THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

TOWARDS NEXT-GENERATION LORAWAN: SCALABLE IOT CONNECTIVITY VIA MASSIVE LOGICAL CHANNELS

YU SHIMING

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Towards Next-Generation LoRaWAN: Scalable IoT Connectivity via Massive Logical Channels

Yu Shiming

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

Oct 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

YU Shiming (Name of student)

Abstract

The Internet of Things (IoT) has rapidly expanded across diverse domains such as smart agriculture, environmental monitoring, and industrial automation, driving the need for scalable, energy-efficient, and long-range wireless communication technologies. LoRa, a leading Low-Power Wide-Area Network (LPWAN) technology, has emerged as a promising solution due to its ability to support long-distance communication with minimal energy consumption. Leveraging unlicensed spectrum and a star-of-stars network topology, LoRaWAN enables cost-effective deployment and massive device connectivity. However, as IoT networks scale to city-wide deployments, existing LoRaWAN infrastructures face significant challenges in maintaining reliable and efficient communication.

This thesis addresses three critical challenges that hinder the scalability of current LoRaWAN systems: (1) the inflexibility of existing gateways in supporting massive concurrent transmissions across logical channels; (2) the near-far effect that breaks the presumed orthogonality among logical channels; and (3) the downlink-uplink asymmetry that limits the support for emerging bidirectional IoT applications. To tackle these issues, this dissertation proposes a suite of novel system-level solutions. First, XGate introduces a software-defined gateway architecture that dynamically detects and receives packets across thousands of logical channels, achieving 8.4× higher concurrency than state-of-the-art. Second, Canas pioneers a logical channel interference cancellation technique that reconstructs and subtracts interfering signals to restore practical orthogonality, enabling 2.3× improvement in concurrent transmissions under near-far conditions. Third, FDLoRa presents a full-duplex gateway and concurrent downlink framework that breaks the sequential downlink bottleneck, achieving 5.7× higher downlink capacity and 2.58× greater per-gateway concurrency.

In summary, this thesis makes significant contributions to the design and im-

plementation of next-generation LoRaWAN systems. It introduces the first practical solutions for scalable logical channel reception, interference cancellation, and concurrent downlink transmission, all validated through extensive real-world experiments and trace-driven evaluations. These innovations unlock the full potential of LoRaWAN for massive IoT deployments, offering a robust foundation for future research and development in wide-area, low-power wireless networking.

Publications arising from the thesis

1. **Shiming Yu**, Xianjin Xia, Ningning Hou, Yuanqing Zheng, Tao Gu, "Revolutionizing LoRa Gateway with XGate: Scalable Concurrent Transmission across Massive Logical Channels", The 30th Annual International Conference on Mobile Computing and Networking (MobiCom), Washington D.C., USA, 2024

2. **Shiming Yu**, Xianjin Xia, Ziyue Zhang, Ningning Hou, Yuanqing Zheng, "FD-LoRa: Tackling Downlink-Uplink Asymmetry with Full-duplex LoRa Gateways", The 22nd ACM Conference on Embedded Networked Sensor Systems (SenSys), Hangzhou, China, 2024

3. **Shiming Yu**, Ziyue Zhang, Xianjin Xia, Yuanqing Zheng, Jiliang Wang, "Are LoRa Logical Channels Really Orthogonal? Practically Orthogonalizing Massive Logical Channels", The 23rd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), Anaheim, CA, USA, 2025

4. **Shiming Yu**, Xianjin Xia, Ningning Hou, Yuanqing Zheng, Tao Gu, "XGate: Scaling LoRa Communications to Massive Logical Channels", IEEE Transactions on Networking (TON), 2025.

5. **Shiming Yu**, Xianjin Xia, Ziyue Zhang, Ningning Hou, Yuanqing Zheng, "FD-LoRa: Scaling Downlink Concurrent Transmissions With Full-Duplex LoRa Gateways", IEEE Transactions on Mobile Computing (TMC), 2025.

6. **Shiming Yu**, Ziyue Zhang, Xianjin Xia, Yuanqing Zheng, Jiliang Wang, "Resolving Inter-Logical Channel Interference for Large-scale LoRa Deployments", IEEE Transactions on Mobile Computing (TMC), 2025.

7. Ningning Hou, Yifeng Wang, Xianjin Xia, **Shiming Yu**, Yuanqing Zheng, Tao Gu, "MoLoRa: Intelligent Mobile Antenna System for Enhanced LoRa Reception in Urban Environments", The 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys), Irvine, CA, USA, 2025

8. Ruonan Li, Ziyue Zhang, Xianjin Xia, Ningning Hou, Wenchang Chai, **Shiming Yu**, Yuanqing Zheng, Tao Gu, "From Interference Mitigation to Toleration: Pathway to Practical Spatial Reuse in LPWANs", The 31st Annual International Conference on Mobile Computing and Networking (MobiCom), Hong Kong, China, 2025

9. Jiamin Jiang, **Shiming Yu**, Hao Wang, Haiyang Li, Yuanqing Zheng, Lu Wang, "SlideLoRa: Reliable Channel Activity Monitoring across Massive Logical Channels in LoRa Networks", The 33rd IEEE International Conference on Network Protocols, Seoul, South Korea, 2025

Acknowledgements

The journey toward my Ph.D. has been filled with breathtaking moments, profound inspiration, and formidable challenges. Before embarking on this path, I could not have imagined how demanding yet unforgettable it would be. I am deeply grateful for the guidance, support, and opportunities I have received along the way, and I would like to express my heartfelt thanks to all who have supported me.

First and foremost, I express my sincere gratitude to my supervisor, Prof. Yuanqing Zheng, for his insightful leadership and unwavering support throughout my doctoral studies. His vision for impactful research and his dedication to excellence have been a constant source of inspiration. When I first joined PolyU as a junior Ph.D. student, he challenged me to deliver weekly presentations on high-quality research papers. Preparing for these sessions—reading, analyzing, making slides, rehearsing, and presenting—consumed nearly all my time for six months, leaving little room for my own research. At the time, it felt overwhelming. However, this rigorous training proved to be a long-term investment, significantly sharpening my critical thinking, presentation skills, thought organization and academic writing. I once struggled to speak in front of an audience, but through years of training, I have grown to enjoy public speaking and seize opportunities to present my work. I have shared my research at top international conferences and received multiple presentation awards at PolyU. The transformation I have undergone during my Ph.D. journey still amazes me. It has been a great honor and privilege to be supervised by Prof. Zheng, and I will always remain indebted to him. If I have the opportunity to supervise students in the future, I hope to guide them with the same dedication and vision that he has shown me.

I would like to extend my sincere gratitude to my collaborators for their invaluable support throughout my research journey. I am especially appreciative of Dr.

Xianjin Xia, whose guidance and mentorship profoundly shaped my understanding of research methodologies and academic writing. Working closely with him provided me with critical insights and facilitated my rapid transition from a novice to a qualified Ph.D. researcher. I am also deeply thankful to Dr. Ningning Hou for her early encouragement and for sharing her vision and perspectives on research, which greatly influenced my initial steps. My appreciation extends to Mr. Ziyue Zhang and Ms. Ruonan Li for their substantial assistance in conducting real-world experiments. Moreover, I am grateful to Prof. Jiliang Wang and Prof. Tao Gu for their insightful feedback and constructive suggestions, which significantly enhanced the quality of my research paper.

I would also like to extend my heartfelt gratitude to my esteemed groupmates—Mr. Leming Shen, Dr. Kaiyan Cui, Dr. Qiang Yang, Mr. Xinyu Huang, Mr. Jianing Zhang, Mr. Zijing Ma, Mr. Boan Yang, Ms. Linya Fu, Mr. Zehua Cao—for their camaraderie, encouragement, and the many joyful moments we shared. Their support during challenging times and the knowledge I gained through our interactions have been invaluable. I am equally thankful to the dedicated support team at PolyU for their generous assistance with my experimental work.

Finally, and most profoundly, I wish to express my deepest appreciation to my beloved parents and all my family members for their unwavering support and unconditional love. In particular, I am immensely grateful to Ms. Haiting Duan for sharing with me the joys and hardships of the Ph.D. journey. Your steadfast belief in me, especially during moments when I doubted myself, has been a source of strength and inspiration. I am forever grateful.

Table of contents

List of Figures **xv**

List of Tables **xix**

1 Introduction **1**

1.1 Background 1

1.2 System Challenges and Technical Limitations 3

1.2.1 Inflexibility in network capacity scaling 3

1.2.2 Near-far effect in wide-area deployment footprint 4

1.2.3 Limited downlink capabilities in supporting IoT applications . 5

1.3 Research Contribution 5

1.3.1 Software-defined LoRa gateway 6

1.3.2 Logical channel interference cancellation 6

1.3.3 Concurrent downlink transmission 7

1.4 Organization 7

2 Literature Review **9**

2.1 Coverage Extension 9

2.2 Collision Recovery 10

2.3 MAC Layer Scheduling 10

3 XGate: Scaling LoRa Communication to Massive Logical Channels with Software-Defined Gateway **13**

3.1 Introduction 13

3.2 Background . . . 17
3.2.1 LoRaWAN . . . 17
3.2.2 LoRa Basics . . . 18
3.2.3 LoRa Channels . . . 19
3.2.4 LoRa Gateway . . . 20
3.3 System Design . . . 21
3.3.1 Overview . . . 21
3.3.2 Packet Detection over Logical Channels . . . 22
3.3.3 Enhancing Packet Sensitivity . . . 29
3.3.4 Massive Packet Reception . . . 31
3.4 Implementation and Evaluation . . . 34
3.4.1 Methodology . . . 34
3.4.2 Basic Reception Performance . . . 36
3.4.3 Microbenchmarks . . . 42
3.5 Discussion . . . 47
3.6 Conclusion . . . 48

4 Canas: Resolving Inter-Logical Channel Interference for Large-scale LoRa Deployments 49

4.1 Introduction . . . 49
4.2 Problem and Motivation . . . 54
4.2.1 LoRa Logical Channels . . . 54
4.2.2 Understanding Logical Channel Interference . . . 56
4.2.3 Objective . . . 59
4.3 Limitations of Existing Solutions . . . 61
4.4 System Design . . . 64
4.4.1 Reconstruction of Interfering Logical Channel . . . 64

4.4.2 Enhancing Reconstruction Granularity 68
4.4.3 *Canas*: Put All Together 70
4.5 Evaluation 72
4.5.1 Methodology 72
4.5.2 Logical Channel Interference Cancellation Performance 74
4.5.3 Comparison with state-of-the-art 75
4.5.4 Microbenchmarks 81
4.6 Discussion 82
4.7 Conclusion 85
5 FDLoRa: Empowering Scalable LoRa Downlink for Multi-Functional IoT Applications 87
5.1 Introduction 87
5.2 Background and Motivation 91
5.2.1 LoRa Background 91
5.2.2 Understanding Downlink Bottleneck 94
5.2.3 Motivation 95
5.3 System Overview 96
5.4 In-band Full-duplex Gateway 97
5.4.1 Requirements of in-band full-duplex LoRa gateway 98
5.4.2 SIC for Full-duplex Communication 100
5.5 *FDLoRa* Downlink Framework 105
5.5.1 Concurrent Downlink over Multiple Logical Channels 107
5.5.2 Coordinating All Available Gateways 110
5.6 Evaluation 111
5.6.1 Methodology 111
5.6.2 Basic Performance 113

5.6.3 Microbenchmarks 119
5.7 Discussion 120
5.8 Limitations and Future Work 121
5.9 Conclusion 123
6 Conclusions **125**
References **127**

List of Figures

3.1 Block diagram of a COTS gateway (1). 19

3.2 Packet detection in a large signal spectrum (2 MHz): (a) Raw signals of a LoRa preamble; (b,c) Correlation detection results with local up-chirps in different channel settings; (d-f) Results of dechirping preamble signals with a local down-chirp. 22

3.3 Detect BW-SF configurations: (a) Signals of three packets having the same chirp slope but different BW-SF settings; (b,c) Dechirped results with a down-chirp in (250 kHz, SF10); (d,e) Dechirped results with a down-chirp in (125 kHz, SF8). 25

3.4 Dechirp with short windows. 26

3.5 Detect concurrent packets: (a) Compound signals of packets A, B in (250 kHz, SF10) and packet C in (500 kHz, SF12); (b,c) Dechirped frequencies of signals from the same packet change at a fixed rate (Δf=125 kHz) over consecutive windows; (d) Track frequency changes to separate the three packets. 28

3.6 Use long windows to detect weak packets: (a) Received signals of two packets, *i.e.*, Pkt A in (125 kHz, SF8) and Pkt B in (500 kHz, SF12); (b) Detected results with 0.25×, 0.5×, and 1× of the chirp length of Pkt B. 30

3.7 System architecture of XGate: XGate scans all available channels and automatically configures software Rx chains to receive massive LoRa packets. 31

3.8 Use long windows to detect weak packets: (a) Received signals of two packets, *i.e.*, Pkt A in (125 kHz, SF8) and Pkt B in (500 kHz, SF12); (b) Detected results with 0.25×, 0.5×, and 1× of the chirp length of Pkt B. 32

3.9 Estimation errors of packets' central frequencies before and after calibration. 33

3.10 Performance of single packet reception 35

3.11 Performance of concurrent reception: (a,b) different numbers of concurrent transmissions in a 1.6 MHz spectrum; (c) comparison with state-of-the-art; (d) concurrency under different spectrum bandwidths. 37

3.12 Performance of massive packet reception. 40

3.13 (a) Deployment map of testbed (each dot represents multiple surrounding sites); (b) Coverage comparison between XGate and a COTS gateway. 41

3.14 Impacts of cross-channel interference under (a) two packets with different frequency gaps, and (b) different numbers of concurrent packets. 42

3.15 Impact of window size on (a) single packet and (b) concurrent packet detection. 43

3.16 Near-far effect of (a) two nodes and (b) different network scales. . . . 44

3.17 Time Overhead. 46

4.1 The orthogonality between logical channels can be broken because of *inter-logical channel interference*. The research problem of how to resolve the inter-logical channel interference remains unexplored. . . . 50

4.2 Demodulation under logical channel interference: (a) Spectra of concurrent logical channels; (b) Demodulation of SF10 logical channel; (c) Orthogonality is broken by the increased noise floor. 54

4.3 Illustration and requirements of logical channel interference cancellation problem. 56

4.4 Packets that experience logical channel interference dominate the traffic in large-scale LoRa deployments. 57

4.5 Packets under collisions can be recovered by SOTA solution. 57

4.6 Packets under logical channel interference suffer from significant packet loss. 57

4.7 Illustration of frequency leakage and impacts on interference cancellation: (a) LoRa symbol experiences frequency leakage at chirp edges; (b) Random phase drift; (c) Standard dechirp: two chirp segments are aliased; (d) Cancellation fails due to phase misalignment between the local chirp and Rx signal; (e) Up-sample dechirp: two chirp segments are separated; (f) Cancellation succeeds after calibrating frequency leakage in the local chirp signal. 60

4.8 Existing works deal with explicit collisions under single-channel (2) and cross-channel (3) scenarios while *Canas* solves the increased noise floor from inter-logical channel interference. 62

4.9 Dynamic channel fluctuation: (a) spectra and (b) amplitude of a received LoRa packet. 67

4.10 Workflow of *Canas*: *Canas* iteratively demodulates, reconstructs, and removes individual logical channels from the Rx signal. 68

4.11 Testbed setting of *Canas*. 70

4.12 Performance of logical channel interference cancellation: (a) Signal strength of strong logical channel. (b) Packet reception of weak logical channel. 72

4.13 Comparision with state-of-the-art: (a) packet reception ratio of each logical channel under concurrent transmission; (b) the maximum number of concurrent logical channels supported on a unit frequency; (c) indoor and (d) outdoor performance of concurrent packet reception in a 1.6 MHz spectrum. 73

4.14 Near-far effect at different network scales. 77

4.15 Performance of supporting massive LoRa nodes in large-scale deployments. 79

4.16 Impact of SIR. 81

4.17 Time overhead. 81

5.1 Downlink-uplink asymmetry: (a) LoRaWAN receives uplink concurrently but transmits downlink sequentially; (b) *FDLoRa* scales concurrent downlink to massive logical channels of available gateways. . . 88

5.2 Chirp Spread Spectrum of LoRa. 93

5.3 Downlink-uplink asymmetry bottlenecks effective network capacity. SOTA indicates multi-channel reception (1) with collision recovery (2). 94

5.4 System overview: *FDLoRa* develops full-duplex gateways and a new downlink framework to tackle downlink-uplink asymmetry. 96

5.5 Full-duplex LoRa gateway: (a) Requirements of self-interference cancellation; (b) full-duplex cancellation architecture of *FDLoRa*. 97

5.6 (a) Suppressing interference bins harms the effective modulation space for uplink. The uplink symbol is submerged if it lies close to interference bins (represented as blue zones). (b) The problem is more severe under payload self-interference. 99

5.7 Uplink-downlink orthogonality: downlink self-interference can be separated with conjugate modulation. 100

5.8 SIC for in-band full-duplex LoRa communication: (a) Spectra of Rx signal: downlink SI is 15 dB stronger than uplink; (b) Demodulation of an uplink symbol: orthogonal downlink SI overwhelms the uplink; (c) Digital cancellation suppresses downlink SI to noise floor; (d) Downlink SFD still interferes with uplink symbols. 101

5.9 The downlink framework of *FDLoRa*. 105

5.10 Concurrent downlink over logical channels: (a) Spectra of aggregated logical channels; (b,c) LoRa node #1 and node #2 can extract and receive over different logical channels from the aggregated PHY signal. 107

5.11 (a) USRP gateway connects with analog cancellation circuit; (b) Deployment map of our testbed. 112

5.12 Performance of full-duplex communication. 113

5.13 Performance of downlink scalability: (a) different number of concurrent transmissions; (b) downlink packet rate in large-scale networks; (c) comparison with state-of-the-art; (d) concurrency with more gateways. 114

5.14 Practical application with bursty downlink traffic. 118

5.15 Downlink SFD interference. 118

5.16 Time overhead of downlink framework. 120

List of Tables

3.1 Comparisons between XGate and SoTA. 47

4.1 Comparisons between *Canas* and SOTA. ***Canas*** addresses related but different research problem. Note that XGate (4) focuses on packet detection and decoding over massive logical channels, and does not address the problem of inter-logical channel interference. 61

5.1 Comparisons between *FDLoRa* and SoTA. 121

Chapter 1

Introduction

1.1 Background

In recent years, the proliferation of Internet of Things (IoT) devices has accelerated markedly, propelled by advancements in low-cost sensing technologies (5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15), embedded systems (16; 17; 18; 19; 11; 9; 20; 21), and wireless communication protocols (22; 23; 24). The widespread and often dense deployment of these devices presents significant challenges to existing communication infrastructures, particularly with respect to scalability, energy efficiency, and spectrum utilization. As IoT networks continue to expand across a broad range of domains, *e.g.*, industrial automation (25), environmental monitoring (26), the demand for communication solutions that are both energy-efficient and capable of supporting long-range connectivity has become increasingly critical. LoRa (27) is a wireless technology specifically developed to address these requirements. Designed to enable long-distance communication with minimal power consumption, LoRa is particularly well-suited for IoT scenarios where devices operate on constrained energy budgets and transmit small volumes of data intermittently. Their capacity to deliver reliable connectivity over several kilometers while maintaining low energy overhead positions LoRa as a foundational element in the development of sustainable and pervasive IoT ecosystems.

LoRa distinguishes itself from other wireless communication technologies through its exceptional long-range transmission and low-power consumption, making it particularly suitable for IoT applications. In contrast to NB-IoT, which operates on licensed spectrum and benefits from enhanced reliability and seamless integration with existing cellular infrastructure, LoRa utilizes unlicensed spectrum, facilitating cost-effective deployment albeit with increased susceptibility to interference. Sigfox, another Low Power Wide Area Network (LPWAN) protocol, similarly emphasizes energy efficiency but suffers from more constrained data throughput and limited network adaptability. Meanwhile, WiFi and Bluetooth are optimized for high data rate transmissions over short distances; WiFi offers superior bandwidth at the expense of substantial power usage, whereas Bluetooth is tailored for personal area networks with minimal energy requirements. Cellular technologies such as 4G and 5G deliver robust, high-speed connectivity ideal for mobile and real-time services, yet their elevated power consumption and operational costs render them less viable for battery-powered IoT devices. LoRa's distinctive combination of extended range, energy efficiency, and economic scalability positions it as a compelling solution for wide area deployment and low-cost IoT applications.

The network architecture of LoRa, typically implemented through the LoRaWAN protocol, follows a star-of-stars topology designed to support scalable, low-power communication across wide geographic areas. At the core of this architecture are end devices, such as sensors and actuators, that transmit data using LoRa modulation to nearby gateways. The gateways act as transparent bridges, forwarding the received messages to a centralized network server via standard IP connections. The network server is responsible for managing data routing, device authentication, and network integrity. Application servers then process the data for end-user applications, while a dedicated join server may handle device activation and security credentials. This layered structure enables efficient, asynchronous communication between devices and

the cloud, making LoRaWAN particularly suitable for massive IoT deployments in smart cities, agriculture, and industrial monitoring.

LoRaWAN is scaling at an unprecedented pace. By mid-2024, deployments exceeded 350 million nodes and 6.9 million gateways, with projections reaching 3.5 billion LPWAN connections by 2030 (27). The market reflects this surge: valued at $7.39 billion in 2024, it is expected to hit $68.12 billion by 2031, growing at a 35.3% annual growth rate (28). This explosive growth creates an urgent need for networks to support massive concurrency. Yet, LoRaWAN's ALOHA-based MAC and limited channelization fundamentally constrain scalability, leading to collisions, degraded capacity, and reduced throughput as device density rises. To sustain smart cities, industrial IoT, and large-scale asset tracking, future LoRa systems must dramatically increase concurrent transmission capacity and reliability, a challenge that becomes critical as global deployments accelerate.

1.2 System Challenges and Technical Limitations

1.2.1 Inflexibility in network capacity scaling

While LoRa is designed to connect millions of edge devices to the network, a fundamental challenge lies in supporting concurrent transmissions from a massive number of LoRa nodes. A recent measurement study (29) revealed that thousands of LoRa nodes may operate within the coverage area of a single gateway, which typically spans several kilometers. Current LoRa gateways (1) rely on pre-configured hardware architectures that can decode up to nine concurrent packets. Although LoRaWAN enforces a low duty cycle for transmissions (27), the high density of nodes often leads to frequent collisions across the limited number of available channels. Since the pioneering work of Choir (30), numerous studies have proposed strategies to mitigate packet collisions by exploiting time (31), frequency (32; 2), and phase domains (33). How-

ever, recent evaluations using validation platforms such as OpenLoRa (34) indicate that even state-of-the-art decoders (2) struggle to support more than two concurrent transmissions per channel. This poor performance stems from LoRa's unique characteristics: long-range and low-SNR communication make physical-layer features unreliable, limiting the effectiveness of collision recovery techniques that work well in traditional wireless systems like Wi-Fi (35). Consequently, SOTA approaches fail to scale under realistic LoRa conditions, where signal quality and synchronization are inherently weak. This limitation underscores the constrained concurrency of existing solutions and highlights the pressing need for more scalable approaches. As the number of edge devices continues to grow, the LoRa infrastructure faces significant challenges in maintaining reliable and efficient connectivity.

1.2.2 Near-far effect in wide-area deployment footprint

LoRa networks are designed to support massive IoT connectivity by leveraging logically orthogonal channels, created through different spreading factors (SFs), to enable concurrent transmissions. However, our extensive real-world measurements reveal a fundamental problem: in practice, these logical channels are not perfectly orthogonal. As LoRa networks scale and cover wide physical areas, they suffer from the near-far effect, where strong signals from nearby nodes significantly raise the noise floor and interfere with ultra-weak signals from distant nodes. This interference, termed inter-logical channel interference, breaks the presumed orthogonality and leads to severe packet loss—up to 61% in some cases. The challenge is exacerbated by the wide-area footprint of LoRa deployments, where signal strength disparities can exceed 30 dB (29), far surpassing the demodulation gain provided by LoRa receivers.

1.2.3 Limited downlink capabilities in supporting IoT applications

LoRa was originally designed for uplink-dominant applications such as smart agriculture (26; 36) and smart metering (25). However, the rapid proliferation of IoT applications in recent years has introduced new demands on network infrastructure. For instance, smart lamppost systems (37) require both uplink and downlink communication for status monitoring and device actuation (e.g., turning lights on/off), while firmware update services (38; 39) rely on bidirectional communication for device authentication and firmware delivery. Moreover, reliable data transfer (40; 41) and adaptive data rate mechanisms (42; 43) necessitate downlink acknowledgments from gateways to confirm successful uplink transmissions. Despite these requirements, current LoRa networks face significant challenges due to the inherent asymmetry between uplink and downlink transmissions. While gateways can receive concurrent uplink packets using multi-channel reception (1) and collision recovery techniques (31), downlink packets must be queued at the network server and transmitted sequentially (44). As the number of connected IoT devices continues to grow, this asymmetry leads to increased downlink queuing delays and network congestion. Consequently, sensors may experience power depletion due to excessive retransmissions, potentially resulting in degraded performance or even network collapse.

