3372224.3419200).
- [48] Norbert Blenn and Fernando Kuipers. *LoRaWAN in the Wild: Measurements from The Things Network*. 2017. arXiv: [1706.03086](https://arxiv.org/abs/1706.03086) [cs.NI].

- [49] P. J. Marcelis, V. Rao, and R. V. Prasad. “DaRe: Data Recovery through Application Layer Coding for LoRaWAN”. In: *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. IoTDI '17. Pittsburgh, PA, USA: Association for Computing Machinery, 2017, pp. 97–108. ISBN: 9781450349666. DOI: [10.1145/3054977.3054978](https://doi.org/10.1145/3054977.3054978). URL: <https://doi.org/10.1145/3054977.3054978>.
- [50] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman. “Efficient erasure correcting codes”. In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 569–584. DOI: [10.1109/18.910575](https://doi.org/10.1109/18.910575).
- [51] Paul J. Marcelis, Nikolaos Kouvelas, Vijay S. Rao, and R. Venkatesha Prasad. “DaRe: Data Recovery Through Application Layer Coding for LoRaWAN”. In: *IEEE Transactions on Mobile Computing* 21.3 (2022), pp. 895–910.
- [52] Niloofar Yazdani, Nikolaos Kouvelas, R Venkatesha Prasad, and Daniel Enrique Lucani Rötter. “Energy Efficient Data Recovery from Corrupted LoRa Frames”. In: *IEEE Global Communications Conference*. Globecom. IEEE Conference and Exhibition. IEEE, 2021.
- [53] Dong-Hoon Kim, Eun-Kyu Lee, and Jibum Kim. “Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed”. In: *Sustainability* 11 (Mar. 2019), p. 1917. DOI: [10.3390/su11071917](https://doi.org/10.3390/su11071917).
- [54] Semtech. *LoRa® and LoRaWAN®: A Technical Overview*. [https://loro-developers.semtech.com/uploads/documents/files/LoRa\\_and\\_LoRaWAN-A\\_Tech\\_Overview-Downloadable.pdf](https://loro-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf). [Online; accessed: 2021-04-12].
- [55] Juha Petäjäjärvi, Konstantin Mikhaylov, Marko Pettissalo, Janne Janhunen, and Jari Iinatti. “Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage”. In: *International Journal of Distributed Sensor Networks* 13.3 (2017), p. 1550147717699412. DOI: [10.1177/1550147717699412](https://doi.org/10.1177/1550147717699412).
- [56] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. “Known and Unknown Facts of LoRa: Experiences from a Large-Scale Measurement Study”. In: *ACM Trans. Sen. Netw.* 15.2 (Feb. 2019). ISSN: 1550-4859. DOI: [10.1145/3293534](https://doi.org/10.1145/3293534).
- [57] Ramon Sanchez-Iborra, Jesus Sanchez-Gomez, Juan Ballesta-Viñas, Maria-Dolores Cano, and Antonio Skarmeta. “Performance Evaluation of LoRa Considering Scenario Conditions”. In: *Sensors* 18 (Mar. 2018), p. 772. DOI: [10.3390/S18030772](https://doi.org/10.3390/S18030772).
- [58] LoRa Alliance. *LoRaWAN® Specification v1.1*. [https://loro-alliance.org/resource\\_hub/lorawan-specification-v1-1/](https://loro-alliance.org/resource_hub/lorawan-specification-v1-1/). [Online; accessed: 2020-10-10].
- [59] Semtech. *LoRaMAC*. <http://stackforce.github.io/LoRaMac-doc/LoRaMac-doc-v4.5.1/index.html>. [Online; accessed 2020-12-28].
- [60] LoRa Developer Portal. *What is an Adaptive Data Rate*. <https://loro-developers.semtech.com/library/tech-papers-and-guides/understanding-adr/>. [Online; accessed: 2021-03-02].