1.3 Research Contribution

To address the aforementioned challenges and limitations, this dissertation proposes a suite of LoRa networking solutions designed to enhance gateway design and bridge the scalability gap. Each chapter presents one or more solutions that tackle specific issues through comprehensive system-level design, practical implementation, and rigorous real-world evaluation.

1.3.1 Software-defined LoRa gateway

I propose XGate (4; 45), a novel software-defined LoRa gateway designed to support massive logical channels for concurrent transmissions. Existing LoRa gateways typically resemble hardware-based receiver (Rx) chains that are statically bound to fixed channels, fundamentally constraining their scalability to accommodate thousands of available logical channels. In contrast, XGate integrates an essential auto-configuration mechanism that dynamically detects incoming packets across arbitrary channels in real time and judiciously allocates software-controlled radio resources (*e.g.*, Rx chains and packet demodulators) on demand for packet reception and demodulation. We implement and evaluate XGate in a campus-scale testbed spanning a physical area of 1.08 km×1.2 km. Experimental results demonstrate that XGate significantly enhances LoRa's capability for concurrent transmissions by 8.4× than the state-of-the-art.

1.3.2 Logical channel interference cancellation

I propose Canas (46; 47), a novel gateway design that reconstructs and cancels interfering signals to restore practical orthogonality among logical channels, thereby enabling reliable concurrent transmissions at scale under the near-far effect. This work is the first to highlight the problem of inter-logical channel interference in large-scale LoRa deployments, emphasizing its distinct characteristics and how it differs from previously studied interference scenarios. We conduct a comprehensive analysis of this problem, supported by extensive real-world measurements, to uncover its implications, practical impact, and root causes. Experimental results show that Canas enhances LoRa's concurrent transmission capability across logical channels by up to 2.3× compared to state-of-the-art solutions.

1.3.3 Concurrent downlink transmission

I propose FDLoRa (48; 49), a novel collaborative design that integrates a full-duplex LoRa gateway with a concurrent downlink transmission framework, fundamentally transforming the traditional sequential downlink paradigm of existing LoRaWAN systems. FDLoRa fully exploits the available spectrum and gateway capabilities to enable concurrent downlink transmissions without compromising uplink performance. We implement and evaluate FDLoRa to validate its effectiveness. Experimental results on a three-gateway testbed demonstrate that FDLoRa increases downlink capacity by up to 5.7× compared to LoRaWAN and achieves 2.58× higher downlink concurrency per gateway than the state-of-the-art.

1.4 Organization

The remainder of this dissertation is as follows: in Chapter 2, I introduce the state-of-the-art works of LoRaWAN and review literature about scalability issues in existing LoRa networks; in Chapter 3, I discuss the software-defined LoRa gateway to flexibly cover massive logical channels; in Chapter 4, I discuss the logical channel cancellation to handle the near-far effect in wide-area deployment; in Chapter 5, I discuss the concurrent downlink transmission framework to support diverse downlink-intensive IoT applications; in Chapter 6, I conclude this dissertation.

Chapter 2

Literature Review

2.1 Coverage Extension

Extending coverage and enabling the reception of ultra-weak packets remain fundamental challenges in LoRa networks, and have been the focus of extensive research efforts (50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62). Charm (63) enhances LoRaWAN coverage by exploiting spatial diversity across multiple gateways. It adopts a Cloud-RAN (C-RAN) architecture to aggregate raw signals from distributed gateways, thereby improving packet reception. MALoRa (64) further improves reception by coherently combining weak signals from multi-antenna gateways. Through precise phase alignment, it aggregates packets received by independent antennas to enhance signal-to-noise ratio (SNR). Falcon (65) extends coverage by allowing weak links to selectively interfere with strong transmissions. The receiver leverages the resulting interference patterns to extract weak-link bits using on-off keying (OOK) modulation. XCopy (66) strengthens weak-link reception by coherently combining retransmitted packets. It enables LoRa nodes to transmit with extended preambles for reliable detection and performs fine-grained frequency and phase alignment during packet aggregation.

2.2 Collision Recovery

Numerous prior studies have investigated concurrent LoRa transmissions through parallel decoding techniques (67; 32; 68; 34; 13; 69; 70; 71). Choir (30) exploits hardware imperfections in LoRa devices to distinguish symbols based on their unique frequency offsets. Specifically, it utilizes the central frequency offset (CFO) of collided symbols as a hardware-specific fingerprint to classify transmissions from different devices. FTrack (31) transforms chirp signals into frequency tracks and leverages the time misalignment of symbol edges to separate overlapping transmissions. PCube (33) employs phase differences among synchronized antennas to resolve symbol collisions. It extracts the Phase Difference of Arrival (PDoA) between antenna pairs to distinguish concurrent transmissions. Pyramid (72) enables real-time LoRa collision decoding with peak tracking. NScale (32) amplifies the time offsets between colliding packets with non-stationary signal scaling. CurvingLoRa (73) employs the non-linear chirp that is sensitive to time misalignment in demodulation to resolve collisions. CIC (2) introduces a sub-symbol level decoding approach for multi-packet collisions. Unlike prior methods that focus on symbol classification, CIC formulates collision recovery as an interference cancellation problem, enabling more robust decoding under high contention.

2.3 MAC Layer Scheduling

LoRaWAN currently employs ALOHA as its MAC protocol, which limits scalability. Several studies (74; 75) have proposed packet scheduling and alternative MAC protocols such as CSMA to address this issue. However, full CSMA implementation is constrained by hardware limitations. DeepSense (76) introduces a neural network-based carrier sensing approach, leveraging the ability of neural models to learn LoRaWAN's coding patterns and detect signals buried in noise. LMAC (77)

utilizes Channel Activity Detection (CAD), supported by modern LoRa chips, to sense channel occupancy. It introduces two enhanced CSMA variants: one balances traffic across logical channels using local node information, while the other incorporates global gateway data for improved performance. Notably, LMAC requires no modifications to commercial off-the-shelf (COTS) LoRa nodes, facilitating deployment. However, it remains susceptible to the hidden terminal problem, a common challenge in wireless networks.

Chapter 3

XGate: Scaling LoRa Communication to Massive Logical Channels with Software-Defined Gateway

3.1 Introduction

Low-Power Wide-Area Networks (LPWANs) have emerged as a promising technology for connecting a vast number of physical things to the Internet. LoRa, as one of the LPWAN technology, features large coverage and ultra-low power communication, making it ideal for low data-rate IoT applications such as smart agriculture (26; 78) and smart metering (79; 25). LoRa is poised to potentially become the de facto IoT technology, enabling ubiquitous low-cost IoT connectivity at city scales (80).

LoRa adopts a series of novel PHY- and MAC-layer designs to support massive IoT connections. *Multi-channel communication:* As a LoRa packet occupies a narrow bandwidth (*e.g.*, 125 kHz by default), the entire LoRa spectrum can be divided into many sub-channels. Multiple LoRa nodes can concurrently communicate at different physical channels without mutual interference. *Orthogonal communication:* LoRa PHY modulates a packet with various packet parameters such as bandwidths (BW) and Spreading Factors (SF). Signals differing in SF or BW can be orthogonal and

demodulated even in the same physical channel. In other words, orthogonal *logical channels* with different packet parameters can be built over the same physical channel. For example, the US902-928 MHz ISM band can be divided into 208 physical channels. With the above orthogonal configurations, thousands of logical channels can be created for concurrent transmissions.

Conventional commodity gateways rely on multiple hardwired Rx chains to receive LoRa packets from a few (*e.g.*, nine channels (1)) out of the thousands of logical channels. Restricted by the existing gateway design, all connected nodes are crowded in a few channels, leading to serious packet collisions and degraded network performance. Furthermore, since the operating channels of a gateway are typically pre-configured in advance, they may not be optimal for changing conditions. Existing gateways do not provide a sufficient number of channels for LoRa nodes to adapt to the ever-changing network conditions. Although one could increase the number of gateways or Rx chains per gateway to expand the operational channels, such a solution would necessitate a substantial number of gateways and Rx chains to effectively cover all logical channels in the entire LoRa spectrum.

Rather than assembling many Rx chains and locking each chain on a fixed channel as existing gateways do, we aim to fully support all possible LoRa channels using a single Rx chain. We achieve this by efficiently detecting incoming packets of any channels on-the-fly and judiciously dispatching software-controlled radio resources (*e.g.*, Rx chains, packet demodulators) on demand for packet reception and demodulation. In addition to breaking the barrier of massive IoT connections and enabling scalable concurrent transmissions across all LoRa logical channels, we believe such a design holds great promise in supporting new communication and networking paradigms for LoRaWAN. For example, with the ability to receive packets without prior knowledge of their configurations in our design, LoRa nodes can seamlessly join deployed networks in a "come-and-be-served" manner, eliminating the need for

cumbersome channel configurations for both nodes and gateways. Additionally, this new gateway design empowers LoRa nodes to freely select logical channels and parameter configurations for their next transmissions, facilitating fast adaptation to network dynamics.

To this end, this chapter presents *XGate*, a novel gateway design that enables flexible packet reception over massive LoRa channels. At its core, XGate decouples the process of LoRa packet reception into two phases, *i.e.*, *packet detection* and *packet decoding*. During the packet detection phase, a packet detector scans the whole Rx spectrum to identify incoming packets and extract their detailed configurations. XGate employs a pool of software-based Rx chains and packet decoders, which can be dynamically configured based on the packet detection results, enabling flexible reception on specific LoRa channels.

However, implementing the idea in practice poses substantial challenges. Firstly, unlike traditional gateways that listen on fixed channels, XGate aims to receive packets from all logical channels within an Rx spectrum. As a packet may arrive in any logical channel unknown a priori, it is non-trivial to identify a packet in a large spectrum, let alone recover the parameters of a packet. The problem turns out to be even more challenging when receiving multiple packets concurrently from different channels. Unlike existing gateways that can use the preset channel information to remove interference from unwanted channels, the signals separated in different sub-channels are now superimposed in a common spectrum, which can cause interference to the detection of concurrent packets (*i.e.*, termed *cross-channel interference*). Moreover, detecting packets with the raw signals of a large Rx spectrum, rather than the narrowband signals of a single LoRa channel, may suffer SNR losses due to increased noise and interference. Therefore, it is crucial, yet challenging, for XGate to detect and receive packets without compromising sensitivity.

XGate introduces a novel design that enables the detection and reception of LoRa

packets in a large spectrum while achieving high resilience to noise and interference. We find that it is feasible to use a chirp signal in one channel to dechirp signals from any other channels once the two signals have the same chirp slope (*i.e.*, rate of frequency change over time). The dechirped signals exhibit periodic patterns in the frequency and time domains, attributed to the repetitive signal structure of a LoRa preamble, which is leveraged to detect any incoming packets. We exploit the fact that the frequency and periodicity of dechirped signals essentially carry information about the channel and chirp profiles (*e.g.*, BW and SF) of a packet. Concurrent packets differing in channels or chirp configurations are usually detected as different frequencies and distinctive periodicity patterns that can be separated in the frequency and time domains. Built on such observations, XGate employs a novel strategy that tracks the frequency changes of dechirped signals across consecutive detection windows to accurately recover communication channels and parameters for concurrent packets. XGate scans the signals in several rounds with diverse chirp slopes and window sizes, where short and long windows are used collaboratively to balance sensitivity and delays for packet detection. Lastly, XGate configures software Rx chains with the correct parameters to receive detected packets.

We implement and evaluate XGate using Software Defined Radio (SDR) and commercial off-the-shelf (COTS) LoRa nodes. Results demonstrate that XGate significantly outperforms existing solutions. Specifically, XGate supports up to 8.4× more concurrent transmissions in the same spectrum than state-of-the-art (2). XGate can be seamlessly integrated with existing LoRa parallel decoding techniques (31; 2) to support 78 % more concurrent transmissions.

In summary, our work makes the following contributions: (1) XGate is the-first-of-its-kind gateway design that enables the scaling of LoRa concurrent transmissions across logical channels, revolutionizing the potential for massive IoT communications; (2) We propose novel techniques to efficiently detect channel activities across all

LoRa channels of a given spectrum and accurately extract the configurations of incoming packets; (3) We implement and evaluate XGate to validate its effectiveness. Our evaluations demonstrate the significant improvements achieved by our design. Beyond this, we believe the design of XGate will have broader impacts on spectrum management, network planning, and channel adaptation in LoRaWAN operations and deployments.

3.2 Background

3.2.1 LoRaWAN

LoRa (Long Range) is a prominent communication technology used in Low Power Wide Area Networks (LPWAN). Originally developed by Cycleo in Grenoble, France in 2012, it was later acquired by Semtech. LoRa is specifically designed for remote, low-power applications and operates on ISM global free bands, including 433MHz, 868MHz, and 915MHz. A LoRa network comprises four main components: LoRa nodes (clients), gateways (or base stations), servers, and clouds.

LoRa employs Chirp Spread Spectrum (CSS) for robust long-range communication. The key advantage of CSS is that it accumulates the energy of the entire chirp into a single point, resulting in a significant signal-to-noise gain. Consequently, LoRa packets can be accurately decoded even below the noise floor. This characteristic enables LoRa to achieve ultra-long-range communication spanning tens of kilometers. Additionally, LoRa clients transmit packets with a very low duty cycle (¡1%, regulated by LoRaWAN), which translates to ultra-low power consumption. LoRa technology has facilitated the development of various emerging applications, such as smart metering (25), smart transportation, smart homes, supply chain management, smart agriculture (26), and healthcare, which have extensive applications in our daily lives.

LoRa's narrow-band modulation and wide-band available spectrum make it capable of connecting a massive number of IoT devices. Furthermore, LoRa utilizes tunable symbol time to balance data rate and receiver sensitivity while providing multiple logical channel options. In real-world IoT systems, the escalating number of IoT devices necessitates massive connectivity and communication capabilities. LoRaWAN, by design, has addressed this need by offering diverse and flexible channel parameter settings. Operating in the ISM band, LoRa supports multiple bandwidth (BW) settings and flexible central frequency (CF) to define a physical channel. Moreover, LoRa supports multiple orthogonal logical channels with different spreading factors (SF) within the same physical channel. Considering widely-used parameters in commercial devices (BW=125,250,500kHz; SF=7-12), there are theoretically 84 available logical channels within just a 1MHz frequency band (comprising eight 125kHz, four 250kHz, and two 500kHz physical channels, each containing six logical channels). Thus, when considering the entire ISM band (902.3-914.9MHz), the total number of available logical channels exceeds 1000.

3.2.2 LoRa Basics

LoRa is a physical layer technique for LPWAN communications (81). It adopts a variant of Chirp Spread Spectrum (CSS) to modulate symbols, where the chirp frequency changes linearly with time. The duration of a chirp (*i.e.*, T_{chirp}) is controlled by Spreading Factor (SF) and bandwidth (BW) parameters of CSS, *i.e.*, $T_{chirp} = \frac{2^{SF}}{BW}$. A chirp with an increasing frequency from $-\frac{BW}{2}$ to $\frac{BW}{2}$ is called *base up-chirp* and represented as follows:

$$C(t) = e^{j2\pi(\frac{1}{2}kt - \frac{BW}{2})t}, \tag{3.1}$$

where k indicates the frequency changing rate, termed *chirp slope*. Namely, $k = \frac{BW}{T_{chirp}} = \frac{BW^2}{2^{SF}}$.

LoRa changes the initial frequency of an up-chirp to modulate symbols. We can

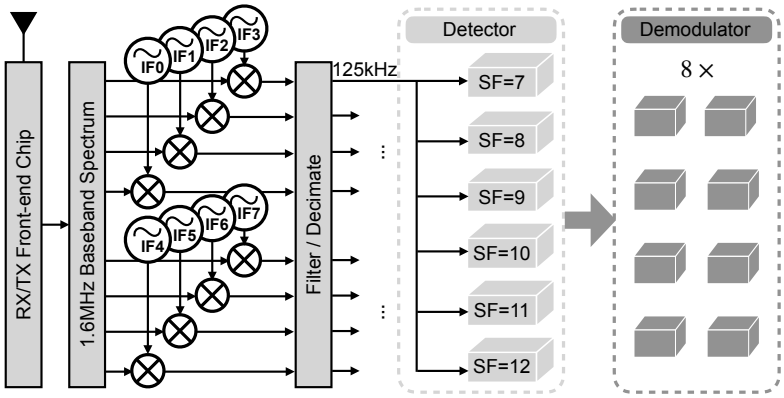


Figure 3.1: Block diagram of a COTS gateway (1).

denote a modulated LoRa symbol $S(f_{sym}, t)$ as follows:

$$S(f_{sym}, t) = C(t)\ e^{j2\pi f_{sym} t}, \tag{3.2}$$

where f_{sym} is the initial frequency of a modulated symbol. To demodulate the symbol, a LoRa receiver first *dechirps* the modulated chirp by multiplying $S(f_{sym}, t)$ with a *down-chirp* denoted as $C^{-1}(t)$ which is the conjugate of a base up-chirp. The operation is represented as follows:

$$S(f_{sym}, t)\ C^{-1}(t) = e^{j2\pi f_{sym} t}. \tag{3.3}$$

Next, the receiver applies FFT to the dechirped signal to extract f_{sym} which indicates the symbol.

A LoRa packet starts with a preamble, followed by two up-chirps as synchronization words and 2.25 down-chirps as a Start Frame Delimiter (SFD), and ends with a packet payload. A preamble is composed of a variant number of base up-chirps with identical initial frequencies. A receiver radio leverages the signal structure of LoRa preamble to detect and demodulate a packet.

3.2.3 LoRa Channels

LoRa operates in unlicensed ISM bands. Different from Wi-Fi and NB-IoT which have fixed channel plans, LoRaWAN does not specify the channel partition in the ISM

bands. Instead, LoRaWAN allows end nodes to select channel frequency and bandwidth which uniquely define a *physical channel*. For example, a commodity LoRa radio (*e.g.*, Semtech SX1276 (82)) supports up to ten bandwidth options ranging from 7.8 kHz to 500 kHz. Unlike conventional technologies that permit one node to communicate per channel, multiple LoRa nodes can simultaneously transmit at the same physical channel using orthogonal parameters (*e.g.*, different SFs and BWs) termed *logical channels*. As COTS LoRa radio supports SFs in 7∼12, six logical channels can be created in the same physical channel. The total number of logical channels exceeds thousands in an ISM band.

3.2.4 LoRa Gateway

Existing COTS LoRa gateways are typically equipped with Semtech chipsets (83). Figure 3.1 shows a simplified block diagram of a COTS gateway consisting of eight Rx chains. The Rx chains can be programmed to receive on different physical channels. A gateway operates on the eight channels simultaneously to detect and receive incoming packets. Note that the operating channels of a gateway are usually configured beforehand and stay invariant after installation. It typically requires in-situ spectrum measurement and analysis to select and configure operating channels for deployed gateways, which can incur high measurement and maintenance overhead.

We note a large gap exists in using COTS gateways to support massive LoRa communications. As all connected nodes can only use the few operating channels of a COTS gateway, it can lead to poor spectrum utilization and sub-optimal network performance due to collisions. Whereas LoRaWAN defines many physical and logical channels with various frequencies and bandwidths. The channel settings available for LoRa nodes can be rather diverse. Ideally, if a gateway can cover all channels in a large spectrum, it can not only support more nodes with reduced collisions but also allow nodes to freely choose among diverse channel settings to better adapt to

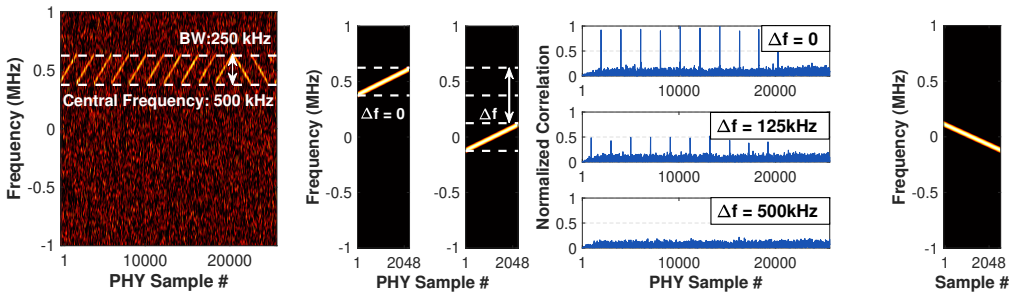
changing network conditions. However, because of the sheer amount of LoRa logical channels, it is challenging and even impractical to use COTS gateways to cover every channel with a dedicated Rx chain. It calls for a better gateway design that can detect and receive packets from all possible LoRa channels in a given spectrum.

3.3 System Design

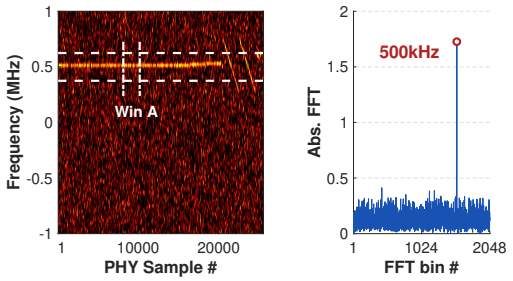
3.3.1 Overview

XGate maintains a pool of 'soft-wired' Rx chains that can be dynamically configured on-the-fly to receive on any frequencies in diverse bandwidth settings. A key enabler of XGate is a powerful packet detector that can scan a large spectrum to detect the detailed parameters of incoming packets (*e.g.*, central frequency, bandwidth, spreading factor) and flexibly configure the software Rx chains to receive detected packets.

However, the implementation of XGate entails substantial challenges. First, as the number of LoRa logical channels can be rather large (*e.g.*, more than hundreds), it is non-trivial to detect packets on so many candidate channels without knowing incoming packets' meta-information. Second, while XGate monitors the whole Rx spectrum to detect LoRa packets, the signals originally separated in different sub-channels can potentially cause *cross-channel interference*. How to correctly detect the concurrent packets of different sub-channels and their detailed parameters is a major challenge faced by XGate. Moreover, as an Rx spectrum spans wider frequency bands than a single LoRa channel, it would include more noise and interference that may reduce signal quality. It is challenging to receive packets without compromising sensitivity. We present novel techniques to address these challenges in the following subsections.



(a) A received preamble (b) Local up-chirps (c) Correlation detection (d) Down-chirp



(e) Dechirp results (f) FFT in Win A

Figure 3.2: Packet detection in a large signal spectrum (2 MHz): (a) Raw signals of a LoRa preamble; (b,c) Correlation detection results with local up-chirps in different channel settings; (d-f) Results of dechirping preamble signals with a local down-chirp.

3.3.2 Packet Detection over Logical Channels

Detecting Packet and Meta-information

As a packet may arrive from any logical channel of an Rx spectrum that is unknown to a gateway, XGate needs to detect not only the arrival of an incoming packet but also the channel and the parameter configuration of the packet.

Packet detection. XGate takes the raw signals of the whole Rx spectrum as inputs and detect incoming packets. A standard LoRa packet detection method uses correlation detection, which correlates incoming signals with a locally-generated LoRa preamble, to detect packets of a specific LoRa channel (1; 77; 84). Can we extend the method to packet detection in the whole Rx spectrum? Figure 3.2(a) illustrates a LoRa packet received in a 2 MHz-wide spectrum. We correlate the

signals with a local preamble chirp in the same SF and BW but different channel settings (see Figure 3.2(b)). As shown in Figure 3.2(c), we can only detect the highest correlation peaks when the central frequencies of the local chirp and the received signals are identical (*i.e.*, $\Delta f = 0$). It means that the channel (*e.g.*, central frequency) of a packet is required to bootstrap the correlation based detection in a particular channel. Obviously, such a method cannot be applied to XGate because the channel information of an incoming packet is not yet available.

How can we detect a packet without knowing the channel? Fortunately, as the Rx spectrum of a gateway covers many narrow LoRa channels, the signals of all LoRa logical channels are physically received. Thus, we can freely access signals of any LoRa logical channels in an Rx spectrum. For instance, we can use a down-chirp signal centered at frequency 0 as shown in Figure 3.2(d) to dechirp the signals displayed in Figure 3.2(a), which gives a result as shown in Figure 3.2(e). If a LoRa preamble is present, we will detect the same tone-frequency in successive windows. It allows us to detect a LoRa packet from the Rx spectrum even when the channel of the packet is unknown.

XGate extends the dechirp operation given by Eq.(5.3) from a single LoRa channel to an entire Rx spectrum to detect a packet, which can be represented as follows:

$$h_i e^{j2\pi f_i t} C_i(t) \cdot C_i^{-1}(t + \Delta t_i) = h_i e^{j2\pi f_i t} \cdot e^{j2\pi \Delta f_i t}, \tag{3.4}$$

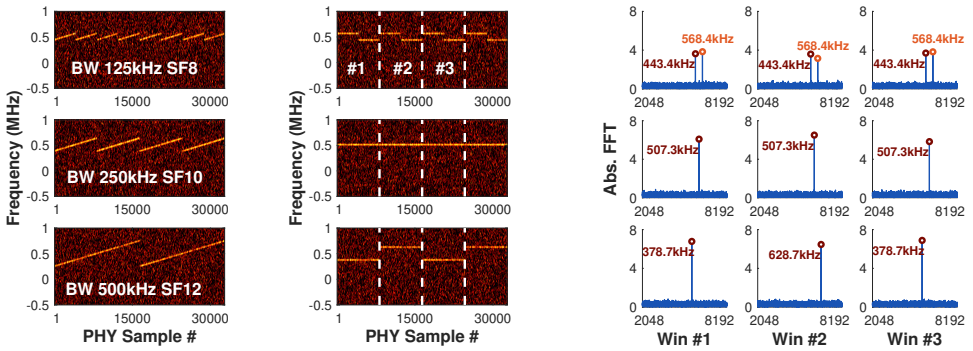
where $h_i e^{j2\pi f_i t} C_i(t)$ denotes a received preamble chirp in a LoRa channel centered at frequency f_i, and $C_i^{-1}(t)$ is a local down-chirp centered at frequency 0 with the corresponding BW_i and SF_i. Here, Δt_i models the misaligned timing between symbol edges of incoming signals and the local down-chirp, which leads to a frequency component $\Delta f_i = (BW_i^2/2^{SF_i})\Delta t_i$ in the dechirped results. Theoretically, the tone-frequency of dechirped signals (Figure 3.2(f)) is essentially $(f_i + \Delta f_i)$. Since edge misalignment Δt_i is within $\pm 0.5\times$ of a chirp duration, the resulting Δf_i shall be

within $\pm 0.5\times$ of the chirp's bandwidth (*i.e.*, BW_i). The tone-frequency ($f_i + \Delta f_i$) deviates at most $\frac{BW_i}{2}$ from f_i (*i.e.*, a channel's central frequency). Hereby, the tone-frequency of dechirped signals not only indicates the presence of an incoming packet, but also roughly estimates its central frequency.

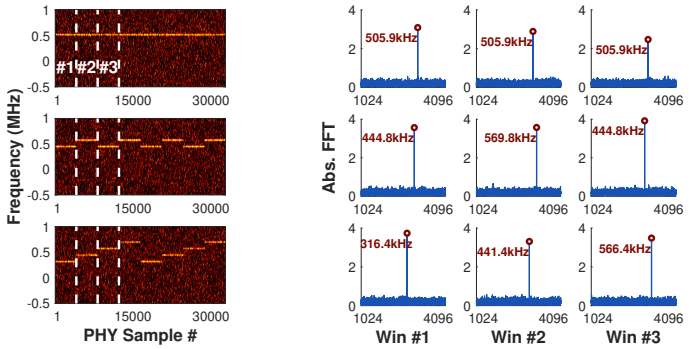
Extracting packet parameters. After detecting an incoming packet, XGate further needs to extract the packet's BW and SF configurations for packet demodulation. Since LoRa chirps in different BWs or SFs are usually orthogonal, if the BW/SF of down-chirp in Eq.(3.4) differs from that of a received packet, the packet signals would not be dechirped to tone frequency nor get detected. By exploiting chirp orthogonality, we may iterate through all BW-SF combinations to find the correct configurations of a packet.

However, such a method may suffer from ambiguous parameter detection since a packet can be dechirped by various chirps differing in BW or SF. For instance, the BW and SF of the three signals shown in Figure 3.3(a) are (125 kHz, SF8), (250 kHz, SF10) and (500 kHz, SF12), respectively. When we use a down-chirp in BW 250 kHz and SF10 to dechirp the three signals, we can detect strong tone frequencies not only from the signals of (250 kHz, SF10) but also from the other two signals. In fact, as the three signals have the same *chirp slope* defined as $k = BW^2/2^{SF}$ (Eq.(5.1)), they are not entirely orthogonal and can be dechirped by the same down-chirp with a slope of $-k$. In particular, Figure 3.3(b,c) plots the dechirped frequencies of those signals in three successive windows. We detect identical single-tone frequencies from successive windows for (250 kHz, SF10). While for (125 kHz, SF8), two frequencies are detected repetitively in every window. In this case, it is ambiguous to tell whether there is only one packet in (125 kHz, SF8) or two packets in (250 kHz, SF10).

XGate resolves such ambiguities by leveraging the fact that chirp signals differing in BW or SF would have different chirp lengths. In particular, if we use a shorter down-chirp (*e.g.*, the shortest chirp length of the three signals) to dechirp the three



(a) Preamble signals (b) Dechirp (setting #1) (c) FFTs of three windows (setting #1)



(d) Dechirp (setting #2) (e) FFTs of three windows (setting #2)

Figure 3.3: Detect BW-SF configurations: (a) Signals of three packets having the same chirp slope but different BW-SF settings; (b,c) Dechirped results with a down-chirp in (250 kHz, SF10); (d,e) Dechirped results with a down-chirp in (125 kHz, SF8).

signals shown in Figure 3.3(a), we will get a unique single-tone frequency in each short window, and the frequency varies across windows with distinctive periodic patterns for the three packets as shown in Figure 3.3(d,e). By detecting the periodicity of frequencies with short windows, XGate can disambiguate the three packets and uniquely recover BW and SF for each packet.

Putting it together. XGate uses down-chirps of different slopes, instead of iterating through BW-SF combinations, for packet detection. As multiple BW-SF settings share a common chirp slope, we can efficiently check fewer slopes rather than all BW-SF settings to detect packets. To avoid ambiguities, we use a short window

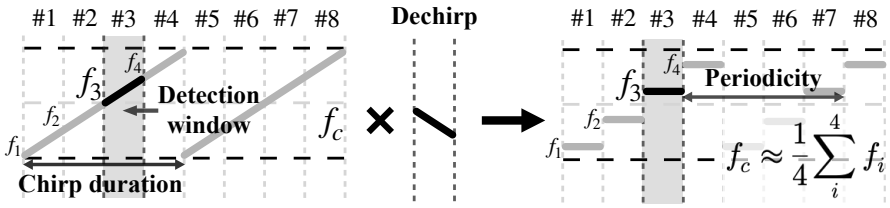


Figure 3.4: Dechirp with short windows.

to dechirp and detect signal patterns, *e.g.*, typically choose a down-chirp with the shortest length of a given slope.

We rewrite Eq.(3.4) to represent the dechirp operation of a short window as below:

$$h_i e^{j2\pi f_i t} \cdot e^{j2\pi f_k t} C_w(k_i, t) \cdot C_w^{-1}(k_i, t) = h_i e^{j2\pi f_i t} \cdot e^{j2\pi f_k t}, \tag{3.5}$$

where $e^{j2\pi f_k t} C_w(k_i, t)$ is part of a long up-chirp located in the target window, f_k denotes the starting frequency of the chirp segment (Figure 3.4), k_i is the chirp slope, and $C_w^{-1}(k_i, t)$ is a corresponding down-chirp in slope k_i. According to Eq.(3.5), the tone frequency shown in any window of Figure 3.3(e) is essentially $(f_i + f_k)$, which signifies the starting frequency of the chirp segment in a short window. The starting frequency $(f_i + f_k)$ would change across windows and the periodicity of frequency changes equals the length of a complete chirp signal as Figure 3.4 illustrates.

XGate detects the periodicity of dechirped frequencies to infer the chirp length of received signals, from which the signals' BW and SF configurations can be restored unambiguously. Additionally, XGate calculates the average of the frequencies (*i.e.*, $(f_i + f_k)$'s) detected from different windows, which would be close to f_i, to estimate the channel frequency of a received packet.

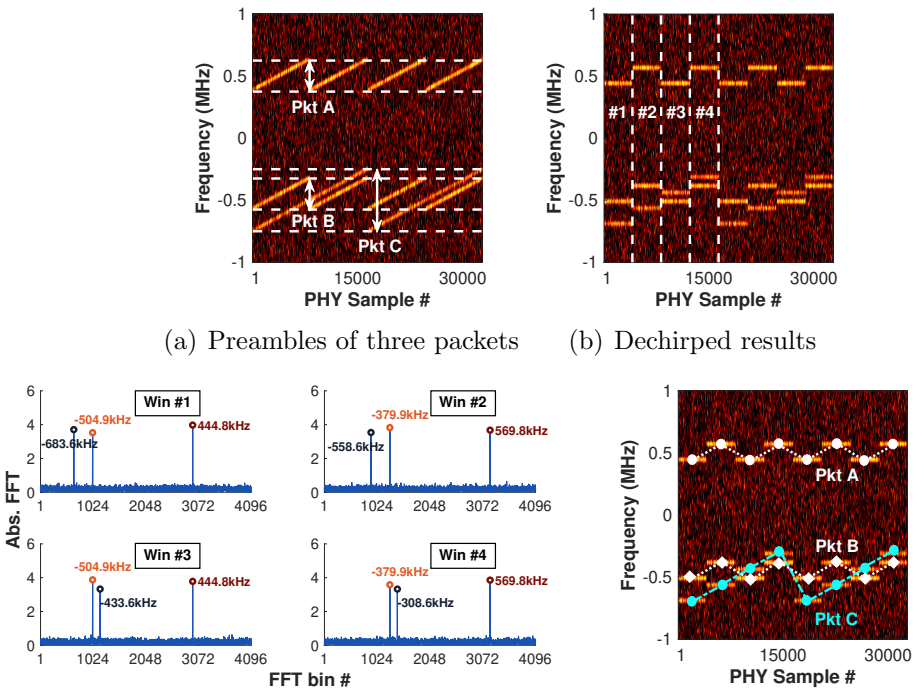
Detection of Multiple Packets

As multiple nodes transmit through different logical channels, XGate shall detect and receive all packets concurrently. Particularly, as concurrent packets have orthogonal parameters (*i.e.*, different chirp slopes), XGate could separate their signals

accordingly by using down-chirps of different slopes to detect orthogonal packets in a divide-and-conquer way. However, if the signals of packets from different logical channels have a common chirp slope, they will interfere with each other (*i.e.*, *cross-channel interference*). For example, Figure 3.5(a) presents three packets in two chirp configurations, *i.e.*, (250 kHz, SF10) and (500 kHz, SF12), which have the same chirp slope. When we dechirp the compound signals of the whole spectrum, three frequencies from different packets are detected in every window as shown in Figure 3.5(c), which adds interference to the packet detection and parameter extraction of the three packets. In this case, XGate needs to distinguish the frequencies of three packets and use the frequencies of each packet separately for correct packet detection and parameter extraction.

If two packets are apart in the frequency spectrum (*e.g.*, Pkt A in Figure 3.5(b)), we can easily separate concurrent packets by checking the ranges of detected frequencies. However, such a method may not function well if the frequencies of concurrent packets are close or overlapping each other (*e.g.*, Pkt B and C in Figure 3.5(b)). To achieve the highest communication concurrency, we expect XGate can correctly receive packets even if the packets have overlapping frequencies.

XGate leverages the signal structures of LoRa preambles to separate concurrent packets in the same chirp slope. The intuition is that the dechirped frequencies of preamble chirps from the same packet (*i.e.*, starting frequencies of preamble chirps) would change across windows at a fixed rate $\Delta f = k_i T_{win}$ that is determined by the chirp slope (k_i) and the length of a detection window (T_{win}). By contrast, the dechirped frequencies of preamble chirps from different packets will not change in Δf across windows (see Figure 3.5(c)). Based on this observation, XGate checks frequency changes across consecutive windows against Δf to track the preamble of a particular packet. XGate iteratively uses the method to disentangle preamble signals of concurrent packets. As shown in Figure 3.5(d), even though the frequencies



(a) Preambles of three packets (b) Dechirped results

(c) FFTs in consecutive windows (d) Freq. tracks of three packets

Figure 3.5: Detect concurrent packets: (a) Compound signals of packets A, B in (250 kHz, SF10) and packet C in (500 kHz, SF12); (b,c) Dechirped frequencies of signals from the same packet change at a fixed rate (Δf=125 kHz) over consecutive windows; (d) Track frequency changes to separate the three packets.

of packets B and C are overlapping, we can separately track preamble signals for the two packets and recover packet configurations from their separated frequency patterns.

To summarize, XGate can detect all concurrent packets in an Rx spectrum in the following two steps: (1) Iterate through all chirp configurations to scan signals of the Rx spectrum for detecting LoRa preambles and packets' meta-information. The concurrent packets in different chirp slopes will be detected separately in different iterations without interfere to each other due to the orthogonality of their chirp signals. (2) For concurrent packets with the same chirp slope, their signal frequencies are simultaneously detected. XGate iterates through all the detected frequencies and tracks the frequency changes across consecutive detection windows to disentan-

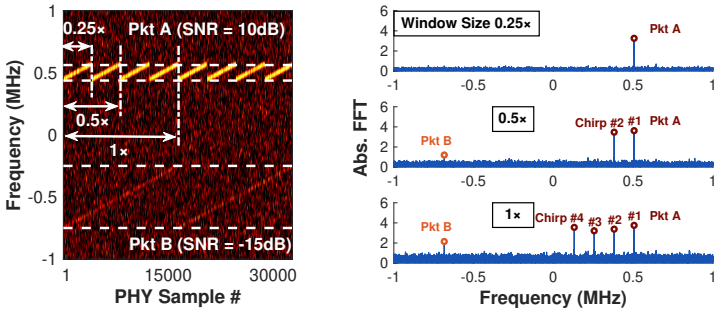
gle their colliding preambles for concurrent packet detection and meta-information extraction.

3.3.3 Enhancing Packet Sensitivity

Although a short detection window is beneficial for avoiding ambiguities of packet parameters, it can suffer *SNR losses* when detecting packets in large SFs because short windows disperse the signal power of a long chirp in multiple windows. Figure 3.6 displays two packets received from two links differing in SNRs, where Pkt A (SF8) is from a strong link and Pkt B goes through a weak link using SF12. While XGate detects packets with a short window (*e.g.*, 0.25× the chirp length of Pkt B), it detects Pkt A correctly but cannot detect Pkt B due to SNR losses caused by short windows as shown in Figure 3.6(b). We address this problem and enhance XGate to reliably detect both long-chirp and short-chirp packets.

A practical solution is to detect packets using longer windows, which can aggregate power from more samples of a chirp signal to improve detection sensitivity. As shown in Figure 3.6(b), a weak frequency of Pkt B is detected as window size enlarges from 0.25× to 0.5× of the chirp length of Pkt B, and the detected frequency becomes stronger with window size 1×. However, longer windows do not bring extra power gains for Pkt A. Instead, as window size exceeds the chirp length of Pkt A, multiple chirps of Pkt A are included in the same window and detected as distinctive frequencies (see Figure 3.6(b)), which can add more interference and cause ambiguity issues to the detection of packet parameters.

How can we deliver high sensitivity while avoiding the side effects of long detection windows (*e.g.*, interference and ambiguity issues)? We notice that although different window sizes produce different frequency results, intrinsic relationships exist among those frequencies. Specifically, for a long-chirp packet (*e.g.*, Pkt B), the detected frequencies remain invariant for different window sizes. Whereas for Pkt A, multiple



(a) Preamble signals (b) Dechirped frequencies

Figure 3.6: Use long windows to detect weak packets: (a) Received signals of two packets, *i.e.*, Pkt A in (125 kHz, SF8) and Pkt B in (500 kHz, SF12); (b) Detected results with 0.25×, 0.5×, and 1× of the chirp length of Pkt B.

short chirps are included in a long window. The frequencies from different chirps are equally spaced by a fixed interval which equals the bandwidth of the chirps, as shown in Figure 3.6(b).

Based on such observations, XGate uses the diverse detection results of multiple window sizes for mutual verification, which can help resolve interference and ambiguity issues of long-window detection. For instance, after detecting Pkt A with window size 0.25×, when we next apply window size 1× to the same signals, we assure those four frequencies are indeed from Pkt A (see Figure 3.6). We use this extra information to manually remove the frequencies of Pkt A from the raw detection results of long windows (*i.e.*, 1×) and focus on detecting the frequencies from Pkt B.

In practice, XGate scans an Rx spectrum multiple rounds to reliably detect packets in diverse SNRs and parameter configurations. XGate first detects high-SNR packets with short windows and gradually increases window sizes to search for any weak packets (*i.e.*, usually in large SFs). To eliminate the implications of high-SNR packets on weak packet reception, XGate reuses the information of high-SNR packets detected in previous short-window rounds to facilitate interference cancellation for the detection results of long windows. As a side benefit, in case some packets cannot be reliably detected due to the impacts of interference and noises in a single round

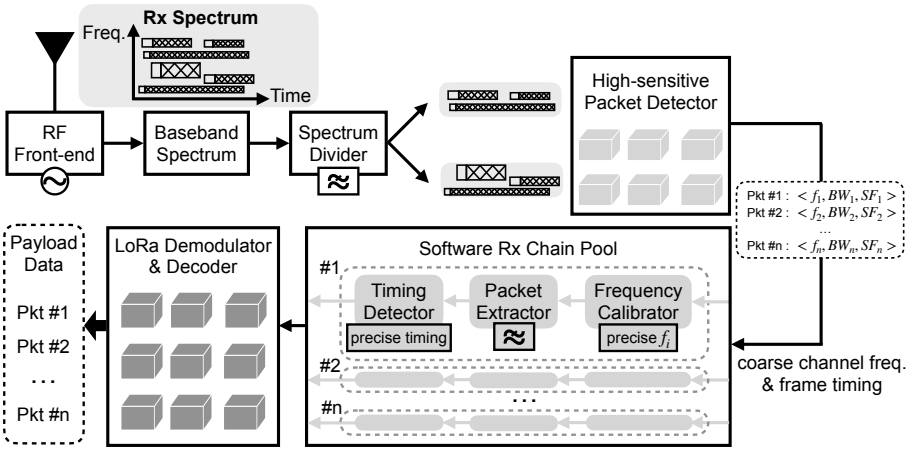


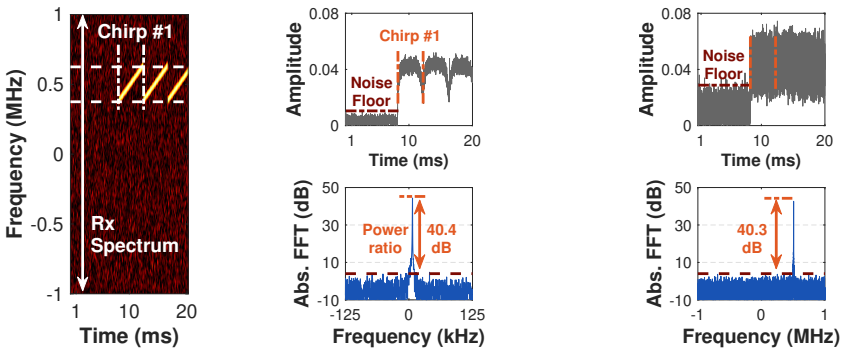
Figure 3.7: System architecture of XGate: XGate scans all available channels and automatically configures software Rx chains to receive massive LoRa packets.

(*e.g.*, Pkt B with window size 0.5× in Figure 3.6), XGate can aggregate the detection results of multiple rounds to further improve packet detection, which can effectively reduce false positive and false negative errors.

3.3.4 Massive Packet Reception

Figure 3.7 illustrates the general workflow of packet reception in XGate. The front end of RF radio receives signals of a large spectrum and down-converts them to baseband as digitized samples. XGate scans the whole Rx spectrum to detect LoRa packets and their parameters including channel frequencies, bandwidths, and spreading factors. XGate uses the detected parameters to configure software Rx chains to receive packets in parallel. The Rx chains are built and activated on-demand upon detecting new packets. A parameterized Rx chain extracts the signals of a packet from the Rx spectrum, which are finally passed to a standard LoRa decoder to demodulate and decode an incoming packet.

Note that a packet will not get received if it cannot be detected. We handle several practical issues that may impact the performance of packet detection in terms of sensitivity and accuracy, which is critical to the success of XGate.



(a) Rx signals (b) Dechirped frequencies (c) Rx signals

Figure 3.8: Use long windows to detect weak packets: (a) Received signals of two packets, *i.e.*, Pkt A in (125 kHz, SF8) and Pkt B in (500 kHz, SF12); (b) Detected results with 0.25×, 0.5×, and 1× of the chirp length of Pkt B.

Detecting large spectrum vs. small spectrum. While XGate scans the whole Rx spectrum to detect packets rather than a specific LoRa channel, a primary concern is whether the sensitivity of packet detection will be compromised since the Rx spectrum can include more noises and interference than a single LoRa channel. Figure 3.8 compares the signals of the same chirps from a packet received in a 2 MHz spectrum before and after being filtered by the target channel (250 kHz). As expected, the time-domain signal with 2 MHz spectrum has a higher noise floor than the signal with 250 kHz. Interestingly, we observe little changes of noise floors in the frequency domain after dechirping the two signals. This is because noises are uniformly distributed throughout the entire spectrum. A larger spectrum increases frequency ranges but not the noise strength per frequency bin. As XGate detects packets in the frequency domain, the detection sensitivity will not be affected much by the bandwidth of spectrum. However, we note that a large spectrum may suffer intensified cross-channel interference which can increase packet detection errors for XGate. To reduce interference, XGate partitions a larger Rx spectrum into multiple smaller chunks (*i.e.*, *Spectrum Divider* in Figure 3.7) and apply packet detectors to each small spectrum separately to detect packets.

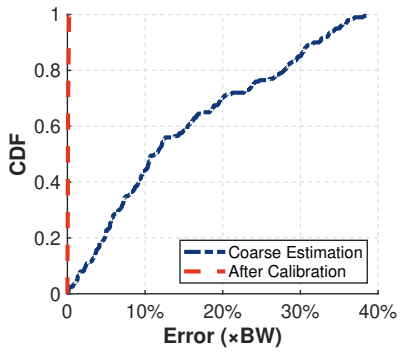


Figure 3.9: Estimation errors of packets' central frequencies before and after calibration.

Refining packet configuration. Recall that XGate detects the frequency patterns of dechirped signals to restore a packet's parameter configurations. The SF and BW of a packet are inferred from the periodicity of frequency changes across windows, which has high resilience to errors. However, the central frequency of a packet is coarsely estimated as the average of dechirped frequencies, which is error-prone since the random time offsets between detection windows and chirp signals (*i.e.*, Δt in Eq.(3.4)) can alter the estimated frequencies. Our measurements show that the coarse estimation of central frequency often deviates from the actual frequency of a packet by 5 %∼35 % of signal bandwidth (see Figure 3.9), which, if not corrected, can adversely affect the followed packet reception and decoding operations.

Our insight is that though the estimations of a packet's channel frequency are not yet accurate enough, the detected meta-information enables us to roughly identify the frame structure of a LoRa packet (*e.g.*, preamble, SFD, *etc.*). Then, by using the signal structure of a LoRa packet, we can use existing time-offset correction methods (85; 33) to mitigate the impacts of misalignment of detection windows. The rationale is that different chirps of the same packet deviate from detection windows in an identical time offset. Yet, this time offset produces opposite frequency effects on preamble up-chirps and SFD down-chirps (85; 33). We can use two coarsely located

chirps from the preamble and SFD parts of a packet, denoted by $C_{pre}(t)$ and $C_{sfd}^{-1}(t)$, to cancel out the effects of time offset as below:

$$h_i e^{j2\pi f_i t} C_{pre}(t + \Delta t_i) \cdot h_i e^{j2\pi f_i t} C_{sfd}^{-1}(t + \Delta t_i) = h_i^2 e^{j2\pi(2f_i)t}, \tag{3.6}$$

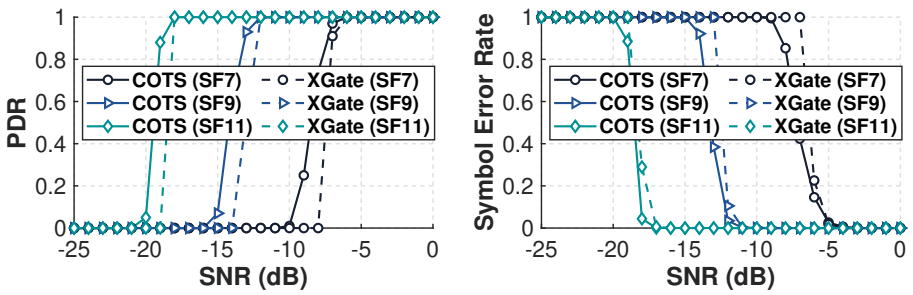
where f_i denotes the central frequency of the i^{th} packet, h_i models the impact of wireless channel, and Δt_i stands for the time offset between detection windows and chirps. The frequency component from the resulting signals of Eq.(3.6) gives the accurate central frequency of the packet (*i.e.*, f_i).

XGate uses Eq.(3.6) to refine the detection of channel frequency. The frequency mismatch between up- and down-chirps can be eliminated. As plotted in Figure 3.9, the refined results achieve almost 100 % accuracy. We note that Eq.(3.6) not only removes the effects of window time offset but also calibrates potential CFOs between sender and receiver radios. After frequency calibrations, a software Rx chain of XGate can correctly filter the interference of unwanted channels and detect precise frame timing from CFO-free signals for reliable packet demodulation and decoding (see Figure 3.7).

3.4 Implementation and Evaluation

3.4.1 Methodology

Implementation. We implement XGate on a software defined radio platform (USRP N210) based on the `gr-lora` project (86). We use USRP to receive signals from LoRa nodes and forward received samples to a workstation running XGate for processing. We use COTS LoRa nodes with Semtech SX1276 radio (82) as transmitters and use Arduino Uno to configure transmission parameters. We build a testbed consisting of 40 LoRa nodes and six gateways (*i.e.*, four COTS gateways and two USRP gateways). We conduct experiments on our campus spanning 1.08 km × 1.2 km that represents a typical urban environment as illustrated in Figure 3.13(a).