- [61] Semtech. *Reading channel RSSI during a CAD*. [https://semtech.my.salesforce.com/sfc/p/#E0000000Je1G/a/2R000000HSPH/zPFZDnsVI\\_9bVkdXQ6Ru\\_X35c\\_TzJBTxmS5RfBA\\_AfE](https://semtech.my.salesforce.com/sfc/p/#E0000000Je1G/a/2R000000HSPH/zPFZDnsVI_9bVkdXQ6Ru_X35c_TzJBTxmS5RfBA_AfE). [Online; accessed 2020-10-8].
- [62] Semtech. *Application Note: SX126x CAD Performance Evaluation*. <https://semtech.my.salesforce.com/sfc/p/#E0000000Je1G/a/2R000000Q1ES/SPexo9njbhEQLcJVUg1i0Su8p3tpAtwX1jhMBG1XsQI>. [Online; accessed 2020-10-8].
- [63] Tallal Elshabrawy and Joerg Robert. “Enhancing LoRa Capacity using Non-Binary Single Parity Check Codes”. In: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2018, pp. 1–7. DOI: [10.1109/WiMOB.2018.8589188](https://doi.org/10.1109/WiMOB.2018.8589188).
- [64] Olivier Bernard, André Seller, and Nicolas Sornin. “Low power long range transmitter”. In: *Semtech Corporation, Application Number 13154071.8/EP20130154071, Publication Number EP2763321A1*. 2015.
- [65] Xianjin Xia, Yuanqing Zheng, and Tao Gu. “FTrack: Parallel Decoding for LoRa Transmissions”. In: *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. SenSys '19. New York, New York: Association for Computing Machinery, 2019, pp. 192–204. ISBN: 9781450369503. DOI: [10.1145/3356250.3360024](https://doi.org/10.1145/3356250.3360024).
- [66] Tommaso Polonelli, Davide Brunelli, Achille Marzocchi, and Luca Benini. “Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment”. In: *Sensors* 19 (Feb. 2019), p. 838. DOI: [10.3390/s19040838](https://doi.org/10.3390/s19040838).
- [67] Rajeev Piyare, Amy Murphy, Michele Magno, and Luca Benini. “On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT”. In: *Sensors* 18 (Nov. 2018), p. 3718. DOI: [10.3390/s18113718](https://doi.org/10.3390/s18113718).
- [68] Y. Hasegawa and K. Suzuki. “A Multi-User ACK-Aggregation Method for Large-Scale Reliable LoRaWAN Service”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–7.
- [69] T. To and A. Duda. “Timemaps for Improving Performance of LoRaWAN”. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–7.
- [70] J. Haxhibeqiri, I. Moerman, and J. Hoebeke. “Low Overhead Scheduling of LoRa Transmissions for Improved Scalability”. In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 3097–3109.
- [71] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch. “FREE —Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection in LoRaWAN”. In: *IEEE Internet of Things Journal* 7.1 (2020), pp. 669–683.
- [72] Z. Xu, J. Luo, Z. Yin, T. He, and F. Dong. “S-MAC: Achieving High Scalability via Adaptive Scheduling in LPWAN”. In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2020, pp. 506–515.

- [73] Dimitrios Zorbas, Khaled Abdelfadeel, Panayiotis Kotzanikolaou, and Dirk Pesch. "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things". In: *Computer Communications* 153 (2020), pp. 1–10. DOI: <https://doi.org/10.1016/j.comcom.2020.01.056>. URL: <http://www.sciencedirect.com/science/article/pii/S0140366419314677>.
- [74] A. -U. -H. Ahmar, E. Aras, T. D. Nguyen, S. Michiels, W. Joosen, and D. Hughes. "CRAM: Robust Medium Access Control for LPWAN using Cryptographic Frequency Hopping". In: *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2020, pp. 95–102.
- [75] G. Hassan and H. S. Hassanein. "MoT: A Deterministic Latency MAC Protocol for Mission-Critical IoT Applications". In: *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*. 2018, pp. 588–593.
- [76] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin. "Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling". In: *IEEE Internet of Things Journal* 5.3 (2018), pp. 1830–1842.
- [77] 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". In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 10812–10823.
- [78] M. Centenaro and L. Vangelista. "Boosting Network Capacity in LoRaWAN Through Time-Power Multiplexing". In: *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2018, pp. 1–6.
- [79] J. Ortín, M. Cesana, and A. Redondi. "Augmenting LoRaWAN Performance With Listen Before Talk". In: *IEEE Transactions on Wireless Communications* 18.6 (2019), pp. 3113–3128.
- [80] E. M. Rochester, A. M. Yousuf, B. Ousat, and M. Ghaderi. "Lightweight Carrier Sensing in LoRa: Implementation and Performance Evaluation". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–6.
- [81] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. "LMAC: Efficient Carrier-Sense Multiple Access for LoRa". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. MobiCom '20*. Association for Computing Machinery, 2020. DOI: [10.1145/3372224.3419200](https://doi.org/10.1145/3372224.3419200). URL: <https://doi.org/10.1145/3372224.3419200>.
- [82] 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)*. 2018, pp. 1–7. DOI: [10.1109/ICC.2018.8422800](https://doi.org/10.1109/ICC.2018.8422800).
- [83] Luca Beltramelli, Aamir Mahmood, Patrik Österberg, and Mikael Gidlund. "LoRa Beyond ALOHA: An Investigation of Alternative Random Access Protocols". In: *IEEE Transactions on Industrial Informatics* 17.5 (2021), pp. 3544–3554. DOI: [10.1109/TII.2020.2977046](https://doi.org/10.1109/TII.2020.2977046).