(a) Packet detection (b) Packet decoding

Figure 3.10: Performance of single packet reception

The gateways are mounted on the roof of a 20-storey building. All nodes operate in the 915 MHz ISM band.

Experiment Setup. We conduct extensive experiments to evaluate XGate, aiming to answer the following questions: (1) How many concurrent LoRa transmissions can be supported by XGate? (§3.4.2) and (2) How does XGate perform with practical network settings? (§3.4.3)

We collect over 10,000 packet traces from more than 200 links in our testbed. The collected data covers various channel conditions (*e.g.*, indoor and outdoor, low and high SNRs). All evaluations with <40 nodes are conducted via real-world experiments in the testbed. For performance evaluation of massive concurrent transmissions (*e.g.*, thousands of LoRa nodes), we synthesize the traffic based on the collected traces. Specifically, we add up multiple packet traces with random time offsets to emulate real-world network traffic. The synthesized traces are then replayed to evaluate the performance of XGate when more than 40 nodes are involved.

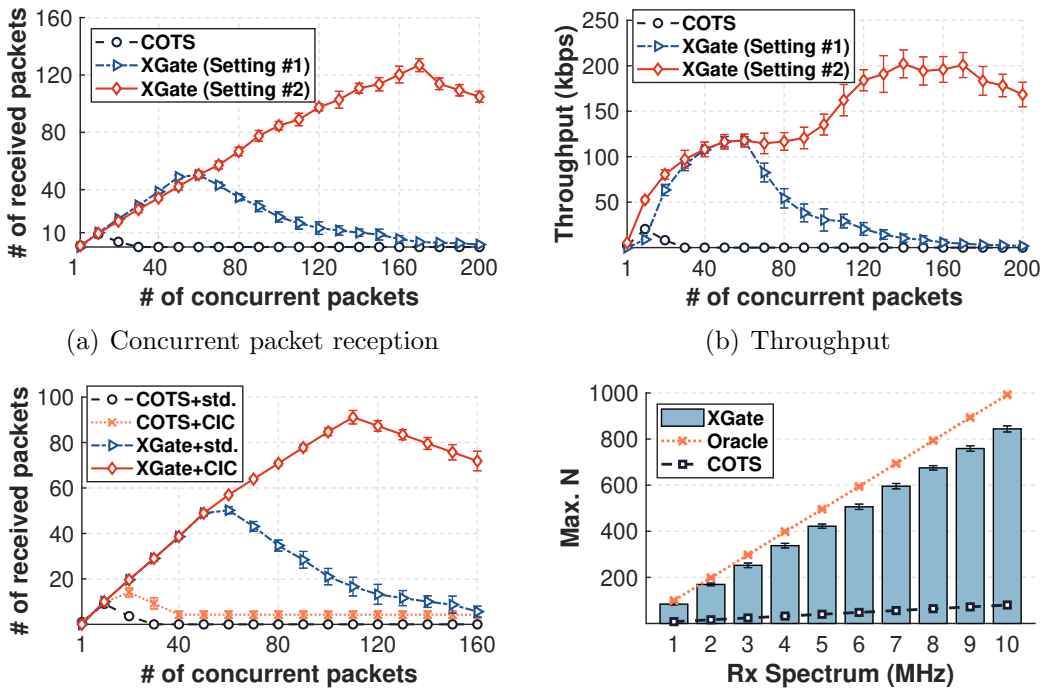
Baseline. We compare XGate with two baselines: (1) *COTS gateways* (1) receive LoRa packets with multiple Rx chains concurrently. The Rx chains of COTS gateways are pre-configured; all nodes communicate in fixed channels supported by the Rx chains of COTS gateway. (2) *CIC* (2) adopts collision resolving technique to receive concurrent LoRa transmissions in the same LoRa channel.

3.4.2 Basic Reception Performance

Packet detection & decoding. We first evaluate XGate in detecting and receiving packets from LoRa logical channels. We set up one gateway to receive packets from one LoRa node under different SNR conditions. The Rx spectrum of gateway is 1.6 MHz. The LoRa node randomly chooses a channel (*i.e.*, central frequency) for every packet with a default bandwidth of 125 kHz. We analyzed >100 packets for each SNR condition. XGate detects and receives a packet from the Rx spectrum without knowing the channel or parameter configuration. For comparison, we use a COTS LoRa gateway, which knows the channels and parameter configurations of all packets in advance.

Figure 3.10 compares the packet detection ratios and decoding errors of XGate with a COTS gateway. We observe that even without the prior knowledge of channel or parameter configuration, XGate achieves comparable performance in packet detection and decoding to that of a COTS gateway which knows all the meta-information in advance. XGate can reliably detect a packet from unknown logical channels in ultra-low SNRs (*e.g.*, -12.4 dB for SF9 and -18.2 dB for SF11), and use detected parameters to decode incoming packets correctly. Compared to a COTS gateway, the absence of channel information in XGate only leads to 0.3 dB $\sim$ 0.6 dB SNR losses on packet detection and decoding while achieving >80 % packet detection ratio and <20 % symbol error rate.

Gains on concurrency. We next evaluate the performance of XGate in supporting LoRa concurrent transmissions. We compare XGate against a COTS LoRa gateway. We configure both XGate and the COTS gateway to operate within the same Rx spectrum of 1.6 MHz. We control a varying number of LoRa nodes to transmit concurrently on different LoRa channels. The COTS gateway (SX1301) has nine Rx chains that receive at nine frequencies distributed equally in the Rx spectrum with



(a) Concurrent packet reception

(b) Throughput

(c) Comparison with SOTA

(d) Concurrency vs. Rx Bandwidth

Figure 3.11: Performance of concurrent reception: (a,b) different numbers of concurrent transmissions in a 1.6 MHz spectrum; (c) comparison with state-of-the-art; (d) concurrency under different spectrum bandwidths.

a fixed 125 kHz bandwidth. We adopt two settings of channel partition for XGate. For Setting #1, XGate uses the same channel partition scheme as in the COTS gateway. By varying SF from 7 to 12, 54 LoRa logical channels are supported. Different from COTS gateway detecting and receiving packets on channels known in advance, XGate detects the channel and parameter configurations of packets on-the-fly. For Setting #2, XGate supports all LoRa logical channels in a 1.6 MHz spectrum. LoRa nodes randomly choose packet parameters from those supported by a COTS radio (82). In particular, we test four bandwidths (*i.e.*, 62.5 kHz, 125 kHz, 250 kHz and 500 kHz) and seven SFs (*i.e.*, 6∼12). The total number of LoRa logical channels is 168.

Figure 3.11(a) plots the number of received packets when different numbers of

nodes transmit concurrently. The number of received packets first increases as more nodes transmit concurrently, then reaches the maximum and starts to drop when the number of concurrent nodes exceeds the number of available channels because of multiple packets colliding in the same channel. As expected, a COTS gateway can receive nine concurrent packets at most, *i.e.*, one packet from each Rx chain. By contrast, XGate (Setting #1) can receive up to 51 concurrent packets, *i.e.*, one packet per logical channel. This number increases to 126 for XGate (Setting #2), *i.e.*, 14× higher than the COTS gateway. The performance gain in concurrency is mainly because XGate can support more LoRa logical channels.

Figure 3.11(b) compares the aggregate network throughput of XGate with a COTS gateway. The highest throughput of a COTS gateway is 20.4 kbps on average when nine packets are transmitted concurrently. XGate can achieve a maximum throughput of 118 kbps in Setting #1 and 200 kbps in Setting #2, which are 5.8× and 9.8× higher than a COTS gateway.

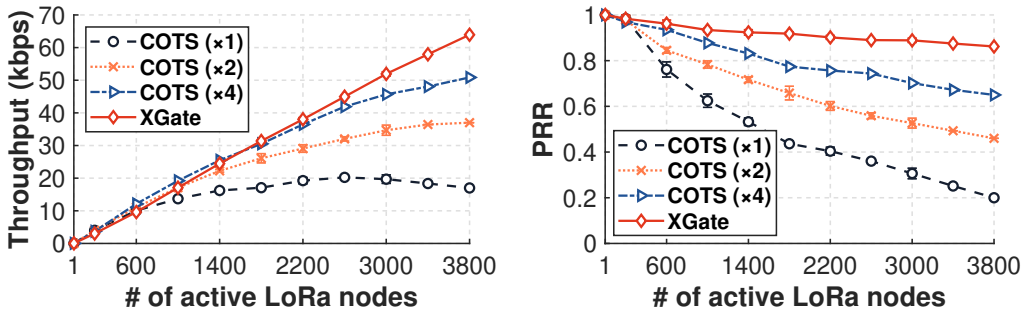
Comparison with the state-of-the-art. This experiment compares XGate with existing LoRa concurrent transmission strategies. We consider two state-of-the-art (SOTA) strategies (*i.e.*, COTS gateway (1) and CIC (2)) and evaluate four schemes that combine different gateway paradigms and packet decoders, *i.e.*, COTS gateway with a standard LoRa decoder (*COTS+std.*), COTS gateway with a CIC decoder (*COTS+CIC*), XGate (Setting #1) with a standard LoRa decoder (*XGate+std.*), and XGate (Setting #1) with a CIC decoder (*XGate+CIC*). We control various numbers of LoRa nodes to transmit concurrently on different channels using the same method as in the former experiment. When a COTS gateway is used, only nine concurrent channels (*i.e.*, one channel per Rx chain) are available in a 1.6 MHz spectrum, while XGate supports 54 logical channels in the same spectrum.

Figure 3.11(c) presents the number of received packets of the four schemes under different numbers of concurrent transmissions. As expected, a COTS gateway with

the standard LoRa decoder receives nine concurrent packets at most, *i.e.*, one packet per Rx chain. When more than nine nodes transmit concurrently, collisions will occur on some channels, resulting in packet loss. CIC recovers two packets from collisions on average. It increases the maximum concurrency of a COTS gateway to 15 packets. This number is slightly less than that reported in OpenLoRa (34) because the bandwidth of channels in a COTS gateway is 125 kHz, while the bandwidth is 250 kHz in (34). Unlike CIC , XGate increases concurrency in the dimension of LoRa logical channels. XGate with a standard decoder receives 51 concurrent packets in the same spectrum without collisions, which is 3.4× higher than the concurrency of the compound SOTA scheme (*i.e.*, CIC's parallel decoding atop multiple Rx chains of a COTS gateway). Moreover, XGate can jointly work with CIC to support up to 91 concurrent transmissions (*i.e.*, 78 % more than XGate with a standard LoRa decoder) as shown in Figure 3.11(c).

Scaling with Rx bandwidth. This experiment evaluates the maximum number of concurrent transmissions supported by XGate under different Rx spectrum settings. We increase the Rx spectrum of a gateway from 1 MHz to 10 MHz and set up various LoRa nodes to transmit concurrently using the same method as XGate (Setting #2) in the former experiment. We compare XGate with two benchmarks: (1) COTS gateway that uses dedicated Rx chains to receive pre-configured channels (*i.e.*, eight 125 kHz LoRa channels per 1 MHz); and (2) Oracle method that can divide a given spectrum into the maximum number of logical channels and know the parameters of all packets in advance.

We measure the maximum number of concurrent transmissions and present the results in Figure 3.11(d). The maximum concurrency increases with the bandwidth of Rx spectrum for all three strategies. The COTS method increases the slowest, and the Oracle increases the fastest. XGate matches closely with Oracle, *i.e.*, within 85 %. Specifically, when spectrum bandwidth increases to 10 MHz, XGate receives



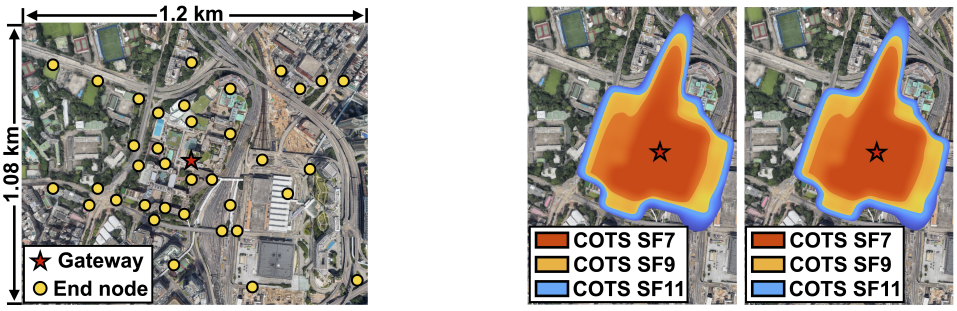
(a) Throughput (b) Packet Reception Ratio

Figure 3.12: Performance of massive packet reception.

844 concurrent packets which is 10.5× higher than the COTS gateway method. The results indicate that XGate can support higher concurrency with an increased Rx spectrum.

Massive reception performance. This experiment evaluates XGate in supporting massive IoT communications in practical network settings. In particular, we deploy LoRa gateways in our testbed area (1.08 km×1.2 km) to cover up to 3,800 IoT sensors using a 4 MHz ISM spectrum. Each IoT sensor transmits a 20-Byte message every 30 min with a duty cycle ⩽1%. We use a trace-driven method to investigate communications for thousands of sensors. Specifically, we measure LoRa packet profiles from more than 200 sites with SNR ranging from −20 dB to 10 dB and use the collected data to synthesize the signals of received packets with randomly selected link profiles. We consider using a single XGate gateway with a 4 MHz Rx spectrum or various COTS gateways with each covering 1 MHz only. Sensors can randomly choose channel frequency (in the 4 MHz spectrum), bandwidth (62.5 kHz, 125 kHz, 250 kHz and 500 kHz) and spreading factor (SF6∼12) for every transmission when XGate is used. We use COTS gateways for benchmarking. LoRa nodes select only from the pre-configured channels (*i.e.*, nine channels per COTS gateway) to transmit messages with an ALOHA-based MAC.

Figure 3.12 compares the aggregated throughput and packet reception ratio (PRR)



(a) Campus-scale testbed (b) Coverage comparison

Figure 3.13: (a) Deployment map of testbed (each dot represents multiple surrounding sites); (b) Coverage comparison between XGate and a COTS gateway.

of XGate and COTS gateways under different numbers of active nodes. We see that the throughput of a COTS gateway first increases and then becomes saturated when the number of nodes exceeds 200. As more nodes transmit actively, it increases the chance of collisions and decreases PRRs (Figure 3.12(b)). A COTS gateway can only support 400 nodes with PRR>80 %. When four COTS gateways are used, up to 1600 nodes can transmit actively with PRR>80 %. By contrast, XGate can use fewer gateways to serve more active nodes with even higher PRRs because many LoRa logical channels are used to reduce potential collisions. As shown in Figure 3.12, a single XGate gateway can support 3,800 active nodes with PRR>85 % in the same 4 MHz spectrum, achieving a throughput close to 64 kbps.

Gateway coverage. This experiment compares the coverage of XGate with that of a COTS gateway in an urban environment. To obtain the coverage maps, we divide our testbed area as shown in Figure 3.13(a) into 40×36 grids. We put LoRa nodes in each grid to send probe packets at 915 MHz with a default bandwidth of 125 kHz and fixed SFs (*i.e.*, 7, 9, and 11). We measure the SNRs of LoRa packets transmitted from each grid and plot a grid in colors if packets from the grid can be reliably received (*e.g.*, PRR>90 %). Figure 3.13(b) visualizes the coverage area of a COTS gateway and XGate with different spreading factors. We see that XGate

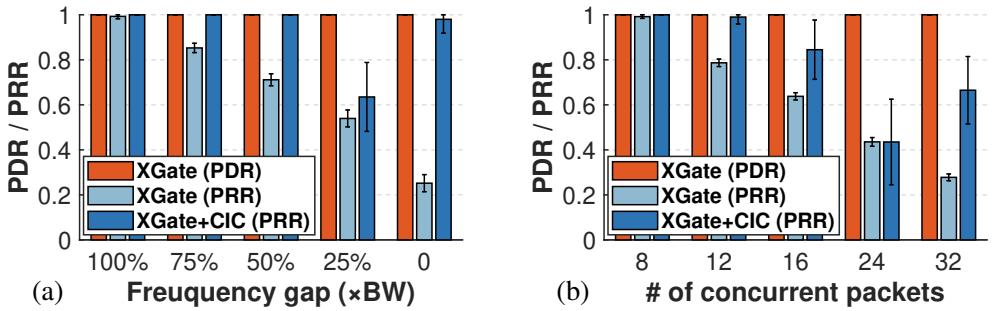


Figure 3.14: Impacts of cross-channel interference under (a) two packets with different frequency gaps, and (b) different numbers of concurrent packets.

achieves almost the same coverage as the COTS gateway. Due to the severe signal blockage and power loss in our deployment area, the maximum communication ranges are 286 m, 340 m and 406 m for SF7, SF9 and SF11, respectively. The differences in communication ranges are smaller than 20 m between XGate and a COTS gateway.

3.4.3 Microbenchmarks

Impact of cross-channel interference. We conduct experiments to evaluate the impacts of cross-channel interference on XGate's performance. We first study the impact of frequency gaps between concurrent packets. We set up two LoRa nodes transmitting concurrently with the same parameters (BW 125 kHz, SF9). Both nodes have good SNRs (>0 dB). We change the central frequencies of two LoRa nodes and measure the PDR and PRR of XGate under various frequency gaps. As shown in Figure 3.14(a), XGate can reliably detect all packets in all frequency settings. However, some packets cannot be correctly decoded when concurrent transmissions are overlapped in frequency (*i.e.*, frequency gap < BW). In this case, the interference between overlapped packets can add errors to packet decoding, *i.e.*, similar to packet collisions. Such decoding errors of two concurrent packets can usually be recovered by a parallel LoRa decoder (*e.g.*, CIC), as marked by XGate+CIC in Figure 3.14(a).

Impact of detection window. This experiment evaluates how the setting of

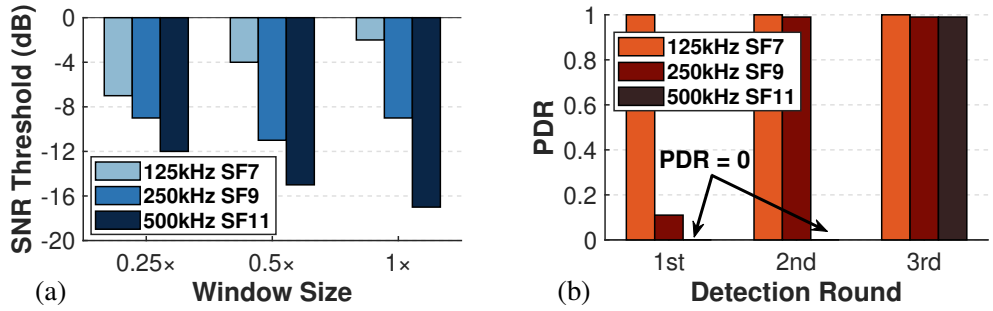


Figure 3.15: Impact of window size on (a) single packet and (b) concurrent packet detection.

detection windows in XGate affects the performance of packet detection. We set up three LoRa nodes and a gateway in the testbed. The SNRs of links from the three nodes to the gateway are -2 dB, -10 dB, and -16 dB, respectively. The BW and SF configurations of the three nodes are (125 kHz, SF7), (250 kHz, SF9), and (500 kHz, SF11). We employ three types of detection windows for XGate, *i.e.*, 0.25×, 0.5×, and 1× of the chirp length of (500 kHz, SF11).

We first examine the packet detection capability of XGate with different detection windows. We control the three nodes to transmit individually with corresponding BW and SF, and change signal SNRs by varying the transmit power of LoRa nodes. We measure the minimum SNRs (*i.e.*, SNR threshold) of packets that XGate can detect reliably. Figure 3.15(a) plots the results. We see that XGate generally gives the highest sensitivity when the length of detection windows equals the chirp length of a packet. For instance, the chirp length of a packet in (125 kHz, SF7) is 0.25× of the chirp length of (500 kHz, SF11). XGate, with a detection window of 0.25×, detects a packet of (125 kHz, SF7) at the highest sensitivity (*i.e.*, −7 dB). For the packets of (500 kHz, SF11), the SNR sensitivity increases (*e.g.*, from −12 dB to −17 dB) as detection window enlarges from 0.25× to 1× of the chirp length. Whereas for packets in (250 kHz, SF9), the SNR threshold first decreases and then increases. The optimal sensitivity (*i.e.*, −11 dB) is produced by window size 0.5×.

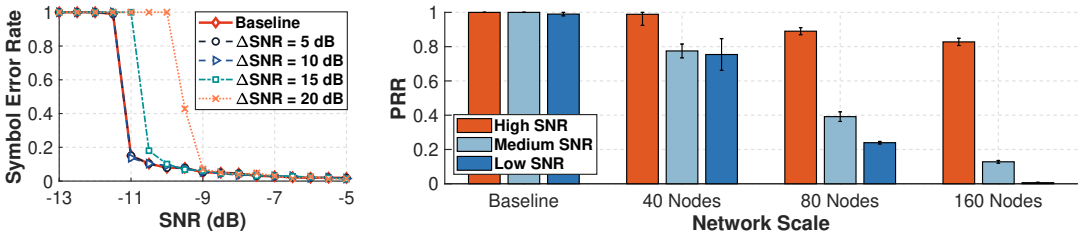


Figure 3.16: Near-far effect of (a) two nodes and (b) different network scales.

Next, we evaluate the full packet detection scheme of XGate, which detects packets in multiple rounds using different windows. We control the three nodes to transmit concurrently on three logical channels. XGate is configured to use three types of windows (*i.e.*, 0.25×, 0.5×, and 1×) to detect packets from all possible logical channels in three rounds. Figure 3.15(b) plots the packet detection ratios (PDRs) of the three nodes after each round of detection. As expected, the packets of (125 kHz, SF7) are detected with 100 % PDR in the first round, whereas packets from the other two nodes cannot be detected due to poor link qualities. The packets of (250 kHz, SF9), and (500 kHz, SF11) are gradually detected in the second and third rounds, as longer detection windows (*e.g.*, 0.5× and 1×) are adopted.

Near-far effect. In practice, concurrent packets may differ in SNRs due to the different distances and heterogeneous link qualities of LoRa nodes (*i.e.*, near-far effects (14)). This experiment investigates the impact of SNR differences between concurrent packets on XGate performance. We set up two LoRa nodes (*i.e.*, one node in higher Tx power and the other in lower power) to transmit concurrently. Both nodes use a default BW 125 kHz. We configure SF7 for the strong node and SF9 for the weak node. In particular, we maintain a fixed SNR difference (*i.e.*, ΔSNR) between the two nodes. We change SNRs of the weak node from −13 dB to −5 dB and measure Symbol Error Rates (SERs) of both nodes under different SNR conditions (SERs of strong packets are not displayed). We also measure SERs from the same packets of the weak node with the strong node absent and use the results

for baseline comparison.

Figure 3.16(a) plots the SERs of weak packet reception in various ΔSNR settings. We see that the SER remains the same as the baseline when ΔSNR$\leqslant$10 dB. It means that XGate can correctly detect and receive weak packets as if no strong packets are present when the SNR difference between concurrent packets is smaller than 10 dB. As the SNR difference increases to 15 dB and 20 dB, though the concurrent packets use orthogonal SFs, a strong packet still increases the background noise power of a weak packet which in turn leads to higher SERs for weak packet reception (see Figure 3.16(a)).

We next evaluate the near-far effect when more nodes are transmitting simultaneously. We test three network scales with 40, 80, and 120 nodes. All nodes are evenly distributed in the testbed area, connecting to the gateway with SNRs ranging from -5 dB to 5 dB. We set up gateways with the channel plan Setting #2. Nodes select channels and SF settings based on their SNRs, *i.e.*, large (low) SFs are used by nodes with low (high) SNRs. We group nodes into three SNR regimes, *i.e.*, high SNR (2∼5 dB), medium SNR (-2∼2 dB), and low SNR (-5∼-2 dB). To investigate the near-far effects, we control all nodes to transmit concurrently and measure the packet reception performance of nodes in different SNR regimes. We also measure the packet reception performance of nodes in each SNR regime when they are communicating individually (*i.e.*, without interference from any concurrent nodes) and use the results for baseline comparisons.

Figure 3.16(b) illustrates the PRR of three SNR regimes. As expected, the PRRs of nodes in each SNR regime are close to 100 % when the nodes communicate individually. The PRRs of packets in high SNRs drop slightly as concurrent transmissions increase but remain higher than 80 % when 160 nodes transmit concurrently. In contrast, the PRRs of medium- and low-SNR packets drop dramatically as the number of concurrent nodes increases to 80 and 160. The reason is although different nodes

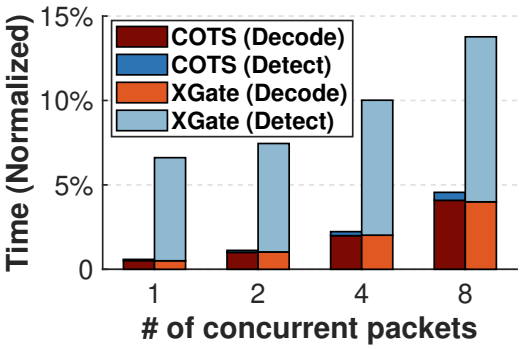


Figure 3.17: Time Overhead.

transmit packets in different logical channels, the concurrent high-SNR packets increase background noises for the medium- and low-SNR packets coexisting in the same spectrum, which affects the reception performance of medium- and low-SNR packets. The more high-SNR packets, the lower PRRs of medium- and low-SNR packets.

Real-time performance. This experiment evaluates the time overheads of XGate. We run XGate on a workstation to process different numbers of concurrent packets (BW 125 kHz, SF9) and measure the time spent on packet detection and decoding. We use the same number of demodulators and decoders for COTS gateway and XGate under each experiment setting. We normalize the processing time to percentages of the air-time of a packet. We also measure the time costs of corresponding operations in a COTS gateway for benchmark comparison. As presented in Figure 3.17, the costs of packet decoding are comparable for XGate and a COTS gateway. The major overhead of XGate comes from packet detection. In particular, the overall processing time of eight packets in XGate only accounts for $<15\,\%$ of a packet duration. The result indicates that both packet detection and decoding operations of XGate can be completed in real-time.

System Name	Multi. logical ch.	# of logical ch.	Concur. per logical ch.	Overall Concur.	Method.
FTrack (31)	✗	1	<2	<2	Collision Recovery (CR) in ONE logical ch.
CoLoRa (67)	✗				
NScale (32)	✗				
CIC (2)	✗	1	2∼3	2∼3	
COTS (1)	✓	9/1.6MHz	1	9	H/w Rx chains
COTS+CIC	✓	9/1.6MHz	1.67	18∼27((34)) 15(Our)	H/w Rx chains & CR
XGate	✓	**168/1.6MHz**	1	**126**	Auto-config & S/w Rx chains

Table 3.1: Comparisons between XGate and SoTA.

3.5 Discussion

XGate vs. SoTA. Table 3.1 compares XGate with the state-of-the-art. The concurrency data of FTrack, CoLoRa, NScale, and CIC are from (34). Different from COTS gateways that use multiple Rx chains to receive concurrent channels, XGate uses a single Rx chain to detect and receive packets from all LoRa logical channels. Existing parallel decoding techniques resolve LoRa packet collisions to achieve concurrent transmissions in one logical channel. Though it is feasible to apply parallel decoding to each Rx channel of a COTS gateway (*e.g.*, COTS+CIC), the overall concurrency remains low (*e.g.*, <30), limited by parallel decoding capability. In contrast, XGate scales concurrency across hundreds of logical channels.

Overheads of XGate. The key advantage of XGate is that it fully covers all LoRa logical channels within an Rx spectrum. The packet reception of XGate can be configured on-the-fly via software, which has better flexibility and supports higher concurrency. XGate incurs higher computational and energy overheads associated with wider bandwidth reception (*e.g.*, MHz-level), detection of concurrent packets

and meta-information, *etc.*. We notice XGate suffers slight SNR losses (*e.g.*, near 0.6 dB) due to the lack of packet meta-information. We believe it is worth the trade-off considering the increased flexibility and scalability.

Logical channel selection. XGate unlocks a broad space of channel configurations for LoRa nodes, which allows a node to freely choose channels in an Rx spectrum without negotiations with a gateway (*i.e.*, come-and-be-served). XGate automatically detects the parameters of an incoming packet and configures software Rx chains on-the-fly to receive the packet. In practice, a node can select channels properly based on the link states and workloads of logical channels.

3.6 Conclusion

This chapter presents XGate, a novel gateway paradigm for LoRaWAN that revolutionizes the way packets are detected and received. Unlike conventional approaches, XGate utilizes a single Rx chain to efficiently scan the entire Rx spectrum, detect incoming packets across thousands of logical channels, and dynamically configure resources on-the-fly, such as software-controlled Rx chains and packet demodulators, to receive all detected packets. This breakthrough enables scalable LoRa concurrent transmissions across all available logical channels, fundamentally breaking the barrier of massive LoRa communications. Importantly, XGate opens up new possibilities for LoRaWAN communication and networking paradigms, facilitating dynamic channel configuration, data-rate adaptation, spectrum sharing, and more.

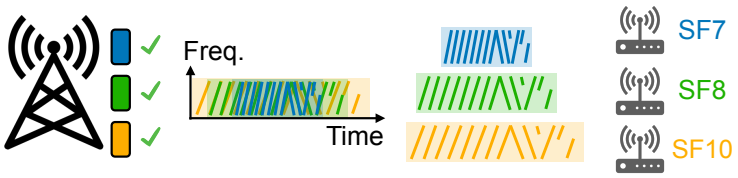
Chapter 4

Canas: Resolving Inter-Logical Channel Interference for Large-scale LoRa Deployments

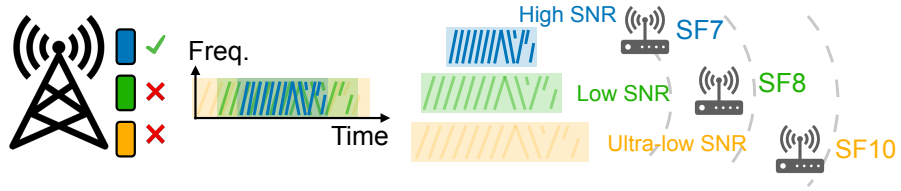
4.1 Introduction

Recent years have witnessed Low-Power Wide-Area Networks (LPWANs) as a promising wireless technology to enable ubiquitous connectivity for wireless sensing devices and empower various mobile services and applications. As a leading LPWAN technology, LoRa stands out with its extensive coverage and energy-efficient communication capabilities, making it suitable for Internet of Things (IoT) applications (87; 88; 89; 90; 91; 92; 93; 13; 94; 95; 96; 97; 98; 99; 100; 101; 102; 103) such as smart agriculture (104; 105; 36), smart metering (79; 25), and satellite (106; 107; 108). According to industry reports (109), the LoRaWAN ecosystem has already connected over 350 million LoRa nodes and 6.9 million gateways globally.

LoRa networks can service vast physical areas at the kilometer level, covering massive end nodes with link SNRs ranging from $-15\,\mathrm{dB}$ to $20\,\mathrm{dB}$ (29). To fully harness the spectrum for massive IoT communications, LoRa adopts diverse and flexible channel configurations. The LoRa spectrum can be divided into many physically separated channels (*e.g.*, 208 channels in US). Within each channel, LoRa



(a) Ideal LoRaWAN: orthogonal logical channels



(b) Practical LoRaWAN: inter-logical channel interference

Figure 4.1: The orthogonality between logical channels can be broken because of *inter-logical channel interference*. The research problem of how to resolve the inter-logical channel interference remains unexplored.

supports overlapping *logical channels* by modulating packets with different spreading factors (*e.g.*, SF6∼12). Owing to the SF-selective LoRa demodulation, signals with different SFs can be separated by the receiver even if they physically overlap with each other. Therefore, overlapping logical channels are traditionally deemed "orthogonal" in LoRa networks. Such orthogonality is pivotal to greatly facilitate wireless access without channel activity detection, enable flexible rate adaptation of individual nodes, and allow operators to scale LoRa networks.

However, large-scale LoRa networks encounter practical challenges due to the *imperfect orthogonality between logical channels*. Our measurements (§4.2) discover an unexpected finding: concurrent transmissions across logical channels presumed orthogonal (which dominate the concurrent traffic in large-scale LoRa deployments), could suffer 61% packet loss due to the imperfect orthogonality. Many ultra-weak packets from distant LoRa nodes are lost due to interference from concurrent logical channels, leading to serious starvation and power depletion of weak nodes. Although much research effort has been dedicated to supporting concurrent transmissions, the

previous research mainly focused on resolving explicit collisions for single-channel (31; 32; 67; 2) or cross-channels (3; 110) scenarios. The interference among logical channels has been largely overlooked and under-explored. Given that numerous LoRa nodes often use different logical channels for concurrent transmissions (42), the lack of reliable orthogonality between logical channels presents a significant scalability challenge for LoRa networks. In this chapter, we ask the following research question: *Is it possible to resolve inter-logical channel interference to practically orthogonalize massive logical channels for concurrent transmission?*

To answer this question, we first investigate the existing strategies to receive concurrent logical channels in LoRaWAN. Existing works primarily rely on the SF-specific LoRa demodulation to separate overlapping logical channels. When a LoRa receiver targets at a specific logical channel, the packets over that logical channel will benefit from a distinct demodulation gain (*e.g.*, 3∼17 dB (82)), while the signals over adjacent logical channels will be suppressed. Nevertheless, the signals from other logical channels, though not demodulated, could physically overlap with the target channel and increase the demodulation noise floor (*i.e.*, termed *logical channel interference*[*]). The problem becomes more challenging in large-scale LoRa deployments. Because of the high concurrency and near-far effect, we notice the increased noise floor can affect and even overwhelm the SF-selective demodulation gain, thereby breaking the logical channel orthogonality.

In this chapter, we present *Canas*, the first gateway design that cancels the mutual interference between logical channels for concurrent LoRa transmissions. At the core of *Canas* is its novel logical channel interference cancellation technique that suppresses the signal of non-targeted logical channels down to the noise floor. We achieve this by precisely reconstructing the raw signal of non-target logical channels and removing them from the Rx signal before demodulation of target packets. By

[*]used interchangeably with "inter-logical channel interference" in this chapter

doing so, *Canas* enhances the orthogonality of logical channels and better supports LoRa communications over massive logical channels in large-scale LoRa deployments.

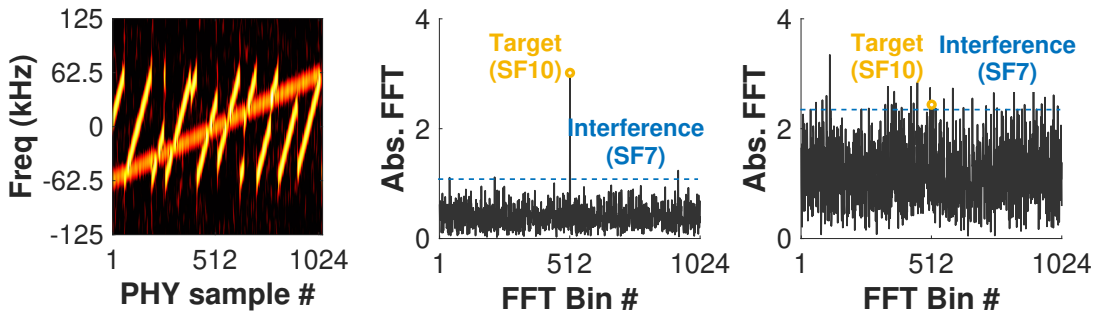
However, turning this basic idea into a practical system faces substantial challenges. Unlike traditional gateways that directly demodulate a target logical channel, *Canas* aims to eliminate other interfering logical channels before demodulation. Since the concurrent logical channels overlap, it is non-trivial to extract the signal of an individual logical channel, let alone subtract it from the Rx signal. Though previous works (35) have demonstrated the effectiveness of strong signal reconstruction and cancellation for other wireless technologies (*e.g.*, Wi-Fi), these solutions cannot be applied in LoRa due to two practical challenges: Firstly, the traditional methods (*e.g.*, Zig-zag decoding (35)) require a clean strong signal chunk to bootstrap signal reconstruction and cancellation process. In contrast, under concurrent logical channels, the concurrent signals can overlap and distort the amplitude and phase of each logical channel, making it challenging to obtain the meta-information of a specific ultra-weak logical channel without being influenced by others. Moreover, ultra-weak LoRa packets are also subject to various hardware imperfections caused by their unique long airtime, low-cost Tx hardware modules, and unreliable power supply. These imperfections increase the randomness and unpredictability of interference signals, which pose significant obstacles to accurate interference signal reconstruction and cancellation. Note that existing techniques, such as frequency offset cancellation methods, are primarily designed for packet reception rather than interference signal reconstruction and thus cannot be directly applied to tackle the obstacles. That is because packet reception only needs to figure out transmitted bits, while interference cancellation requires precise reconstruction of physical samples corresponding to their waveforms. Therefore, achieving logical channel interference cancellation over massive LoRa logical channels remains challenging.

Canas presents a novel design to practically orthogonalize massive logical chan-

nels by precisely reconstructing and canceling the interference of ultra-weak LoRa signals. We find that orthogonality between logical channels is compromised when the noise floor of a weak logical channel is elevated by stronger ones. Conversely, the strong logical channels are less affected by the weaker ones. Based on this observation, it is feasible to clearly demodulate the relatively strong logical channel and translate the payload bits to its raw signal. To achieve accurate raw signal reconstruction, *Canas* employs novel techniques to capture various signal offsets of the target logical channel without being affected by others. These meta-information capturing the impact of various factors (*e.g.*, hardware imperfections, air-channel, *etc.*) are then emulated to faithfully reconstruct the raw interference signals. Subsequently, we can subtract and cancel the strong interference from the Rx signal. This process can be iterated to receive all concurrent logical channels, progressively from the strongest to the weakest.

We implement and evaluate *Canas* with commercial off-the-shelf (COTS) LoRa devices and Software Defined Radio (SDR). Results demonstrate that *Canas* significantly outperforms existing techniques adapted to cancel interference among logical channels. Specifically, *Canas* achieves 2.3× concurrency gain on packet reception across logical channels than the best known baselines by enhancing the orthogonality between logical channels. *Canas* can be seamlessly integrated with the existing LoRa gateways with software modification. In summary, our work makes the following contributions:

- For the first time, we draw attention to the problem of inter-logical channel interference in large-scale LoRa deployments, emphasizing its uniqueness and how it differs from previously studied problems (§4.3).
- We conduct a comprehensive analysis of the inter-logical channel interference problem, supported by extensive measurements, to illustrate its implications,



(a) Raw signal (b) ΔSNR = 3 dB (c) ΔSNR = 10 dB

Figure 4.2: Demodulation under logical channel interference: (a) Spectra of concurrent logical channels; (b) Demodulation of SF10 logical channel; (c) Orthogonality is broken by the increased noise floor.

practical impacts, and root causes (§4.2).

- We present *Canas* to effectively orthogonalize massive LoRa logical channels. *Canas* employs innovative techniques to overcome the unique challenges involved in precise reconstruction and iterative cancellation of ultra-weak LoRa signals (§4.4).

- We implement and evaluate *Canas* on an outdoor testbed. Our evaluations demonstrate significant improvements in concurrent transmission across logical channels compared to the best baselines, as well as seamless integration and enhancement of packet reception in existing systems under practical near-far effects (§4.5).

4.2 Problem and Motivation

4.2.1 LoRa Logical Channels

LoRa leverages chirp spread spectrum (CSS) modulation as the physical layer technique to enable its long-range communication capabilities while maintaining low power consumption (111; 81; 112; 113; 114; 115). CSS uses chirps to modulate sym-

bols. The duration of a chirp is determined by two key LoRa parameters: Spreading Factor (SF) and Bandwidth (BW). Specifically, the chirp duration T is calculated as $T = \frac{2^{SF}}{BW}$. A *base up-chirp*, whose frequency linearly increases from $-\frac{BW}{2}$ to $\frac{BW}{2}$ over the chirp duration, can be represented as:

$$C(k,t) = e^{j2\pi(\frac{1}{2}kt - \frac{BW}{2})t} \tag{4.1}$$

where the chirp slope $k = \frac{BW}{T}$ denotes the rate of frequency change over time.

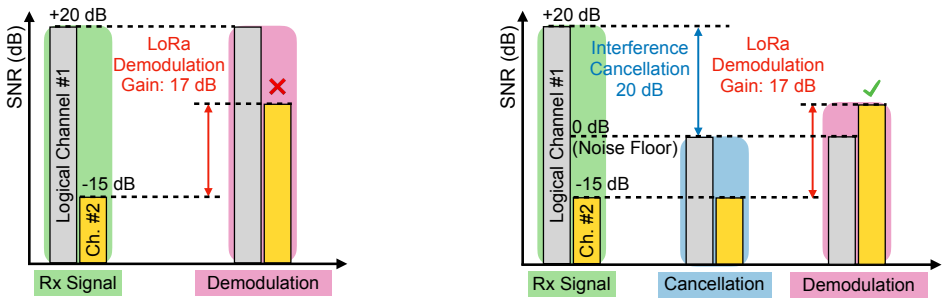
LoRa modulates symbols by changing the initial frequency of the base up-chirp. A modulated symbol $S(f_{sym}, k, t)$ can be represented as:

$$S(f_{sym}, k, t) = C(k,t)\ e^{j2\pi f_{sym} t} \tag{4.2}$$

The frequency shift is in a cyclic manner, where the frequencies higher than $\frac{BW}{2}$ will align to $-\frac{BW}{2}$. To demodulate the received symbol, the LoRa receiver performs *dechirp* by multiplying $S(f_{sym}, k, t)$ with the conjugate of the base up-chirp, denoted as $C^{-1}(k,t)$. The operation is represented as:

$$S(f_{sym}, k, t)\ C^{-1}(k,t) = e^{j2\pi f_{sym} t} \tag{4.3}$$

The *dechirp* operation removes the CSS modulation and converts the LoRa symbol into a single tone. Importantly, the energy of the symbol, which was initially spread across the chirp bandwidth, is concentrated at the tone frequency after *dechirp*. Subsequently, the Fast Fourier Transform (FFT) consolidates the signal energy into a peak at f_{sym} in the frequency domain, identifying the symbol. Essentially, the FFT not only identifies the frequency but also gathers the entire chirp energy into a single FFT bin, resulting in a significant *SNR gain after dechirping*. This SNR gain raises the LoRa symbol above the surrounding noise, enabling long-distance and below-the-noise communications.



(a) Standard LoRa (b) *Canas*

Figure 4.3: Illustration and requirements of logical channel interference cancellation problem.

4.2.2 Understanding Logical Channel Interference

Imperfect orthogonality among logical channels. LoRa aims to build orthogonal logical channels by utilizing different spreading factors (SF), which controls the chirp slope k and duration T. When receiving a signal on a specific target logical channel with chirp slope k, the signal originating from another logical channel with chirp slope k' can be represented as follows after the dechirping process:

$$S(f_{sym}, k', t)\ C^{-1}(k, t) = e^{j2\pi[\frac{1}{2}(k'-k)t + f_{sym}]t} \tag{4.4}$$

Since $k' \neq k$, we observe in the above equation that the signal frequency after dechirp continues to vary over time, rather than exhibiting as a constant frequency. Consequently, the energy of the other interfering chirp will spread across the spectrum instead of being concentrated into a single peak after performing FFT. Therefore, LoRa could potentially enable concurrent transmissions over these logical channels (whose SFs differ, *i.e.*, $k' \neq k$), as if they were orthogonal.

In practice, however, the energy of the other interfering chirp (SF=k') spreads across rather than disappears in the demodulation window. As a result, the spread energy inevitably brings up the noise floor. Thus, the increased noise floor caused by a nearby interfering node could sometimes completely overwhelm the ultra-weak

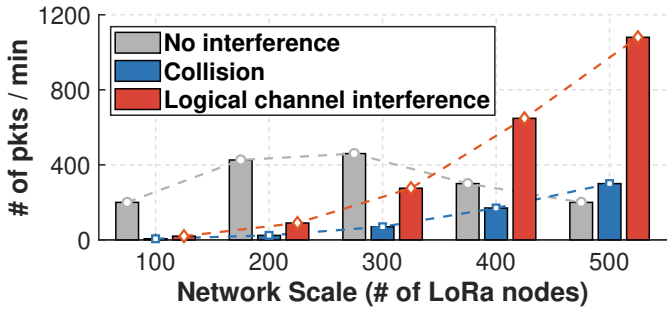


Figure 4.4: Packets that experience logical channel interference dominate the traffic in large-scale LoRa deployments.

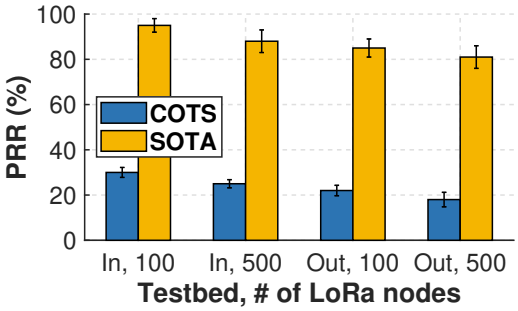


Figure 4.5: Packets under collisions can be recovered by SOTA solution.

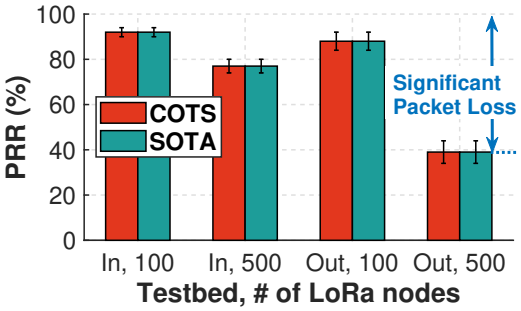


Figure 4.6: Packets under logical channel interference suffer from significant packet loss.

signals from a remote target node even when the target signals (SF=k) can be concentrated into a single peak.

As a matter of fact, practical deployments and measurements reveal that logical channels are seldom completely orthogonal. Figure 4.2(a) illustrates the spectrum of two concurrent logical channels with different SFs (*e.g.*, SF7 and SF10). When the receiver locks on SF10, the target logical channel is clearly distinguishable from the Rx signal, benefiting from the processing gain after dechirping (as shown in Figure 4.2(b)). However, the signals from other logical channels do not vanish but remain present within the demodulation window, dramatically increasing the noise floor. As the SNR difference increases, the orthogonality between logical channels weakens and can even be broken, as shown in Figure 4.2(c). Such SNR differences (*e.g.*, >10 dB) are frequently encountered due to various factors (*e.g.*, near-far effect,

adaptive transmission power, diverse antenna gains) in LoRa networks that cover vast physical areas at the kilometer scale (29). Thus, the logical channel interference could severely affect the packet reception of ultra-weak nodes.

Limitations of current logical channel reception. The root cause for logical channel interference is that standard LoRa solely relies on the demodulation SNR gain to separate different logical channels. As shown in Figure 4.3(a), LoRa demodulation can provide 3∼17 dB SNR gain (82) for specific logical channel. However, a large-scale LoRa network covers massive nodes with varying distances from meters to kilometers and SNRs from $-15\,\mathrm{dB}$ to $20\,\mathrm{dB}$ (42; 29). Given that different SFs typically serve different SNR conditions, the power gap between logical channels can be as large as $35\,\mathrm{dB}$, which easily exceeds the demodulation gain and destroys the orthogonality.

Requirement on orthogonalizing logical channels. In this chapter, we define orthogonality breaks as situations where a weak logical channel cannot be demodulated due to interference from stronger logical channels. The general strategy of orthogonalizing the logical channels is to cancel the interference of strong logical channels on the weak ones. As shown in Figure 4.3(a), a gateway's Rx signal consists of concurrent logical channels from multiple nodes, where the power of strong logical channels can be orders of magnitude higher than the weak ones (29). To achieve reliable concurrent transmission across logical channels, it is crucial to minimize and cancel the interference of strong logical channels, *i.e.*, *logical channel interference cancellation*.

Note that LoRa demodulation provides each logical channel with a unique SNR gain, allowing it to stand out from the noise floor and enable long-range below-the-noise communication. However, the presence of strong logical channels raises the noise floor of weaker ones, thus compromising the orthogonality between the concurrent logical channels. Therefore, to effectively cancel the interference caused

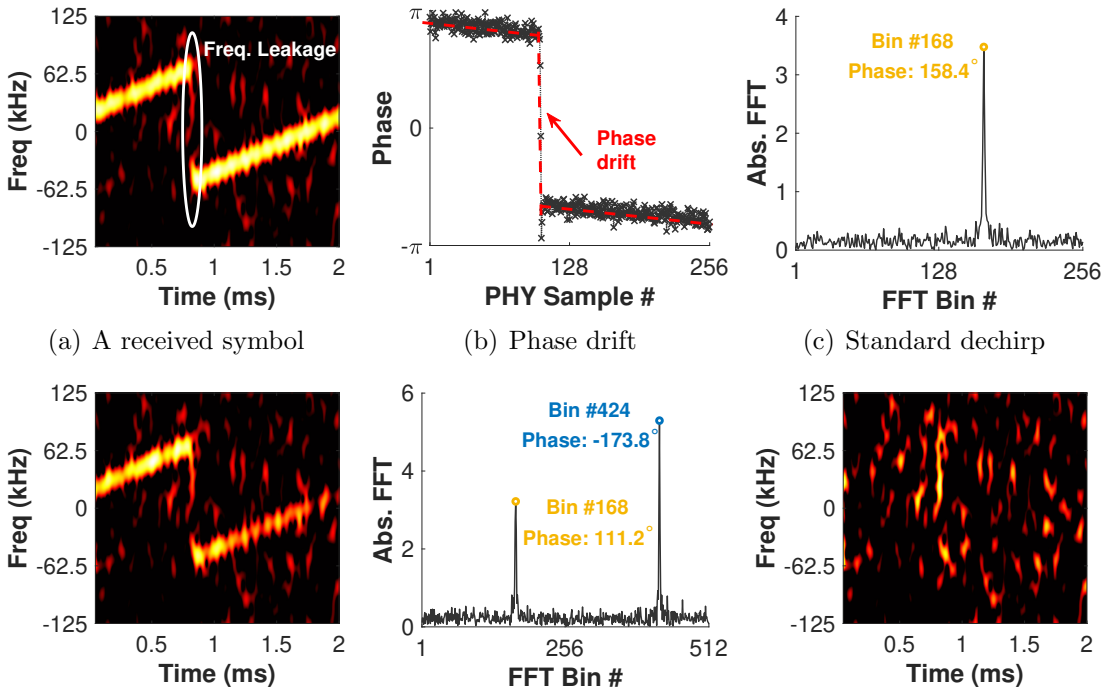
by strong logical channels, it is imperative to suppress them below the noise floor. Next, we can exploit the SNR gain of LoRa demodulation to efficiently cancel the residual interference (see Figure 4.3(b)).