- [84] Anna Triantafyllou, Panagiotis Sarigiannidis, Thomas Lagkas, Ioannis D. Moscholios, and Antonios Sarigiannidis. “Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance”. In: *Computer Networks* 186 (2021), p. 107735. ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2020.107735>.
- [85] Yang Liu, Lei Liu, Jiacheng Liang, Jin Chai, Xuemei Lei, and Hui Zhang. “High-Performance Long Range-Based Medium Access Control Layer Protocol”. In: *Electronics* 9.8 (2020). ISSN: 2079-9292. DOI: [10.3390/electronics9081273](https://doi.org/10.3390/electronics9081273). URL: <https://www.mdpi.com/2079-9292/9/8/1273>.
- [86] Mounika Baddula, Biplob Ray, and Morshed Chowdhury. “Performance Evaluation of Aloha and CSMA for LoRaWAN Network”. In: *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. 2020, pp. 1–6. DOI: [10.1109/CSDE50874.2020.9411539](https://doi.org/10.1109/CSDE50874.2020.9411539).
- [87] Congduc Pham. “Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks”. In: Apr. 2018, pp. 1–6. DOI: [10.1109/WCNC.2018.8376997](https://doi.org/10.1109/WCNC.2018.8376997).
- [88] Seungku Kim, Heonkook Lee, and Sungho Jeon. “An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem”. In: *Sensors* 20.4 (2020). ISSN: 1424-8220. DOI: [10.3390/s20041008](https://doi.org/10.3390/s20041008). URL: <https://www.mdpi.com/1424-8220/20/4/1008>.
- [89] Morgan O’Kennedy, Thomas Niesler, Riaan Wolhuter, and Nathalie Mitton. “Practical evaluation of carrier sensing for a LoRa wildlife monitoring network”. In: *2020 IFIP Networking Conference (Networking)*. 2020, pp. 614–618.
- [90] Ting Xu and Ming Zhao. “A LoRaWAN-MAC Protocol Based on WSN Residual Energy to Adjust Duty Cycle”. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. 2020, pp. 1415–1420. DOI: [10.1109/ICDCS47774.2020.00187](https://doi.org/10.1109/ICDCS47774.2020.00187).
- [91] Kasama Kamonkusonman and Rardchawadee Silapunt. “Channel Activity Detection with the Modified Backoff Algorithm for LoRaWAN”. In: *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. 2021, pp. 86–89. DOI: [10.1109/ECTI-CON51831.2021.9454714](https://doi.org/10.1109/ECTI-CON51831.2021.9454714).
- [92] 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. ACM, 2016, pp. 59–67. DOI: [10.1145/2988287.2989163](https://doi.org/10.1145/2988287.2989163). URL: <http://doi.acm.org/10.1145/2988287.2989163>.
- [93] D. Bankov, E. Khorov, and A. Lyakhov. “Mathematical model of LoRaWAN channel access with capture effect”. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–5.
- [94] R. Fernandes, R. Oliveira, M. Luís, and S. Sargento. “On the Real Capacity of LoRa Networks: The Impact of Non-Destructive Communications”. In: *IEEE Communications Letters* 23.12 (2019), pp. 2437–2441.

- [95] D. Magrin, M. Capuzzo, and A. Zanella. “A Thorough Study of LoRaWAN Performance Under Different Parameter Settings”. In: *IEEE Internet of Things Journal* 7.1 (2020), pp. 116–127.
- [96] Raphael Rom and Moshe Sidi. *Multiple Access Protocols: Performance and Analysis*. Berlin, Heidelberg: Springer-Verlag, 1990. ISBN: 0387972536.
- [97] G. Bianchi. “Performance analysis of the IEEE 802.11 distributed coordination function”. In: *IEEE Journal on Selected Areas in Communications* 18.3 (2000), pp. 535–547.
- [98] Robert M. Metcalfe and David R. Boggs. “Ethernet: Distributed Packet Switching for Local Computer Networks”. In: *Commun. ACM* 19.7 (July 1976), pp. 395–404. DOI: [10.1145/360248.360253](https://doi.org/10.1145/360248.360253). URL: <https://doi.org/10.1145/360248.360253>.
- [99] P802.11. “IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. In: (Nov. 1997).
- [100] Nikos Kouvelas, Vijay Rao, and R. R. Venkatesha Prasad. *Employing p-CSMA on a LoRa Network Simulator*. 2018. arXiv: [1805.12263](https://arxiv.org/abs/1805.12263) [cs.NI].
- [101] Chuong B. Do and Serafim Batzoglou. “What is the expectation maximization algorithm?” In: *Nature Biotechnology* 26 (2008), pp. 897–899.
- [102] Michiel Aernouts, Raf Berkvens, Koen Vlaenderen, and Maarten Weyn. “Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas”. In: *Data* 3 (Apr. 2018), p. 13. DOI: [10.3390/data3020013](https://doi.org/10.3390/data3020013).
- [103] *The Things Network*. <https://www.thethingsnetwork.org/>. [Online; accessed 2021-5-7].
- [104] Nikos Kouvelas. “Energy allocation strategies for Micro-grids”. In: *MSc thesis, Delft University of Technology* (2017).
- [105] I. Gurobi. *Gurobi optimizer reference manual*. <http://www.gurobi.com>.
- [106] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. “Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance”. In: *IEEE Communications Letters* 22.4 (2018), pp. 796–799.
- [107] B. Zhou, X. Wang, and X. Tang. “Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2871–2878.
- [108] Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, and Ümit Özgüner. *A Vision-based Social Distancing and Critical Density Detection System for COVID-19*. 2020. arXiv: [2007.03578](https://arxiv.org/abs/2007.03578) [eess.IV].
- [109] Andri Rahmadhani and Fernando Kuipers. “Understanding collisions in a LoRaWAN”. In: *SURF Wiki* (2017).
- [110] Aloÿs Augustin, Jiayi Yi, Thomas Clausen, and William Mark Townsley. “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things”. In: *Sensors* 16.9 (2016). ISSN: 1424-8220. DOI: [10.3390/s16091466](https://doi.org/10.3390/s16091466). URL: <https://www.mdpi.com/1424-8220/16/9/1466>.

- [111] Jafar Ababneh and Omar Almomani. "Survey of Error Correction Mechanisms for Video Streaming over the Internet". In: *(IJACSA) International Journal of Advanced Computer Science and Applications* 5.3 (2014).
- [112] Xing-Jun Zhang and Xiao-Hong Peng. "A testbed of erasure coding on video streaming system over lossy networks". In: *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*. Oct. 2007, pp. 535–540.
- [113] I. S. Reed and G. Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. ISSN: 03684245. URL: <http://www.jstor.org/stable/2098968>.
- [114] Stephen B Wicker and Vijay K Bhargava. *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.
- [115] M. Mitzenmacher. "Digital fountains: a survey and look forward". In: *Information Theory Workshop, 2004. IEEE*. Oct. 2004, pp. 271–276.
- [116] Robert Gallager. "Low-density parity-check codes". In: *IRE Transactions on information theory* 8.1 (1962), pp. 21–28.
- [117] A. Morello and V. Mignone. "DVB-S2: The Second Generation Standard for Satellite Broad-Band Services". In: *Proceedings of the IEEE* 94.1 (Jan. 2006), pp. 210–227.
- [118] Jalaluddin Qureshi, Chuan Heng Foh, and Jianfei Cai. "Primer and recent developments on fountain codes". In: *Recent Advances in Communications and Networking Technology (Formerly Recent Patents on Telecommunication)* 2.1 (2013), pp. 2–11.
- [119] M. Luby. "LT codes". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings*. 2002, pp. 271–280. DOI: [10.1109/SFCS.2002.1181950](https://doi.org/10.1109/SFCS.2002.1181950).
- [120] A. Shokrollahi. "Raptor codes". In: *IEEE Transactions on Information Theory* 52.6 (2006), pp. 2551–2567. DOI: [10.1109/TIT.2006.874390](https://doi.org/10.1109/TIT.2006.874390).
- [121] Petar Maymoukov. "Online codes". In: (Dec. 2002).
- [122] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: *Communications, 1993. ICC, IEEE International Conference on*. Vol. 2. May 1993, 1064–1070 vol.2.
- [123] Mohammed Usman. "Convolutional fountain distribution over fading wireless channels". In: *International Journal of Electronics - INT J ELECTRON* 99 (Aug. 2012), pp. 1–14. DOI: [10.1080/00207217.2011.651694](https://doi.org/10.1080/00207217.2011.651694).
- [124] Hrvoje Jenkac, Joachim Hagenauer, and Timo Mayer. "The Turbo-Fountain and its Application to Reliable Wireless Broadcast". In: *11th European Wireless Conference 2005 - Next Generation wireless and Mobile Communications and Services*. 2006, pp. 1–7.
- [125] Chris Studholme and Ian F Blake. "Random Matrices and Codes for the Erasure Channel". In: *Algorithmica* 56.4 (Apr. 2010), pp. 605–620. ISSN: 0178-4617.