Orthogonality breaks as networks scale. We conduct measurement studies on the indoor and outdoor testbeds (see Figure 4.11) to quantitatively investigate the impact of increased networks on the orthogonality requirement. Our setup utilizes both COTS and USRP gateways to capture measurements and record data traces. The testbed comprises 50 COTS LoRa nodes distributed across various ranges and SNRs. We increase the transmit duty cycle to practically emulate a larger number of nodes. The nodes adopt an ALOHA-based MAC protocol.

Figure 4.4 illustrates the variation among three types of packets (*i.e.*, no interference, collision, and logical channel interference). As the network scales, an increasing number of packets experience collisions or logical channel interference. Notably, packets experiencing logical channel interference (*i.e.*, transmitted across different logical channels) dominate, significantly outweighing those experiencing collisions (*i.e.*, transmitted within a single logical channel). While the SOTA solution of collision recovery (*e.g.*, CIC (2)) can recover some packets affected by collisions (see Figure 4.5), the packet reception ratio for packets affected by logical channel interference—which dominate the traffic—still drops significantly to 39% in the outdoor scenario (see Figure 4.6). Such logical channel interference cannot be resolved by the collision recovery solution. The reason is that collision recovery techniques (*e.g.*, CIC (2)) are designed for resolving explicit collision (see Figure 4.8(a)) and cannot handle the increased noise floor from logical channel interference (see Figure 4.8(c)).

4.2.3 Objective

Logical channels have demonstrated significant advantages in enabling high concurrency (4; 48), efficient channel access (77; 116), and adaptive data rate (43) in



(a) A received symbol (b) Phase drift (c) Standard dechirp

(d) w/o calibration (e) Up-sample dechirp (f) w calibration

Figure 4.7: Illustration of frequency leakage and impacts on interference cancellation: (a) LoRa symbol experiences frequency leakage at chirp edges; (b) Random phase drift; (c) Standard dechirp: two chirp segments are aliased; (d) Cancellation fails due to phase misalignment between the local chirp and Rx signal; (e) Up-sample dechirp: two chirp segments are separated; (f) Cancellation succeeds after calibrating frequency leakage in the local chirp signal.

LoRa networks. The LoRa nodes usually switch between logical channels to adapt to diverse link quality and data rate requirements (117; 118; 119; 120). The use of concurrent logical channels is widely applied in LoRa and brings great opportunities to connect millions of end nodes with high spectrum efficiency (29; 4). However, the limitation of imperfect orthogonality practically hinders the performance of connecting massive LoRa nodes in large-scale deployments, necessitating novel solutions. Conventional LoRa gateways solely rely on LoRa demodulation to separate logical channels. Unfortunately, this approach is inadequate in large-scale deployments with high concurrency and significant SNR disparity among nodes. To unleash the poten-

System Name	Concurrent Transmission	Target Problem
FTrack (31)	Same ¡BW, SF¿ Same Chirp Slope (non-orthogonal)	Single-channel Collision
NScale (32)		
CIC (2)		
Mc-LoRa (3)	Different ¡BW, SF¿ Same Chirp Slope (non-orthogonal)	Cross-channel Collision
XGate (4)	Different ¡BW, SF¿ Different Chirp Slope ("orthogonal")	Scale to More Logical Channels
Canas	Different ¡BW, SF¿ Different Chirp Slope ("orthogonal")	Inter-Logical Channel Interference

Table 4.1: Comparisons between *Canas* and SOTA. ***Canas*** addresses related but different research problem. Note that XGate (4) focuses on packet detection and decoding over massive logical channels, and does not address the problem of inter-logical channel interference.

tial of massive logical channels, we propose a new LoRa gateway design - *Canas*, to orthogonalize logical channels by canceling inter-logical channel interference. *Canas* only requires updates on gateways, without any modifications to the deployed COTS LoRa nodes.

4.3 Limitations of Existing Solutions

Existing solutions for LoRa concurrent transmissions mainly focus on addressing collisions within a single logical channel or cross-channel collisions. To the best of our knowledge, this chapter is the first to systematically investigate the inter-logical channel interference as summarized in Table 4.1.

Packet Reception over Multiple Non-overlapping Channels. To facilitate concurrent packet reception, commercial off-the-shelf (COTS) gateway chips (*e.g.*, SX1301 (1)) divide the 1.6 MHz bandwidth spectrum into eight independent, non-overlapping physical channels, along with an additional configurable high-rate

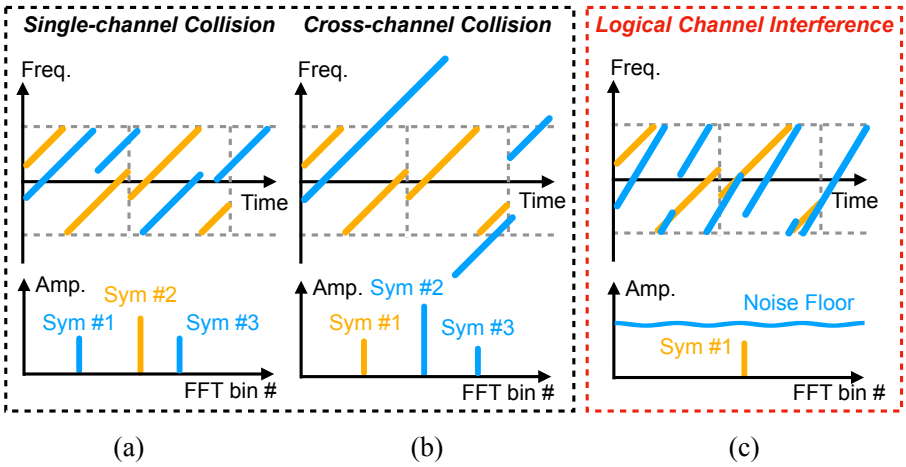


Figure 4.8: Existing works deal with explicit collisions under single-channel (2) and cross-channel (3) scenarios while *Canas* solves the increased noise floor from inter-logical channel interference.

channel. After frequency shifting and filtering through the hardware Rx chains, gateways employ SF-specific LoRa detectors and decoders to receive concurrent logical channels.

Resolving Single-channel Collisions. Most recent works primarily focus on resolving collisions within one single logical channel (see Figure 4.8(a)) to support concurrent transmissions (30; 121; 31; 71; 2; 72; 73). The primary approach involves separating collided symbols within each decoding window and then decoding them individually using standard LoRa techniques. State-of-the-art separation methods exploit hardware imperfections (30), time misalignment (31), non-stationary features (32), reception diversity (33), and sub-symbol features (2). However, when faced with inter-logical channel interference from overlapping logical channels, LoRa packet reception encounters the unique challenge of an increased noise floor, rendering these symbol separation techniques ineffective (as detailed in §4.2.2).

Decoding Cross-channel Collisions. A few studies (3; 110) have noticed that non-orthogonal logical channels with different bandwidths can also cause collisions in LoRa networks (see Figure 4.8(b)). To enable concurrent transmission under

cross-channel collisions, Mc-LoRa (3) employs signal viper techniques to diversify time-frequency domain features and separate collisions. SD-LoRa (110) further enhances performance by utilizing self-dechirp operations. Similar to single-channel collision recovery methods, these techniques are ineffective under inter-logical channel interference due to the increased noise floor, caused by imperfect orthogonality among logical channels.

Scaling to More Logical Channels. Recent work (4) highlights the disparity between the numerous available logical channels in the LoRa spectrum and the limited number covered by existing gateways. To enable flexible scaling, XGate (4) introduces an auto-configured LoRa gateway capable of covering all available logical channels in the Rx spectrum. Although XGate achieves hundreds of concurrent transmissions using the same spectrum as COTS gateways, it still suffers from inter-logical channel interference, leading to significant packet loss when all "orthogonal" logical channels are used concurrently.

Successive Interference Cancellation. Previous studies (122; 123; 124; 125) have applied successive interference cancellation (SIC) in Code Division Multiple Access (CDMA) systems to support concurrent transmissions. However, under inter-logical channel interference, traditional SIC methods fail because LoRa experiences significant signal fluctuation and instability due to longer packet airtime as well as frequency drift due to low-cost hardware and clocks. Additionally, SIC requires high signal quality, whereas concurrent LoRa packets often have diverse SNR conditions and poor signal quality. We draw strength from these pioneering works and address the unique technical challenges involved in accurate reconstruction and cancellation of ultra-weak LoRa signals affected by various practical factors (*e.g.*, hardware imperfection of lightweight LoRa devices, frequency offset and jitters, large disparity in signal strength and quality).

Difference with Existing Works. Existing works mainly focused on the colli-

sion recovery for concurrent transmission, addressing both single-channel (30; 31; 2) and cross-channel (3; 110) scenarios where explicit collisions occur during demodulation. In contrast, our work focuses on the concurrent transmission over multiple logical channels (see Figure 4.8(c)). These channels introduce strong mutual interference instead of explicit collision in large-scale deployments. Our extensive measurements (§4.2) have revealed that this issue poses a significant bottleneck for reliable packet delivery from distant nodes in large-scale LoRa deployments, a problem that has been largely overlooked until now. Unlike the widely studied collision recovery problem (126; 30; 33; 3), this study is, to the best of our knowledge, the first to identify and tackle the problem of inter-logical channel interference.

4.4 System Design

In this section, we introduce *Canas*, a novel gateway design that eliminates logical channel interference to support reliable concurrent transmission across logical channels. We first present the key technical components for signal reconstruction and then give the system architecture.

4.4.1 Reconstruction of Interfering Logical Channel

Without loss of generality, suppose N nodes communicate concurrently on overlapping logical channels. The Rx signal of concurrent logical channels can be denoted as $y(t) = \sum_{i=1}^{N} S_i(t - t_i) + n(t)$, where $S_i(t)$ denotes the raw signal of i^{th} logical channel, t_i indicates the time shift of raw signal, and $n(t)$ is the Rx noises.

Assuming that $S_1(t)$ has a high signal strength, it interferes with other logical channels. Accurately subtracting the interfering logical channel from the Rx signal requires a precise replica of $S_1(t)$. However, extracting $S_1(t)$ directly from the received signal $y(t)$ is ineffective due to the presence of overlapping concurrent logical channels. These overlapping signals introduce distortions to the amplitude and phase

characteristics of $S_1(t)$, making it challenging to obtain an accurate replica of the raw signal.

To address this practical issue, the basic idea behind *Canas* signal reconstruction is as follows: we leverage the fact that the interfering logical channel $S_1(t)$ is typically strong and less interfered with by other weaker logical channels. This allows us to demodulate $S_1(t)$ and extract its payload data explicitly. Instead of directly estimating $S_1(t)$ from the Rx signal $y(t)$, *Canas* generates a local signal replica of the interfering logical channel $S_1'(t)$ using the received payload data and reconstructs $\hat{S}_1(t)$ based on this signal replica.

Nevertheless, reconstructing a LoRa packet is considerably more complex than traditional wireless signals (35; 122). The reason is that LoRa utilizes CSS modulation that has a much longer symbol duration, making it more vulnerable to various sophisticated signal offsets. Since the LoRa packets are transmitted by low-cost hardware, the received signals experience significant hardware imperfections, causing them to deviate substantially from the ideal signals. As a result, the locally generated signal $S_1'(t)$ is not yet an effective signal replica of the interfering logical channel $S_1(t)$. *Canas* further calibrates the various frequency and phase offsets introduced by low-cost RF components to achieve precise signal reconstruction.

Phase Drift. Since LoRa modulates data by cyclically shifting the initial frequency of the chirp signal, a typical LoRa symbol consists of two chirp segments, as illustrated in Figure 4.7(a). LoRa radios are susceptible to frequency leakage (33; 20) when the frequency transitions from one chirp to another (*e.g.*, at the chirp edges). This frequency leakage often introduces phase drifts to the transmitted symbols, resulting in inner-symbol phase variance (see Figure 4.7(b)).

A naive reconstruction method is to directly build on the locally generated signal. We use the demodulation result in Figure 4.7(c) to reconstruct a local chirp and subtract it from the received signal (*i.e.*, the symbol in Figure 4.7(a)). The outcome

is displayed in Figure 4.7(d)). We observe that the residual signal still retains high energy, indicating that the cancellation has failed. This failure occurs because the local signal and the received signal, despite carrying the same symbol, are not phase-aligned. To effectively reconstruct the received signal and further subtract it from the Rx signal, it is essential to ensure their phase consistency.

Therefore, the key problem is to precisely capture and emulate the phase drift when reconstructing the local signal $S_1'(t)$. This task is non-trivial because the phase drifts introduced by frequency leakage are typically random and unpredictable. Moreover, the phase drift information is distorted during LoRa demodulation since the two chirp segments overlap in the same FFT bin due to spectrum aliasing (see Figure 4.7(c)).

To address this practical issue, we exploit the observation that the phase drift at the chirp edges does not affect the inner phase variation within the two chirp segments. it is possible to estimate the phase drift by measuring the phase difference between these segments, provided they can be accurately separated. Our key insight is that the distortion of phase drift information is caused by spectrum aliasing (127), resulting from the limited sampling rate. By up-sampling the LoRa symbol to a higher sampling rate (*e.g.*, 2× bandwidth), we can separate the two chirp segments after dechirping and FFT, as depicted in Figure 4.7(e). From the two peaks, we can explicitly extract the phase difference between the two chirp segments, which exactly indicates the phase drift at the edge. Subsequently, we calibrate the phase drift during the local chirp reconstruction process and subtract it from the received signal. As shown in Figure 4.7(f), the received symbol is successfully suppressed to a noise level, demonstrating effective signal reconstruction.

Carrier Frequency Offset. The oscillator frequency of the gateway and the LoRa node cannot be perfectly matched. As a result, there is usually a carrier frequency offset (CFO) in the received signal, causing linear phase rotations repre-

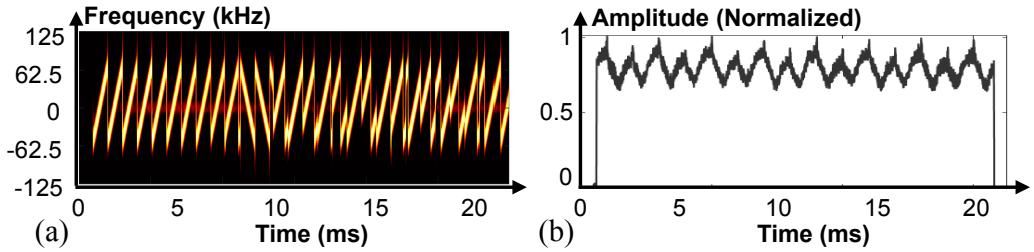


Figure 4.9: Dynamic channel fluctuation: (a) spectra and (b) amplitude of a received LoRa packet.

sented as $e^{j2\pi\Delta ft}S_1'(t)$. The CFO has been measured to be approximately 10 kHz (128). Given the narrow bandwidth of LoRa (e.g., 125 kHz), if left uncorrected, the CFO can significantly impact signal reconstruction.

Only using the preamble to estimate CFO requires a fine-grained sliding window at the sample level, incurring significant computational overhead. To achieve lightweight CFO estimation during the signal reconstruction process, we exploit the packet structure of LoRa to extract the CFO information. The key idea is that LoRa packets contain not only base up-chirps as preamble but also base down-chirps as Start Frame Delimiter (SFD) (85; 33; 129), where different parts of the same packet experience the same CFO. Moreover, the CFO produces opposite frequency effects on the preamble and SFD parts of the packet. By utilizing the up-chirp preamble ($C_{pr}(t)$) and the down-chirp SFD ($C_{sfd}^{-1}(t)$), we can cancel out the CFO effects as follows:

$$he^{j2\pi\Delta ft}C_{pr}(t) \cdot he^{j2\pi\Delta ft}C_{sfd}^{-1}(t) = h^2e^{j2\pi(2\Delta f)t} \tag{4.5}$$

Here, h represents the impact of the wireless channel. By performing FFT on the resulting signal in Equation 4.5, the peak location indicates the frequency of $2\Delta f$. This information allows us to estimate and calibrate the CFO during signal reconstruction.

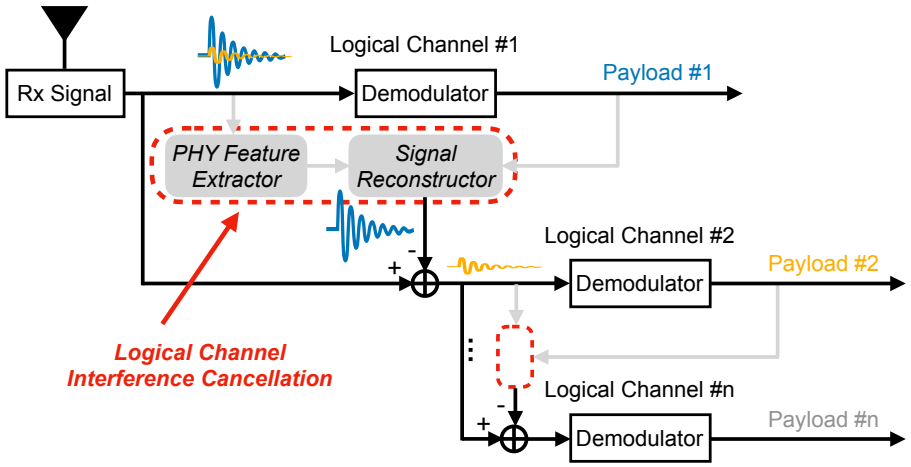


Figure 4.10: Workflow of *Canas*: *Canas* iteratively demodulates, reconstructs, and removes individual logical channels from the Rx signal.

4.4.2 Enhancing Reconstruction Granularity

Although calibrating the frequency leakage and CFO makes the local signal replica closely resemble the interfering logical channel in the Rx signal, it remains unsatisfactory to directly subtract this replica from the Rx signal. The reason is that the received LoRa packets are subject to dynamic channel fluctuation in addition to various hardware imperfections. Figure 4.9 illustrates the spectrum and signal amplitude of a received LoRa packet. The long air-time of a LoRa packet (*e.g.*, hundreds of ms) and the inherent instability of the Tx hardware (e.g., Low-cost LoRa node) result in noticeable channel fluctuations in the received raw signal, as depicted in Figure 4.9(b).

A naive solution is to apply the same amplitude fluctuation on the reconstructed signal replica. However, it is challenging to obtain precise amplitude fluctuation of individual logical channels from the Rx signal as it contains multiple concurrent logical channels. Additionally, the dynamic channel fluctuations introduce not only amplitude variations but also phase variations. We address this problem to reconstruct the signals of individual logical channels with finer granularity.

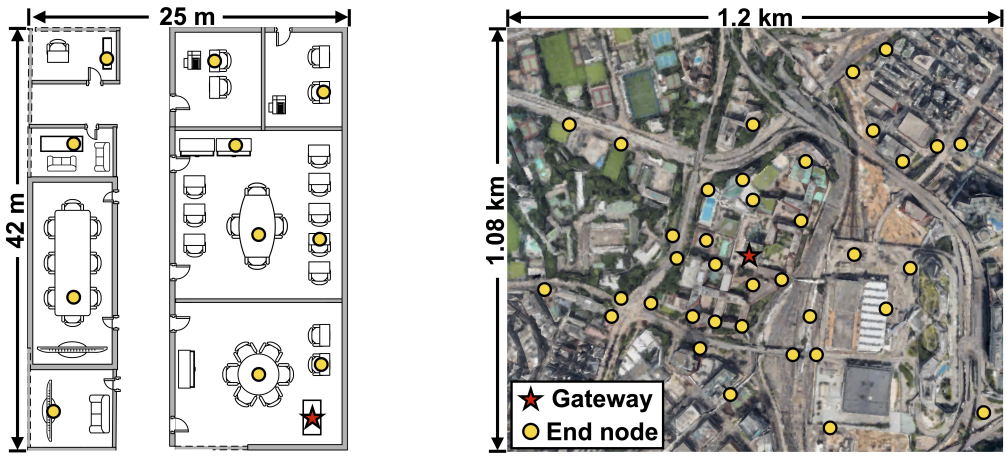
To address the time-varying nature of practical LoRa channels, *Canas* introduces a sophisticated channel estimation technique to facilitate precise signal reconstruction. capitalizes on the observation that each chirp present in a LoRa packet can serve as a basis for channel estimation. By adopting a per-chirp channel estimation strategy, *Canas* enhances the granularity of signal restoration. The received signal $y(t)$ can be represented as the combination of one logical channel to be reconstructed and other logical channels:

$$y(t) = S_1(t) + \sum_{i=2}^{N} S_i(t) = h_1^j S_1'^j(t) + \sum_{i=2}^{N} h_i^j S_i(t) \tag{4.6}$$

Initially, we cross-correlate the received signal $y(t)$ with a locally generated raw signal $S_1'(t)$ to pinpoint the signal start of the interfering logical channel. Let $y^j(t)$ denote the j^{th} chirp of $y(t)$, we next employ conjugate multiplication and FFT to derive the chirp-level channel response h_1:

$$\mathcal{F}[y^j(t) \cdot [S_1'^j(t)]^*] \approx \mathcal{F}[h_1^j S_1'^j(t) \cdot [S_1'^j(t)]^*] = h_1^j \tag{4.7}$$

where $(\cdot)^*$ indicates the complex conjugate operation. It is worth noting that conjugate multiplication (*i.e.*, Equation 4.7) serves a dual purpose. Firstly, it eliminates the common chirp signal in $S_1^j(t)$ and $S_1'^j(t)$. Secondly, the signals of other concurrent logical channels (*i.e.*, $\sum_{i=2}^{N} h_i^j S_i(t)$) are spread across the spectrum and diminish towards the noise floor. Consequently, the conjugate multiplication results in a tone frequency corresponding to $S_1^j(t)$, representing the signal deviation from $S_1'^j(t)$ to $S_1^j(t)$. By employing FFT to aggregate the signal power at this tone frequency into a distinct peak, the chirp-level channel estimation $\hat{h_1^j}$ can be derived. We then iterate through all chirps in $S_1(t)$ and $S_1'(t)$, the collected fine-grained channel data $\hat{h_1} = \{\hat{h_1^j}\}$ can be utilized for accurate reconstruction of $\hat{S}_1(t)$. *Canas* not only tracks the dynamic nature of practical LoRa packet channels but also mitigates the influence of other concurrent logical channels during channel estimation. Subsequently,



(a) Indoor testbed (b) Outdoor testbed

Figure 4.11: Testbed setting of *Canas*.

we align $\hat{S}_1(t)$ with $S_1(t)$ and then subtract it from $y(t)$ to eliminate the interfering logical channel.

It is worth noting that the channel response estimation is robust even with multiple interfering signals of similar strength. Note that interfering logical channels use different chirp slopes, and *Canas* uses conjugate multiplication to extract the channel response $\hat{h_1^j}$. The conjugate multiplication (*i.e.*, Equation 4.7) spreads other interfering signals across the spectrum while $S_1(t)$ accumulates to a peak. This process provides a significant SNR gain, ensuring a sufficiently accurate $\hat{h_1^j}$ estimation. The only exception occurs when an interfering signal $S_2(t)$ is significantly stronger than $S_1(t)$ (*e.g.*, >5 dB). In such cases, *Canas* would initiate cancellation starting with $S_2(t)$.

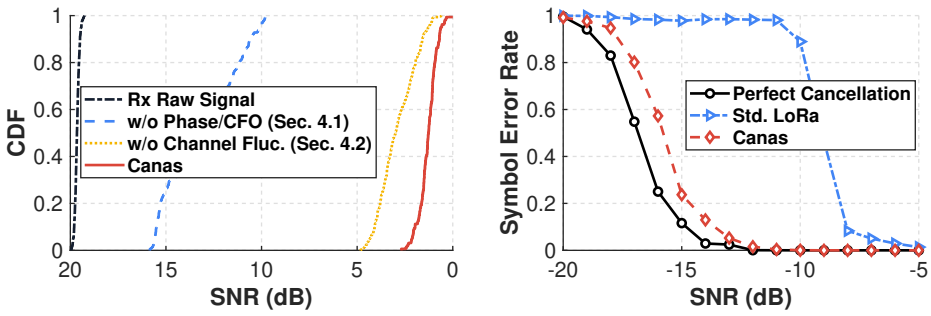
4.4.3 *Canas*: Put All Together

Figure 4.10 shows the general workflow of concurrent logical channel reception in *Canas*.

Logical channel interference cancellation. *Canas* generally applies signal re-

construction to cancel the logical channel interference. Initially, *Canas* demodulates the strongest logical channel, which experiences minimal interference from weaker logical channels. We use the received payload data to locally generate an ideal raw signal. Furthermore, *Canas* extracts diverse signal offsets present within the received LoRa packet, serving as physical layer fingerprints. These offsets are added to the ideal local raw signal to facilitate more precise signal reconstruction. Additionally, *Canas* utilizes the local raw signal to track channel variations in practical received packets, enabling fine-grained signal reconstruction. Finally, *Canas* by subtracting the reconstructed logical channel from the Rx signal, *Canas* effectively suppresses its signal strength to the level of Rx noises. Consequently, the remaining logical channels are free from interference caused by the strong logical channel.

Iteration to receive concurrent logical channels. *Canas* adopts an iterative approach to receive more concurrent logical channels. Since concurrent logical channels may exhibit different signal strengths, they are affected by other logical channels to different degrees. To ensure successful logical channel interference cancellation, *Canas* requires prior demodulation of payload data. We begin by using cross-correlation to detect the preamble of the LoRa packet across all logical channels. This process allows us to determine the packet arrival timing and measure the signal's SNR by analyzing the peak level. Next, we sort the detected packets according to their SNRs and proceed to demodulate the strongest signal. After demodulating the strongest signal, we remove it and move on to the next logical channel. We continue this iterative process to receive concurrent logical channels from the strongest channel to the weakest. The weak logical channels will be demodulated in the later rounds if they are interfered with by other strong packets, even if they arrive earlier. However, a practical challenge arises when the signal strengths of weak LoRa packets fall below the noise floor and become overshadowed by stronger packets, making direct measurement from the Rx signal difficult. To address this, measurements for



(a) Cancellation performance (b) Weak packet reception

Figure 4.12: Performance of logical channel interference cancellation: (a) Signal strength of strong logical channel. (b) Packet reception of weak logical channel.

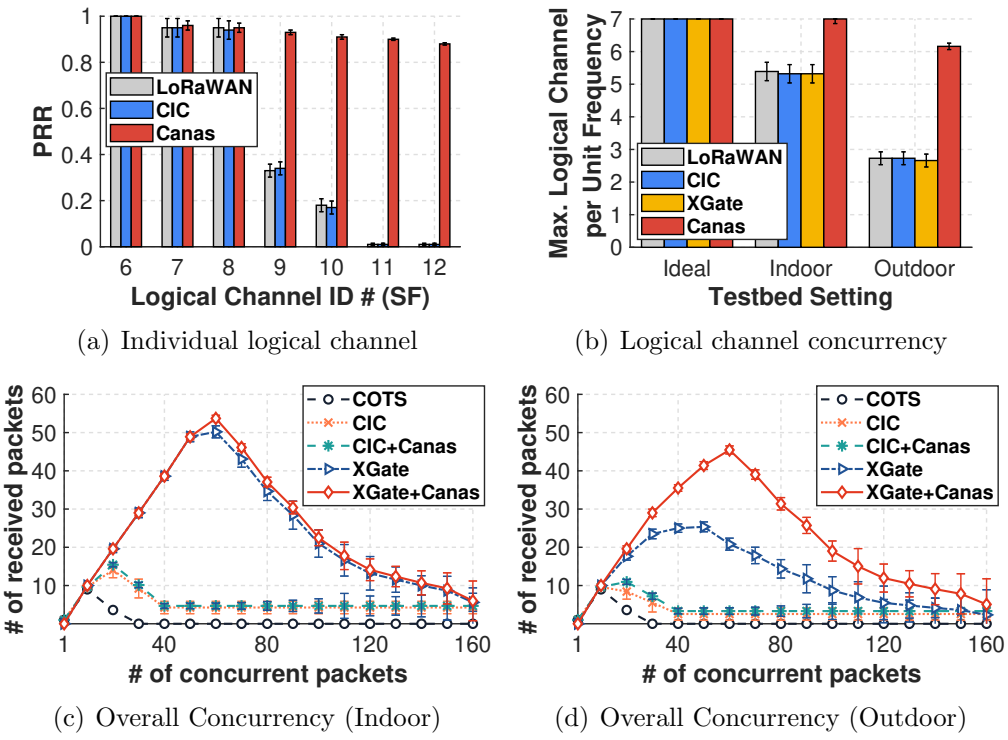
weak packets can be obtained in subsequent cancellation rounds after removing the signals of strong logical channels.

Integration with collision recovery. *Canas* can seamlessly integrate with SOTA collision recovery techniques to deal with hybrid interference (*i.e.*, coexistence of collision and logical channel interference). The integrated approach still follows the iterations from the strongest to the weakest signals. Within each iteration focusing on one logical channel, if collisions are detected, we can replace the standard demodulator (see Figure 4.10) with SOTA decoder (*e.g.*, CIC (2)) to handle collisions. Then, Canas reconstructs and cancels each colliding packet in the logical channel.

4.5 Evaluation

4.5.1 Methodology

Implementation. We implement *Canas* based on the software-defined radio platform (USRP N210) and `gr-lora` project (86). The USRP is employed to receive signals from LoRa nodes, and the received samples are forwarded to a workstation running *Canas* for signal processing. We use COTS LoRa nodes with Semtech SX1276 radio (82) as transmitters and employ Arduino Uno boards for node config-



(a) Individual logical channel

(b) Logical channel concurrency

(c) Overall Concurrency (Indoor)

(d) Overall Concurrency (Outdoor)

Figure 4.13: Comparision with state-of-the-art: (a) packet reception ratio of each logical channel under concurrent transmission; (b) the maximum number of concurrent logical channels supported on a unit frequency; (c) indoor and (d) outdoor performance of concurrent packet reception in a 1.6 MHz spectrum.

uration. We deploy a testbed including 50 LoRa nodes and 2 gateways. We conduct experiments on our campus, spanning an area of 1.08 km $\times$ 1.2 km that represents a typical urban environment. The gateways are installed on the rooftop of a 20-storey building, and all nodes operate in the 915 MHz ISM band.

Experiment Setup. We collect data traces from >200 links for over two months in our testbed. The traces encompass diverse channel conditions in typical urban settings, including indoor and outdoor, low and high SNRs (see Figure 4.11). All evaluations with $\leqslant$50 nodes are conducted via real-world experiments in the testbed. Additionally, we also perform large-scale trace-driven evaluations using collected data traces. Our evaluation primarily focuses on addressing the following questions: (1) How does *Canas* perform in logical channel interference cancellation? (2) How many

concurrent logical channels can be supported by *Canas*? and (3) How does *Canas* perform in supporting practical IoT applications?

Baselines. We compare *Canas* with three baselines on concurrent transmission for LoRa. (1) *LoRaWAN* directly demodulates individual logical channels from the Rx signal; (2) *CIC* adopts collision recovery technique to receive concurrent packets within a single logical channel; (3) *XGate* implements auto-configuration gateway to cover all available logical channels in the Rx spectrum.

4.5.2 Logical Channel Interference Cancellation Performance

Cancellation effectiveness. We first evaluate *Canas* in logical channel interference cancellation. We set up two LoRa nodes that transmit concurrently at low and high SNRs (*i.e.*, −15 dB and 20 dB) using SF10 and SF7, respectively. Figure 4.12(a) illustrates the SNR of SF7 packets over 15 seconds. We observe that the strong logical channel (*i.e.*, SF7) is originally received with high SNR (*i.e.*, around 20 dB) in the Rx signal, which significantly interferes with other logical channels. *Canas* effectively reduces its power to the noise floor, resulting in a median SNR of 1.3 dB after cancellation. We also conduct ablation studies to evaluate the novel techniques in *Canas*. The results demonstrate the crucial role of calibrating the phase drift, frequency offset, and channel fluctuation in achieving precise signal reconstruction and enabling effective logical channel interference cancellation. The median SNR are 12.9 dB and 3.1 dB without canceling phase drift, frequency offset, and calibrating channel fluctuation, respectively (see Figure 4.12(a)).

Ultra-weak packet reception under interference. We next investigate the Rx sensitivity of *Canas* after logical channel interference cancellation. We fix the SNR of the interferer and measure the symbol error rate of the transmitter at different SNRs. Specifically, we measure the received packets of the SF10 node under different SNR conditions and analyze >100 packets for each SNR condition. The

experiments are under interference from a concurrently transmitting SF7 node at the SNR of 10 dB. For comparison, we also disable the SF7 node (*i.e.*, the interferer) to represent perfect cancellation. As shown in Figure 4.12(b), the perfect cancellation performs best as it has no interference from other logical channels. LoRaWAN (27), which directly demodulates the weak logical channel, suffers from significant SNR loss due to interference from strong logical channels. In contrast, *Canas* can receive weak packets at ultra-low SNRs (*e.g.*, −14.8 dB for SF10) by reconstructing and subtracting the strong logical channel. Compared to the perfect cancellation, the interference from the strong logical channel only leads to 0.9 dB SNR loss on Rx sensitivity of *Canas* while achieving a symbol error rate of <20 %.

4.5.3 Comparison with state-of-the-art

Reception of individual logical channel. This experiment delves into each logical channel to evaluate the performance of *Canas* in supporting concurrent logical channels. We configure seven LoRa nodes to transmit concurrently using different logical channels (SF 6∼12) on the same central frequency. These nodes are evenly distributed with SNRs ranging from −15 dB to 10 dB in the outdoor testbed area. Nodes select logical channels based on their SNRs, where large (low) SFs are used for low (high) SNRs. We compare *Canas* against LoRaWAN (27) and CIC (2). LoRaWAN employs a standard LoRa decoder for receiving each logical channel, while CIC utilizes a collision recovery decoder. *Canas* receives concurrent logical channels with iterative interference cancellation.

Figure 4.13(a) presents the packet reception ratio (PRR) of each logical channel. We observe that LoRaWAN can only reliably receive the low-SF logical channels, and the PRR drops significantly for the high-SF ones. The reason is that the high-SF logical channels are utilized for low SNR links and are susceptible to logical channel interference. The signals from strong logical channels (SF 6∼8) can raise the Rx noise

floor for the weaker ones, leading to disrupted demodulation. CIC cannot improve the reception of concurrent logical channels as it focuses on resolving collisions within a single logical channel. In contrast, *Canas* demonstrates a substantial enhancement in packet reception by effectively canceling the mutual interference between logical channels. Even the weakest logical channel (SF12) achieves an average PRR of >80%.

Logical channel concurrency. This experiment compares *Canas* with the current leading strategies for concurrent transmissions in LoRa. We consider two state-of-the-art (SOTA) strategies, *i.e.*, CIC (2) and XGate (4). To ensure a fair comparison, we focus on the maximum number of concurrent logical channels on the same physical channel (*i.e.*, overlap on the same central frequency). We configure seven LoRa nodes to transmit concurrently using different logical channels (SF 6∼12) on the same central frequency. Both CIC and XGate are implemented on USRP devices, as they require access to the PHY raw signal. We test three scenarios in our testbed: (1) Ideal, where nodes transmit independently; (2) Indoor; and (3) Outdoor, representing different environmental conditions (see Figure 5.11).

Figure 4.13(b) displays the maximum number of concurrent logical channels per unit frequency received by four strategies. When transmitting independently, all logical channels can be received, as expected. However, when transmitting concurrently, LoRaWAN experiences packet loss due to logical channel interference, resulting in an average concurrency of 5.3 and 2.7 for indoor and outdoor deployments, respectively. The outdoor performance is particularly impacted due to the more pronounced near-far effect and larger SNR variance. CIC improves concurrency by resolving collisions within a single logical channel, yet it fails to support concurrent logical channels. XGate enhances concurrency by automatically configuring and receiving all available logical channels, but it does not address the mutual interference between overlapping logical channels. In comparison, *Canas* outperforms the other strategies, achieving an average concurrency of 7 and 6.1 for indoor and outdoor deployments, respec-

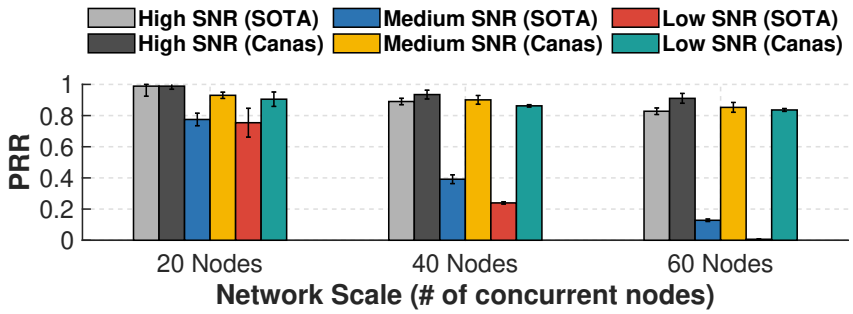


Figure 4.14: Near-far effect at different network scales.

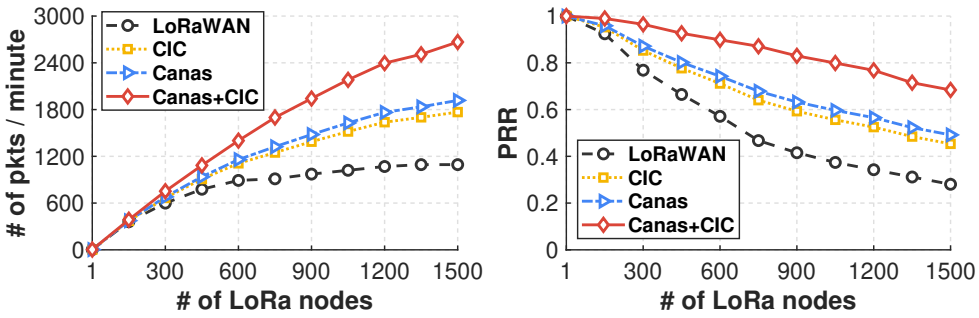
tively (*i.e.*, 1.3× and 2.3× higher logical channel concurrency than SOTA for indoor and outdoor testbed). This superior performance is attributed to *Canas*'s ability to eliminate logical channel interference through iterative signal reconstruction and subtraction.

Integration with SOTAs. This experiment evaluates the performance of *Canas* in enhancing concurrent transmission alongside existing SOTAs. We compare *Canas* with CIC (2) and XGate (4) within the same 1.6 MHz spectrum and 9 Rx chains as the COTS gateway (1). We control a varying number of LoRa nodes to transmit concurrently on different LoRa channels, each with a 125 kHz bandwidth. By varying the spreading factor (SF) from 7 to 12, we support 54 LoRa logical channels. We utilize a beacon to coordinate the simultaneous transmission of 50 nodes, aiming to assess system performance under extreme concurrency. For experiments involving more than 50 nodes, we aggregate multiple signal traces with randomly assigned time offsets to simulate real-world network traffic. These synthesized traces are then replayed for evaluation. Importantly, we also integrate SOTAs with *Canas* to evaluate the additional benefits that *Canas*'s logical channel interference cancellation technique can bring to existing LoRa systems. Specifically, we implement *CIC+Canas* by replacing the standard LoRa demodulator in *Canas* with CIC (see Figure 4.10). We implement *XGate+Canas* by replacing the demodulators and decoders in XGate with Canas. The evaluations are conducted on both indoor and outdoor testbeds,

representing the signals with similar and diverse SNRs, respectively. The outdoor testbed experiences more significant near-far effect.

Figure 4.13(c,d) presents the number of received packets for these strategies under different levels of concurrent transmissions. As expected, network concurrency initially increases, reaches a peak, and then declines. CIC achieves 14 and 10 concurrent transmissions for indoor and outdoor testbeds, respectively. *Canas* can work in conjunction with CIC to improve the average concurrency to 15.5 and 11. Their gains are complementary, as they address different types of packet loss. For XGate, it scales concurrent transfers to more logical channels, which also results in more inter-logical channel interference than CIC. We observe that among the 54 concurrent logical channels, XGate in an indoor deployment (*i.e.*, high signal quality) achieves an average of 51 concurrent transmissions with minimal packet loss (see Figure 4.13(c)). However, in practical outdoor deployments, the performance of XGate drops significantly due to more severe inter-logical channel interference. We note that the maximum concurrency of XGate drops to 25 with considerable packet loss. By augmenting with *Canas*, many of the lost ultra-weak packets can be recovered, resulting in an overall concurrency of 45, boosting the XGate by 80% in the outdoor testbed through seamless integration. These results demonstrate that *Canas* complements SOTAs and significantly enhances LoRa systems under near-far effects in outdoor deployments.

Near-far effect at scale. This experiment evaluates the performance gains of *Canas* under the near-far effect. We test three network scales with 20, 40, and 60 nodes, all evenly distributed within the testbed area and connecting to the gateway with SNRs ranging from -10 dB to 10 dB. The gateways are configured within the same 1.6 MHz spectrum and 9 Rx chains as the COTS gateway (1). The SOTA combines XGate (4) and CIC (2). LoRa nodes select logical channels based on their SNRs, with larger (lower) SFs used by nodes with lower (higher) SNRs. To



(a) Throughput (b) Packet reception ratio

Figure 4.15: Performance of supporting massive LoRa nodes in large-scale deployments.

investigate the near-far effects, we control all nodes to transmit concurrently and measure the packet reception performance across different SNR regimes.

Figure 4.14 illustrates the packet reception ratio (PRR) for the three SNR regimes as received by SOTA and *Canas*. Most of the received packets experience imperfect orthogonality and suffer from varying degrees of PRR loss. We observe that the PRRs for high-SNR packets are less affected as the network scales, remaining above 80 % even when 60 nodes transmit concurrently. In contrast, the PRRs for medium- and low-SNR packets drop dramatically, as weaker packets suffer from significantly more logical channel interference than stronger packets, which SOTAs cannot resolve. *Canas* addresses this by iteratively reconstructing and removing strong logical channels, thereby canceling the interference and improving the PRRs for medium- and low-SNR packets to above 80 %. These results demonstrate that *Canas* significantly enhances the reception of ultra-weak packets under practical near-far effects in large-scale deployments.

Massive packet reception performance. This experiment assesses *Canas* for supporting massive LoRa nodes in large-scale deployments using trace-driven emulations. Specifically, we aim to deploy LoRa gateways to connect up to 1,500 IoT sensors. To ensure a fair comparison, we adopt the same channel plan as the COTS

gateway (1), which includes 54 logical channels (BW125 kHz, SF7∼12) distributed over 9 central frequencies. Each IoT sensor transmits a 20-byte message every 30 minutes, with a duty cycle of ⩽1%. To investigate communications for thousands of sensors, we employ a trace-driven approach. Specifically, we capture LoRa packet profiles from over 200 sites in the outdoor testbed, with SNR ranging from −20 dB to 10 dB. We use the collected data traces to synthesize the received packet signals with randomly selected link profiles. LoRa nodes freely select from the available logical channels to transmit messages using an ALOHA-based MAC. For benchmarking, we compare *Canas* with two baselines: (1) LoRaWAN, which utilizes a COTS gateway (1) for reception; and (2) CIC, which implements collision recovery (2) upon LoRaWAN.

Figure 4.15 compares the receive packet throughput and packet reception ratio (PRR) of *Canas* and baselines under different numbers of active LoRa nodes. We see that the packet throughput of LoRaWAN first increases and then becomes saturated. The reason is twofold. Firstly, packets using the same logical channel may arrive simultaneously, leading to collisions. Secondly, packets from strong logical channels can interfere with weaker ones, resulting in significant packet loss for high-SF logical channels. As shown in Figure 4.15(a), CIC improves the packet rate through collision recovery techniques. Meanwhile, *Canas* achieves higher improvement by canceling the logical channel interference. Importantly, the benefit provided by *Canas* is orthogonal to traditional collision recovery methods (2; 31; 67). *Canas* can jointly work with CIC to support >800 LoRa nodes with PRR>85% (see Figure 4.15(b)). Importantly, the concurrency improvement of *Canas+CIC* compared to pure CIC is higher than Figure 4.13(c,d). This is because *Canas+CIC* experiences a performance decrease when handling hybrid interference. In this experiment, which uses a real-world duty cycle, the concurrency level is not as pronounced as in Figure 4.13(c,d), where strict concurrency is evaluated.

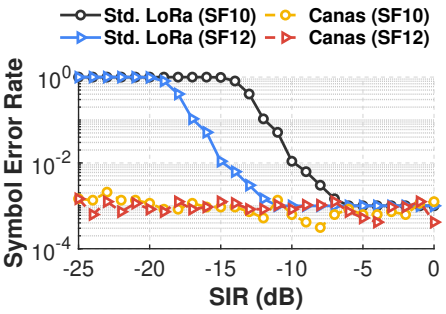


Figure 4.16: Impact of SIR.

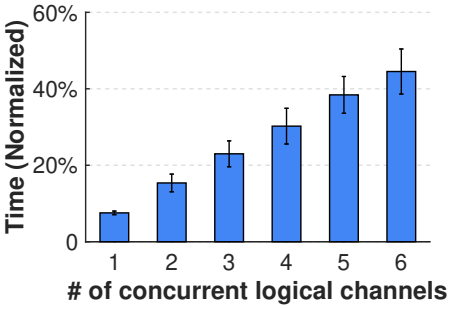


Figure 4.17: Time overhead.

4.5.4 Microbenchmarks

Impact of SIR. To assess the impact of the signal-to-interference ratio (SIR) on packet reception, we conduct experiments that quantitatively measure the interference caused by strong logical channels on weaker ones. We fix the SNR of the transmitter and change the SNR of the interferer, aiming to measure the symbol error rate of the transmitter at different SIRs. We deploy a LoRa node as the transmitter, which transmits packets with a fixed link SNR of -10 dB. Simultaneously, another LoRa node was deployed as the interferer, transmitting packets with varying link SNRs ranging from -10 dB to 15 dB. The transmitter utilizes SF10 and SF12, while the interferer uses SF7.

Figure 4.16 illustrates the decoding errors observed at the transmitter node under different SIR conditions. Generally, the symbol error rate (SER) increases as the interference becomes stronger (*i.e.*, as the SIR decreases). We observe that the SIR thresholds (with SER <10%) for SF10 and SF12 are -17 dB and -12 dB, respectively. High-SF logical channels demonstrate greater resilience to logical channel interference. However, such SIR conditions between logical channels frequently occur in large-scale LoRa deployments characterized by significant SNR variances and concurrent transmissions. Compared to LoRaWAN, *Canas* significantly reduces the SIR during LoRa demodulation, thereby enhancing the reception performance of weaker

logical channels.

Time overhead. In this experiment, we examine the time overhead of logical channel interference cancellation in *Canas*. The algorithm is executed on a laptop, and we measure the runtime for canceling different numbers of logical channels. We normalize the overhead to the ratio of packet duration to assess the real-time performance. The experimental results demonstrate that the overhead increases linearly, as depicted in Figure 4.17. Importantly, even when canceling six logical channels, the overhead remains below 50% of packet duration. This duration is much shorter than the airtime of LoRa packets and can be effortlessly handled by the processors in the gateway.

4.6 Discussion

Tackling hardware and channel uncertainty. LoRa packets experience significant signal offsets, such as CFO, phase drift, and amplitude fluctuation, primarily due to the long packet air-time and low-cost hardware. This hardware-induced uncertainty differs among devices and transmission parameters, making it challenging to predict. Previous offset compensation methods aim to enhance demodulation. In contrast, *Canas* aims to reconstruct the raw signal of individual logical channels with special consideration for replaying these offsets. PCube (33) utilizes synchronized antenna pairs to calibrate the CFO before demodulation, while *Canas* leverages the inherent conjugate chirp in the LoRa packet to estimate CFO and further replay it to reconstruct a local signal. LoRaTrimmer (130) uses probabilistic model to mitigate phase drift but does not precisely capture it, while *Canas* does and further replays on local signal for precise signal reconstruction. GLoRiPHY (131) extracts preamble channel information for payload denoising. This method can decode packets, but the granularity is insufficient for precise reconstruction (see Figure 12(a)). Canas

presents new methods to capture and replay the signal fluctuation for precise signal reconstruction.

Interference signal cancellation. Previous studies have investigated interference signal cancellation techniques for successive interference cancellation (SIC) (122; 123; 124; 125). While both *Canas* and traditional SIC-based methods involve reconstructing and canceling signals from the strongest to the weakest, they differ fundamentally in several key aspects: (1) Research Problem: Canas focuses on addressing inter-logical channel interference in LoRa networks, which arises due to imperfect orthogonality between logical channels. Traditional SIC, on the other hand, deals with multi-user interference signals within the same channel, where differences in signal strength are the primary concern. Canas, however, handles concurrent logical channels where interfering signals vary not only in signal strength but also in modulation parameters and symbol/packet durations. (2) Unique Challenges: Canas introduces new techniques to address these issues. Precise signal reconstruction in LoRa is challenged by factors such as weak signal strength, hardware imperfections, and long symbol/packet durations. Canas develops innovative methods to extract hardware imperfections and track the air-channel with fine granularity.

Benefits and overheads of *Canas*. *Canas* greatly empowers the reception reliability of weak packets (*i.e.*, from distant LoRa nodes) in large-scale deployments. The evaluation of corresponding energy efficiency gains will be carried out in future work. *Canas* can integrate with existing LoRa gateways without requiring any modifications on the deployed COTS LoRa nodes. To cancel the logical channel interference, *Canas* introduces higher computational overheads at the gateway but can be processed in real time.