- [126] Samuel Montejo-Sánchez, Cesar A. Azurdia-Meza, Richard Demo Souza, Evelio Martin Garcia Fernandez, Ismael Soto, and Arliones Hoeller. “Coded Redundant Message Transmission Schemes for Low-Power Wide Area IoT Applications”. In: *IEEE Wireless Communications Letters* 8.2 (2019), pp. 584–587.
- [127] Tallal Elshabrawy and Joerg Robert. “Evaluation of the BER Performance of LoRa Communication using BICM Decoding”. In: *IEEE ICCE*. 2019, pp. 162–167.
- [128] Ulysse Coutaud, Martin Heusse, and Bernard Tourancheau. “Fragmentation and Forward Error Correction for LoRaWAN Small MTU Networks”. In: ACM EWSN '20. Junction Publishing, 2020, pp. 289–294.
- [129] Ulysse COUTAUD and Bernard TOURANCHEAU. “Channel Coding for Better QoS in LoRa Networks”. In: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2018, pp. 1–9. DOI: [10.1109/WiMOB.2018.8589121](https://doi.org/10.1109/WiMOB.2018.8589121).
- [130] Siddhartha Borkotoky, Udo Schilcher, and Christian Raffelsberger. “Application-Layer Coding with Intermittent Feedback Under Delay and Duty-Cycle Constraints”. In: June 2020, pp. 1–6. DOI: [10.1109/ICC40277.2020.9148646](https://doi.org/10.1109/ICC40277.2020.9148646).
- [131] Y. Wang, Q. Chen, D. Gan, J. Yang, D. S. Kirschen, and C. Kang. “Deep Learning-Based Socio-Demographic Information Identification From Smart Meter Data”. In: *IEEE Trans. Smart Grid* 10.3 (2019), pp. 2593–2602.
- [132] Jean Michel de Souza Sant’Ana, Arliones Hoeller, Richard Demo Souza, Samuel Montejo-Sánchez, Hirley Alves, and Mario de Noronha-Neto. “Hybrid Coded Replication in LoRa Networks”. In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5577–5585. DOI: [10.1109/TII.2020.2966120](https://doi.org/10.1109/TII.2020.2966120).
- [133] Gonglong Chen, Jiamei Lv, and Wei Dong. “Exploiting Rateless Codes and Cross-Layer Optimization for Low-Power Wide-Area Networks”. In: *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. 2020, pp. 1–9. DOI: [10.1109/IWQoS49365.2020.9212919](https://doi.org/10.1109/IWQoS49365.2020.9212919).
- [134] Georgios Angelopoulos, Anantha P. Chandrakasan, and Muriel Médard. “PRAC: Exploiting partial packets without cross-layer or feedback information”. In: *2014 IEEE International Conference on Communications (ICC)*. 2014, pp. 5802–5807. DOI: [10.1109/ICC.2014.6884247](https://doi.org/10.1109/ICC.2014.6884247).
- [135] <http://www.thingpark.com/>. [Online; accessed 14-Aug-2020].
- [136] Sodaq. *Sodaq Mbili*. <http://support.sodaq.com/sodaq-one/mbili/schema-rev-4/>. [Online; accessed 03-May-2020].
- [137] Semtech. *ETSI Compliance of the SX1272/3 LoRa Modem*. <http://www.semtech.com/images/datasheet/etsi-compliance-sx1272-lora-modem.pdf>. [Online; accessed 16-Jan-2020]. July 2013.
- [138] Yi Huang and Kevin Boyle. *Antennas: from theory to practice*. John Wiley & Sons, 2008.
- [139] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

- [140] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. “Automated Estimation of Link Quality for LoRa: A Remote Sensing Approach”. In: *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2019, pp. 145–156. DOI: [10.1145/3302506.3310396](https://doi.org/10.1145/3302506.3310396).
- [141] W.E. Leland, M.S. Taquq, W. Willinger, and D.V. Wilson. “On the self-similar nature of Ethernet traffic (extended version)”. In: *IEEE/ACM Transactions on Networking* 2.1 (1994), pp. 1–15. DOI: [10.1109/90.282603](https://doi.org/10.1109/90.282603).
- [142] E. N. Gilbert. “Capacity of a burst-noise channel”. In: *The Bell System Technical Journal* 39.5 (1960), pp. 1253–1265. DOI: [10.1002/j.1538-7305.1960.tb03959.x](https://doi.org/10.1002/j.1538-7305.1960.tb03959.x).
- [143] Haowei Bai and Mohammed Atiquzzaman. “Error modeling schemes for fading channels in wireless communications: A survey”. In: *IEEE Communications Surveys Tutorials* 5.2 (2003), pp. 2–9. DOI: [10.1109/COMST.2003.5341334](https://doi.org/10.1109/COMST.2003.5341334).
- [144] M. Zorzi and R.R. Rao. “Latency probability of a retransmission scheme for error control on a two-state Markov channel”. In: *IEEE Transactions on Communications* 47.10 (1999), pp. 1537–1548. DOI: [10.1109/26.795822](https://doi.org/10.1109/26.795822).
- [145] Semtech. *SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver*. Mar. 2015.
- [146] Taoufik Bouguera, Jean-François Diouris, Jean-Jacques Chaillout, Randa Jaouadi, and Guillaume Andrieux. “Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN”. In: *Sensors* 18.7 (2018). ISSN: 1424-8220. DOI: [10.3390/s18072104](https://doi.org/10.3390/s18072104). URL: <https://www.mdpi.com/1424-8220/18/7/2104>.
- [147] Z.J. Haas and Jing Deng. “Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks”. In: *IEEE Transactions on Communications* 50.6 (2002), pp. 975–985. DOI: [10.1109/TCOMM.2002.1010617](https://doi.org/10.1109/TCOMM.2002.1010617).



# PROPOSITIONS

Propositions accompanying the dissertation

## **Towards Ubiquitous and Efficient LoRaWAN**

by

**Nikolaos Kouvelas**

1. To capture the channel, being early is more important than being loud. (Chap. 2)
2. The "polite" way of accessing the channel can go against the collective interest. (Chap. 3 and 4)
3. Reducing the number of hidden terminals to raise the packet reception ratio comes at the hidden costs of false detections, extra energy, and additional traffic. (Chap. 5)
4. Knowing which symbols to decode first saves time, but it takes time to select them. (Chap. 6)
5. In channel access, there is no "one size fits all". It is not a matter of equality, instead a matter of equity.
6. In-field experiments are the most trustworthy way to evaluate a MAC-protocol.
7. Using 10 gateways to meet the requirements of 1,000 LoRa-devices is Engineering. Making these 1,000 devices smart enough to be served by 1 gateway is Science.
8. Posing the right question is the only job of a researcher.
9. The hardest goals start looking pretty normal after you fail.
10. Learning from one's own mistakes is smart. Learning from others' mistakes is wise.

These propositions are regarded as opposable and defensible,  
and have been approved as such by promotor prof. dr. K.G. Langendoen  
and promotor dr. R. Venkatesha Prasad



# ACKNOWLEDGEMENTS

These lines are written during a classic Dutch afternoon; windy, rainy, and cold. I used to hate these afternoons when I first came here, back in September of 2014. Now I barely notice them; humans are adaptable animals! Among them, some take the adaptability to a whole new level. These are the Ph.D. students, because adapting your approach until you succeed in a continuum of failures is the greatest skill you need for a Ph.D. For teaching me the above, and also several other skills, I am obliged to the following three persons in the order of their academic standing.

I start with my promotor prof. dr. Koen Langendoen who, as chair of the ENS group, gave me the opportunity to contribute to the group. Koen showed me that a researcher must be meticulous and analytical in his work, taking care of every little detail, down to the level of commas (seriously, I remember him correcting even the commas in my manuscripts). Further, Koen was breaking every day the stereotypical "gap" between students and professors by being approachable and simple. His door was always open to step in. I still remember having beers with him around Delft, getting pancakes at the city center, and the nice trip we had with the group to Maastricht.

I continue with my other promotor and daily supervisor dr. R. Venkatesha Prasad, better known as VP. He recognized potential in me, and on a Wednesday afternoon, after a group meeting, he asked: "do you want to do a Ph.D.?". VP has shown different aspects of himself to me; the friend, the mentor, the critical judge, the scientist, and the philosopher. But I keep one above all, the dreamer! Sir, thank you for teaching me how to think. And please watch your health because the community can afford to lose scientists, but the truly creative souls are rare.

Then, I will refer to dr. Vijay S. Rao who I consider as my second (unofficial) supervisor. Vijay is the paradigm of a true researcher; humble as he possesses the knowledge and proves this in his work, always up-to-date regarding the state-of-the-art, fair to colleagues and students, and ready to be challenged any time for the sake of science. Our interactions always worked for me as lessons on proper scientific reasoning of concepts and methods.

The above persons shaped me critically as a researcher and it was a great honor for me to have them as my teachers. However, there are also a handful of other people, who contributed in several ways to this milestone that I have now reached. I will start with Sujay Narayana, with whom I have interacted on numerous occasions, from writing papers together to winning the prestigious "Fly Your Ideas" prize of Airbus with our team, the "Zero-heroes". Thank you Sujay for your continuous efforts to reach higher. Your work-ethic and commitment were examples for me to look up to. Further, I extend my gratitude to the other two members of the golden team of "Zero-heroes", Ashvij Narayanan and Niels Hokke. To continue, I like to thank dr. Niloofer Yazdani and her promotor prof. dr. Daniel E. Lucani for our fruitful collaboration in designing coding mechanisms for the APP layer of LoRaWAN. Further, I express my deep gratitude to all MSc students of the ENS group that I partly or completely guided: Gauri Tawde, Sergio

Sotto, Victoria Mavrikopoulou, Teresa Blanco, Niels Hokke, and Milan Saliya. I put my efforts in transferring to them a part of my passion about research. I would like to thank all those who coauthored papers with me and are not already mentioned: prof. dr. Dirk Pesch, dr. Artemios G. Voyiatzis, dr. Akshay U. Nambi, dr. William Liu, dr. Kishor Chandra, ir. Paul J. Marcelis, Venkatramanan Balasubramanian, and Ajay Keshava. Among my colleagues I want to highlight Belma, Vito, Kees, Jorik, Eric, and Ioannis for our beer-nights, bouldering sessions, and coffee conversations. These people are considered not only colleagues but also friends. I would also like to thank all the other colleagues and staff I met and interacted at the ENS group – Fernando, Przemek, Qing, Marco, Georgios, Vineet, Amjad, James, Keyarash, Suryansh, Miguel, Jasper, Talia, Anup, Stef, Kishor, Rens, Sinan, Silvia, Antonia, Sezen, Renan, Paulo, Munire, Ruud, Minaksie, and Paula.

Now, let me move out of academia. First, a big thanks to the employees of Coffee-star and Coffee Company where I always love to sit with my laptop. My love for research can be only compared to my love for a proper espresso. Second, what would we be without friends, people sharing nice moments, and caring about us? Therefore, I send my love to Makis and the nice gang of Rotterdam/Amsterdam for all the crazy (home)-parties, the bars, the festivals, and the deep techno vibes that we survived. Next, big love to Tamisha for everything, especially for accepting me with all my flaws and for caring about me, even when I do not care about myself. And a huge kiss to Jessie (most cunning cat I know). Greetings and love to Georgia who elevates the prestige of LUMC with her scientific work. Much respect and love to Dimitra, Vassilis, Tzo, and Konstantinos. I appreciate all the nice moments we spent in Delft and in Greece and I am looking forward to the next ones. Apart from my friends in the Netherlands, I need to express my gratitude for all the support and nice moments to my friends back in Greece, with whom we are still close after so many years. Polivios, Miltos, and Konstantinos, thank you for all the years that we care for each other. In addition, I am obliged to Diana; thank you for tolerating me in my worst moments, for making me believe in myself every time I was lost. Thank you for your love and care, and finally for making me a better person. At the end, my biggest gratitude goes to my mother, Fani. Being a mathematician, you were the first inspiration for me in loving science and technology. Thank you for teaching me to always ask, to always experiment, and to live my life with curiosity. Thank you for offering me the safe space inside which I am free to win, to lose, and to make mistakes. Thank you for being my mom and my best friend at the same time. And thanks, by the way, for contributing actively in my on-field experiments of CAD that took place in Greece! It is so cool when your mom can understand and give feedback regarding your algorithms.

Allow me now a piece of advice to all aspiring Ph.D. students. If you start a Ph.D. for the title of doctor or for (any) prestige that follows, please do a favor to Science and drop out. But if you start a Ph.D. to contribute to Science, make sure that you listen at least as much as you talk. This advice also holds for professors, as they tend to forget with the years.

*Nikos  
Out and About, 2022*



# LIST OF PUBLICATIONS

1. Niloofar Yazdani, **Nikolaos Kouvelas**, R. Venkatesha Prasad, Daniel E. Lucani, "Divide and Code: Efficient and Real-time Data Recovery from Corrupted LoRa Frames", Under review (2022).
2. Niloofar Yazdani, **Nikolaos Kouvelas**, R. Venkatesha Prasad, Daniel E. Lucani, "Energy Efficient Data Recovery from Corrupted LoRa Frames", IEEE Global Communications Conference (GLOBECOM), 7-11 December 2021, Madrid, Spain.
3. **Nikolaos Kouvelas**, R. Venkatesha Prasad, Niloofar Yazdani, Daniel E. Lucani, "*np*-CECADA: Enhancing Ubiquitous Connectivity of LoRa Networks", IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS), 4-7 October 2021, Denver, CO, USA.
4. Paul J. Marcelis, **Nikolaos Kouvelas**, Vijay S. Rao, and R. Venkatesha Prasad. "DaRe: Data Recovery Through Application Layer Coding for LoRaWAN", IEEE Transactions on Mobile Computing 21.3 (2022), pp. 895–910. doi: 10.1109/TMC.2020.3016654.
5. **Nikolaos Kouvelas**, Vijay S Rao, R. Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. 2020. *p*-CARMA: Politely Scaling LoRaWAN. In Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks (EWSN '20). Junction Publishing, USA, 25–36.
6. N. H. Hokke, S. Sharma, R. Venkatesha Prasad, L. Mottola, S. Narayana, V. S. Rao, **N. Kouvelas**, "RF Information Harvesting for Medium Access in Event-driven Batteryless Sensing", ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 4-6 May 2022, Milan, Italy
7. **N. Kouvelas** and R. V. Prasad, "Efficient Allocation of Harvested Energy at the Edge by Building a Tangible Micro-Grid—The Texas Case," in IEEE Transactions on Green Communications and Networking, vol. 5, no. 1, pp. 94-105, March 2021, doi: 10.1109/TGCN.2020.3047432.
8. M. Saliya, **N. Kouvelas**, R. V. Prasad and N. Hokke, "Characterizing and Optimizing Piezo Harvesters for Train Interiors," 2020 IEEE SENSORS, 2020, pp. 1-4, doi: 10.1109/SENSORS47125.2020.9278637.
9. **N. Kouvelas**, R. V. Prasad and A. U. Nambi, "Efficient Power Sharing at the Edge by Building a Tangible Micro-Grid – the Texas Case," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9148918.
10. **N. Kouvelas**, A. K. Keshava, S. Narayana and R. V. Prasad, "Pushing the Boundaries of IoT: Building and Testing Self-powered Batteryless Switch," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 231-236, doi: 10.1109/WF-IoT.2019.8767185.
11. **Nikos Kouvelas**, Vijay Rao, and R. R. Venkatesha Prasad. Employing *p*-CSMA on aLoRa Network Simulator. 2018. arXiv:1805.12263 [cs.NI].

12. **N. Kouvelas**, V. Balasubramanian, A. G. Voyiatzis, R. R. Prasad and D. Pesch, "On inferring how resources are shared in IoT ecosystems; a graph theoretic approach," 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 760-766, doi: 10.1109/WF-IoT.2018.8355137.
13. V. Balasubramanian, **N. Kouvelas**, K. Chandra, R. V. Prasad, A. G. Voyiatzis and W. Liu, "A unified architecture for integrating energy harvesting IoT devices with the Mobile Edge Cloud," 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 13-18, doi: 10.1109/WF-IoT.2018.8355198.