Interaction with PHY error correction. LoRa's physical layer utilizes a forward error correction mechanism that employs a coding rate to introduce redundancy into the transmitted data, enabling it to correct a certain percentage of symbol errors

(*e.g.*, <20% symbol error for a coding rate of 4/5). This error correction can handle limited instances of logical channel interference, such as when a strong SF7 packet interferes with only a few SF12 symbols due to the significant difference in symbol duration. However, it cannot fully resolve the issue. Firstly, SF12 packets have a long duration and can be disrupted by multiple low-SF packets, leading to accumulated symbol errors that exceed the error correction's capacity. Secondly, neighboring SFs can interfere with each other due to similar packet airtime, resulting in symbol errors that surpass the 20% threshold. To effectively address logical channel interference, the error correction mechanism should be complemented by *Canas*. For instance, *Canas* can reduce the symbol error rate of weak packets from over 30% to below 10%, allowing the error correction algorithm to resolve the residual errors.

Handling hybrid interference. Both *Canas* and SOTA collision recovery methods, such as CIC (2), address interference scenarios where multiple nodes transmit simultaneously on the same channel frequency. SOTA methods can resolve traditional collision issues (same BW, same SF), but fail to handle logical channel interference (same SF, different SF). A further challenge is hybrid interference, where traditional packet collisions occur alongside logical channel interference. Preliminary results suggest that Canas+CIC can partially address this issue. However, the performance of both Canas and CIC diminishes under conditions of extreme hybrid concurrency. The combined performance improvement of Canas+CIC is less than the optimal sum of their individual gains, as illustrated in Figure 4.13(c,d). Our future research will focus on developing unified and comprehensive strategies to tackle hybrid interference signals.

4.7 Conclusion

This chapter introduces *Canas*, a novel design of LoRa gateway aimed at practically orthogonalizing massive LoRa logical channels. Unlike traditional LoRa gateways that directly demodulate each logical channel, *Canas* incorporates a new logical channel interference cancellation technique, which reconstructs the raw signal of individual logical channels from the superimposed Rx signal and iteratively eliminates their interference. This breakthrough addresses the imperfect orthogonality and strong mutual interference between the long-believed "orthogonal" logical channels in large-scale LoRaWAN deployments, thereby unlocking the full potential of utilizing massive logical channels for achieving high concurrency, flexible channel access, adaptive data rates, and more.

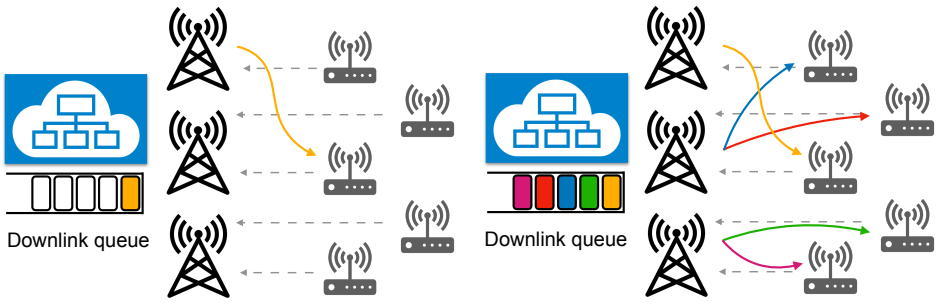
Chapter 5

FDLoRa: Empowering Scalable LoRa Downlink for Multi-Functional IoT Applications

5.1 Introduction

Recent years have witnessed Low-Power Wide-Area Networks (LPWANs) as promising technologies to enable ubiquitous connectivity of massive networked sensors to cyberspace, underpinning a variety of embedded AI and sensing systems and their applications (104; 105; 93; 11; 132; 77; 12; 20; 96; 97; 133; 15; 54; 65; 53; 24; 52; 134; 91; 135; 136; 137; 138; 139; 140; 36; 3; 110; 141; 106; 142; 99; 120; 119; 56; 55). As a leading LPWAN technology, LoRa features extensive coverage and energy-efficient communication, connecting over 300 million LoRa nodes and 5.9 million gateways worldwide (109).

These deployed nodes are supporting various IoT applications, where both uplink and downlink transmissions are indispensable to read and control sensors, enable essential systems and networking functionalities, ensure communication reliability, and optimize overall performance. For example, smart lamppost systems (37) necessitate both uplink and downlink transmissions for status acquisition and device actuation (*e.g.*, turn on/off lights), and firmware update services (38; 39) need both uplink and



(a) Traditional LoRa downlink (b) Concurrent downlink framework

Figure 5.1: Downlink-uplink asymmetry: (a) LoRaWAN receives uplink concurrently but transmits downlink sequentially; (b) *FDLoRa* scales concurrent downlink to massive logical channels of available gateways.

downlink for device authentication and firmware transfer. To support reliable data transfer (40; 41) and data rate adaptation of LoRa nodes (42; 43), gateways need to acknowledge successful uplink transmissions to the nodes via downlink.

In supporting the above applications and systems operations, current LoRa networks face practical challenges due to the *asymmetry between downlink and uplink transmissions*. Although recent research works have made remarkable strides (31; 2) in improving the uplink performance (*e.g.*, concurrent uplink transmissions (31; 2; 33), weak packet reception (51; 64)), the problem of poor downlink performance largely remains under-explored. While commodity LoRa gateways are typically equipped with multiple Rx chains (*e.g.*, nine channels (1)) to handle concurrent uplink packets, they have only one Tx chain for downlink transmission. As a result, while multiple uplink packets can be received concurrently, their corresponding downlink packets have to be queued and processed one by one. As systems scale in supporting ever-increasing numbers of IoT devices, more and more downlink packets are being accumulated at network servers, leading to increased latency and excessive congestion. The asymmetry between downlink and uplink transmissions could also cause the power depletion of sensors in excessive packet retransmissions and even potential network collapse.

Current LoRaWAN transmits downlink packets sequentially (as illustrated in Figure 5.1(a)), creating a bottleneck that hampers overall network performance due to limited downlink capacity. In this chapter, we explore whether it is possible to improve downlink concurrency within the LoRaWAN architecture. We notice that LoRaWAN adopts massive logical channels (116; 4) for both uplink and downlink transmissions. The downlink packets queued at the network server could be assigned to orthogonal logical channels and transmitted simultaneously without causing much interference (Figure 5.1(b)). Rather than transmitting downlinks sequentially through a single logical channel as the existing LoRaWAN, we aim to scale concurrent downlink transmissions to massive logical channels. By enabling downlink concurrency, we aim to fundamentally break the barrier of the current downlink transmission paradigm and fill the gap caused by the asymmetry between the downlink and uplink transmissions to better support various IoT applications with diverse communication demands.

However, implementing the above idea into a practical system entails substantial technical challenges. Firstly, unlike traditional LoRaWAN which transmits only one downlink packet via a single gateway, we aim to support concurrent downlink transmissions over multiple logical channels through available gateways. However, current LoRa gateways (143; 144) face a dilemma: allocating more gateways for downlink transmissions would miss uplink packet receptions. Previous full-duplex radio designs (145; 146) cannot support full-duplex LoRa communication because of high self-interference of gateways to the ultra-weak uplink packets over long ranges. Moreover, concurrent downlink over logical channels would divide the transmit power and increase the demodulation noise floor, potentially compromising the reliability of downlink packet reception at low-cost COTS end nodes. Scaling to more gateways even requires a new scheduling strategy and faces inter-gateway interference. Therefore, it is crucial, albeit challenging, to enhance downlink performance without

compromising reliability and uplink reception.

To address the practical challenges, this chapter presents *FDLoRa* which supports in-band full-duplex communications for LoRa gateways. We find that uplink and downlink transmissions in LoRaWAN can be configured to adopt orthogonal modulations that cause minimum self-interference. By leveraging this orthogonality, we develop self-interference cancellation methods to further reduce the impact of downlink transmissions to the receptions of ultra-weak uplink packets. By doing so, *FDLoRa* can make the full use of wireless spectrum by simultaneously supporting downlink transmissions and uplink receptions. Unlike traditional full-duplex designs (146; 145), *FDLoRa* leverages the unique LoRa modulation and develops novel self-interference cancellation algorithms to further eliminate the self-interference below the noise floor. To unlock the potential of full-duplex gateways, we develop a new downlink framework that transmits downlink packets on multiple logical channels simultaneously and utilizes all available gateways. Based on the global information of instant link qualities and network dynamics, *FDLoRa* predicts the physical layer performance of concurrent downlinks to ensure communication reliability. The prediction results can further guide packet configurations (*e.g.*, channel selection, data rate adaptation, power control) and downlink scheduling across gateways.

We implement and evaluate *FDLoRa* using commercial off-the-shelf (COTS) LoRa devices and Software Defined Radio (SDR). Results show that *FDLoRa* significantly outperforms existing LoRaWAN downlink. Specifically, *FDLoRa* supports up to 5.16× downlink concurrency per gateway and scales the network downlink capacity by 5.7× than current LoRaWAN. *FDLoRa* can be seamlessly integrated with existing LoRaWAN architecture and hardware with software modification on the network server.

In summary, this chapter makes the following contributions: (1) *FDLoRa* presents the-first-of-its-kind gateway design that enables in-band simultaneous downlink trans-

mission and uplink reception, fully unlocking the spectrum and gateways for concurrent downlink transmission without compromising uplink; (2) *FDLoRa* introduces a new downlink framework to scale concurrent downlink transmissions to massive logical channels through available gateways; (3) We propose novel techniques to cancel the self-interference in in-band full-duplex LoRa communication and enable concurrent downlink to massive end nodes over logical channels and gateways; (4) We implement and evaluate *FDLoRa* to validate its effectiveness. Our evaluations demonstrate significant downlink scalability gains than LoRaWAN and state-of-the-art.

5.2 Background and Motivation

In this section, we introduce the background of LoRa (§5.2.1) and analyze the root cause of downlink-uplink asymmetry in LoRa networks and its impact on network functions and applications (§5.2.2). Finally, we present our motivation of full-duplex gateways and concurrent downlink transmissions over massive logical channels (§5.2.3).

5.2.1 LoRa Background

LoRa PHY. LoRa employs chirp spread spectrum (CSS) modulation to enable long-range communication while maintaining low power consumption (111; 81; 112; 113; 147; 148; 114). CSS utilizes chirps to modulate symbols. The duration of a chirp is defined by Spreading Factor (SF) and bandwidth (BW) parameters, *i.e.*, $T = \frac{2^{SF}}{BW}$. As shown in Figure 5.2(a), a *base up-chirp* whose frequency increases from $-\frac{BW}{2}$ to $\frac{BW}{2}$ can be represented as follows:

$$C(k,t) = e^{j2\pi(\frac{1}{2}kt - \frac{BW}{2})t} \tag{5.1}$$

where the chirp slope $k = \frac{BW}{T}$ denotes the frequency changing rate.

LoRa modulates a symbol by varying the initial frequency of an up-chirp. As shown in Figure 5.2(b), we denote a modulated symbol $S(f_{sym}, k, t)$ as follows:

$$S(f_{sym}, k, t) = C(k, t)\ e^{j2\pi f_{sym} t} \tag{5.2}$$

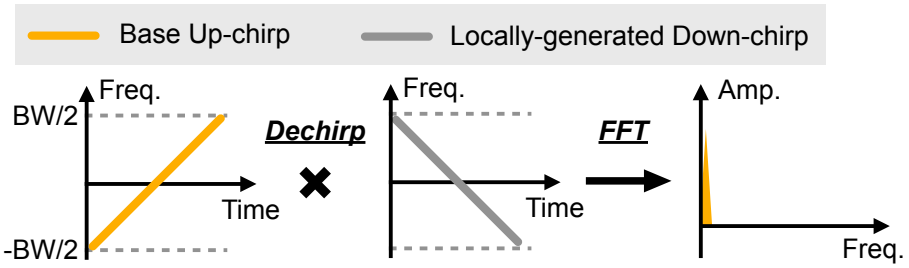
To demodulate the symbol, a LoRa receiver performs *dechirp* by multiplying $S(f_{sym}, k, t)$ with a *down-chirp* represented as $C^{-1}(k, t)$, which is the conjugate of a base up-chirp. The operation is represented as follows:

$$S(f_{sym}, k, t)\ C^{-1}(k, t) = e^{j2\pi f_{sym} t} \tag{5.3}$$

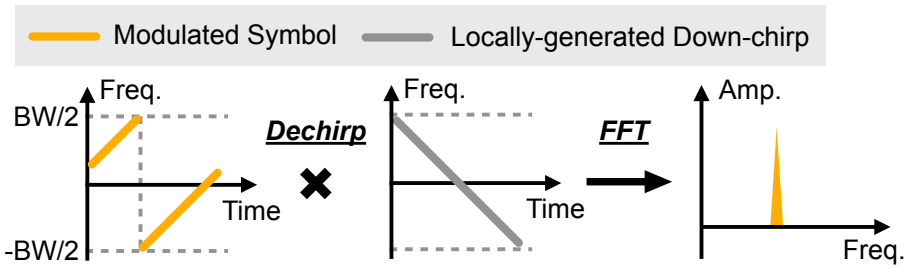
The *dechirp* operation eliminates the CSS modulation and transforms the LoRa symbol into a single tone. Notably, the symbol energy that initially spread across the chirp bandwidth is migrated at the tone frequency after *dechirp*. Finally, Fast Fourier Transform (FFT) aggregates the signal energy into a peak at f_{sym} on the frequency domain, which indicates the symbol. Essentially, FFT not only extracts the frequency itself but also accumulates the whole chirp energy into a single FFT bin, which yields a substantial *SNR gain after dechirping*. The SNR gain elevates the LoRa symbol above the ambient noise, facilitating long-distance and below-the-noise communications.

A LoRa packet initiates with a preamble, continues with two up-chirps serving as synchronization words and 2.25 down-chirps as a Start Frame Delimiter (SFD), and ends with a packet payload. The preamble, synchronization words, and SFD serve as the header of the LoRa packet and have a fixed signal pattern. In contrast, the payload carries modulated symbols with different initial frequencies.

LoRa Channels. LoRa allows nodes to select and configure channel frequency and bandwidth (BW) before packet transmissions. The channel frequency and bandwidth uniquely define a *physical channel*. Additionally, multiple LoRa nodes can simultaneously transmit in the same physical channel but with different spreading



(a) Base Up-chirp



(b) Modulated Symbol

Figure 5.2: Chirp Spread Spectrum of LoRa.

factors (SFs), creating multiple orthogonal *logical channels* (4). Different logical channels are orthogonal because they use different chirp slopes. When receiving over the logical channel with chirp slope k, the interference from another logical channel with chirp slope k' after dechirp can be represented as follows:

$$S(f_{sym}, k', t)\ C^{-1}(k, t) = e^{j2\pi[\frac{1}{2}(k'-k)t + f_{sym}]t} \tag{5.4}$$

In the above equation, we see the signal frequency after dechirp still changes over time, since $k' \neq k$. As a result, the chirp energy will spread over the spectrum rather than concentrating into a single peak after FFT (if $k' = k$). Hence, chirps with different slopes can be largely considered orthogonal after dechirping.

LoRa Network Architecture. A LoRa network typically consists of massive end nodes, multiple gateways, and a network server. Gateways are equipped with multiple Rx chains to receive uplink packets simultaneously, and one Tx chain for downlink transmission. Generally, gateways forward received packets to the network

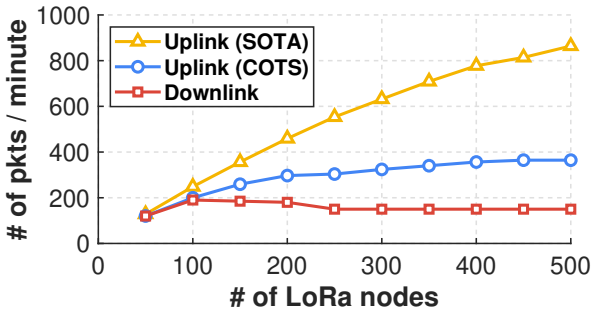


Figure 5.3: Downlink-uplink asymmetry bottlenecks effective network capacity. SOTA indicates multi-channel reception (1) with collision recovery (2).

server over the backhaul. The network server implements and optimizes various network functionalities such as uplink consolidation, downlink scheduling, and network resource optimization, with a global view of network status (149).

5.2.2 Understanding Downlink Bottleneck

Need for robust LoRa downlink. LoRa and other LPWAN technologies are designed for IoT communications, where the uplink is dominant, and the downlink requirements are typically minimal (*e.g.*, <10 downlink packets per day (150)). However, recent booming IoT applications pose an increasing demand on downlinks. For example, emerging applications like smart lamppost control systems (37) require frequent status acquisition and device control through downlink communication. Even in uplink-dominant applications such as smart agriculture and environmental monitoring, practical systems still rely on various downlink services such as data rate adaptation, uplink acknowledgment, and firmware updates to facilitate essential network operations. Considering the massive IoT devices deployed worldwide (27), efficient and reliable downlink communication becomes increasingly important.

Downlink-uplink asymmetry. Current LoRaWAN faces significant downlink-uplink asymmetry. As shown in Figure 5.3, our measurements*reveal that the down-

*We conduct measurements on a three-gateway testbed with fifty nodes and emulate more nodes by increasing the transmit duty cycle. Nodes randomly select uplink/downlink channel for each

link remains highly constrained (*i.e.*, <200 packets per minute) as the number of LoRa nodes increases. In contrast, the uplink capacity continues to grow with the support of multi-channel gateways with concurrent transmission capabilities (31; 2). As a result, we observe high latency and excessive congestion because the downlink capacity cannot catch up with the relatively high uplink transmissions. Moreover, as more nodes keep retransmitting the same unacknowledged packets in vain of downlink feedback, the downlink-uplink asymmetry leads to ineffective network transmissions and power depletion of nodes.

Limitations of current downlink. The main reason for downlink-uplink asymmetry lies in three folds. Firstly, as most gateways (144; 143) operate in the half-duplex mode, downlink transmissions can conflict with uplink. Recent work has made tremendous progresses but only enables *out-of-band* full-duplex, where the uplink and downlink spectra need to be separated (151; 152). The *out-of-band* full-duplex limits their applicability to specific LoRaWAN regional specifications, such as AU915 and US915 (151), and results in doubled spectrum usage. Secondly, gateways suffer from high asymmetry in uplink and downlink concurrency. Commodity gateways employ multiple Rx chains (*e.g.*, nine channels (1)) to receive concurrent uplink reception, while only one Tx chain is reserved for downlink transmission. Lastly, the gateway utilization is also asymmetric. LoRa networks typically deploy multiple gateways to collect uplink transmissions simultaneously. In contrast, current LoRaWAN only selects one network server and gateway at a time for downlink transmission (153; 44).

5.2.3 Motivation

The downlink-uplink asymmetry limits network operations in embedded AI and sensing systems, calling for novel solutions. We observe that conventional LoRaWANs typically employ half-duplex gateways and transmit downlink messages one by one

transmission.

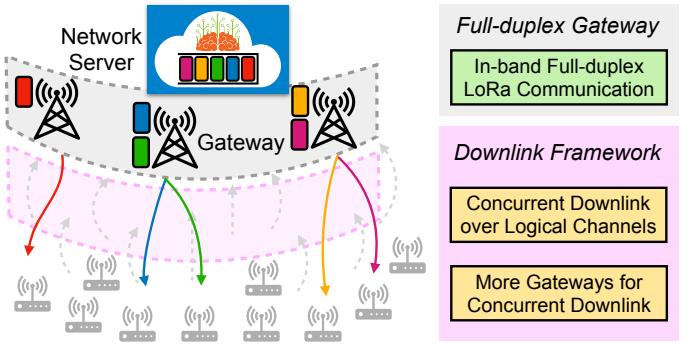


Figure 5.4: System overview: *FDLoRa* develops full-duplex gateways and a new downlink framework to tackle downlink-uplink asymmetry.

through a single gateway. This design paradigm fundamentally limits the downlink capacity. To deal with resource conflict, we propose *a new in-band full-duplex LoRa gateway design – FDLoRa*, to make the full use of available network resources without compromising uplink reception. In addition, we aim to scale the downlink transmissions across massive logical channels and enable concurrent downlink transmissions. We note that *FDLoRa* can be seamlessly integrated with existing LoRaWANs without any hardware or software modification to deployed COTS LoRa nodes. Compared with half-duplex and out-of-band full-duplex designs, *FDLoRa* can offer broader feasibility for various channel divisions, better spectrum utilization efficiency, and hold greater promise for representative applications such as real-time data transmission, rapid response to safety commands, and efficient management of high-density networks (154).

5.3 System Overview

To tackle the downlink-uplink asymmetry of LoRaWAN, our work *FDLoRa* enables in-band full-duplex gateways with concurrent downlink transmission capabilities. The workflow of *FDLoRa* is shown in Figure 5.4.

To resolve the resource conflicts between uplink and downlink, *FDLoRa* develops

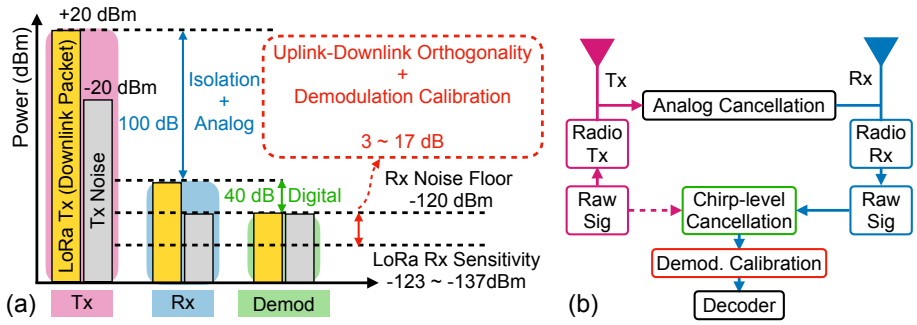


Figure 5.5: Full-duplex LoRa gateway: (a) Requirements of self-interference cancellation; (b) full-duplex cancellation architecture of *FDLoRa*.

in-band full-duplex LoRa gateways. *FDLoRa* addresses unique challenges involved in achieving in-band full-duplex LoRa communication without compromising either uplink or downlink performance.

Building upon the full-duplex gateways, *FDLoRa* introduces a novel concurrent downlink transmission framework to substantially improve the scalability of downlink transmissions. *FDLoRa* leverages massive LoRa logical channels to enable one gateway to concurrently send multiple downlink packets that can be readily received by deployed COTS LoRa nodes. *FDLoRa* also utilizes all available gateways for downlink transmission. By filling the gap between downlink and uplink (see Figure 5.3), *FDLoRa* is envisioned to meet the practical diverse demands of current and emerging IoT applications (37; 38; 39).

5.4 In-band Full-duplex Gateway

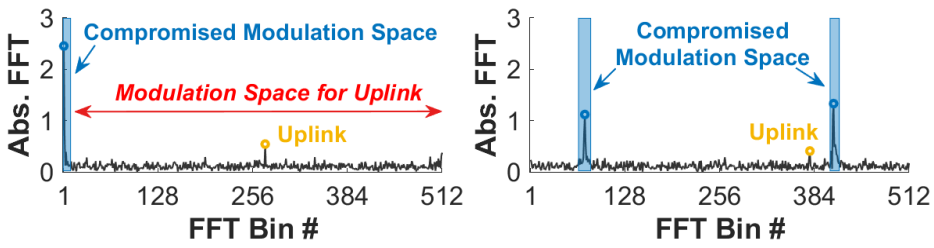
In this section, we introduce the design of *FDLoRa* in-band full-duplex gateway, which enables simultaneous downlink transmission and uplink reception. We first identify the requirements of an in-band full-duplex LoRa gateway and introduce the technical challenges (§5.4.1). Next, we present our solutions (§5.4.2).

5.4.1 Requirements of in-band full-duplex LoRa gateway

The key challenge of developing an in-band full-duplex LoRa gateway is to cancel the strong *self-interference (SI)* to uplink packets. As illustrated in Figure 5.5, a gateway's downlink transmission consists of the modulated signals of a downlink packet and the hardware noise of the transmitter, both of which can be orders of magnitude stronger than the incoming uplink packets received over long communication distances. The SI can be times stronger in an in-band full-duplex radio than that of an out-of-band full-duplex radio (*e.g.*, WisGate RAK7285 (152)) because SI degrades fast across channels. However, an out-of-band full-duplex radio occupies extra channels that could otherwise be used for more LoRa transmissions.

To enable in-band full-duplex communication without compromising Rx sensitivity, it is crucial to minimize and cancel the in-band self-interference from the downlink transmissions (both downlink packet and Tx noise), which is known as *Self-Interference Cancellation (SIC)*. Existing works (145; 155; 156) typically adopt a three-stage solution on both analog and digital domains: (1) Isolation between Tx-Rx antennas; (2) Analog cancellation; and (3) Digital cancellation. As digital cancellation alone cannot completely remove SI and Tx noises (145), we need to preliminarily suppress SI and Tx noises through analog components before digital processing. To achieve this, the Tx and Rx antennas of a full-duplex radio are physically separated apart (157) and wire-connected with a specifically designed analog circuit for SIC (146). Such antenna isolation and analog cancellation techniques suppress SI and Tx noises by >100 dB (146) (see Figure 5.5(a)). Subsequently, digital cancellation leverages the prior knowledge of the Tx signal to remove the residual SI signals in the digital domain, which can further reduce interference by around 40 dB and suppress SI to the same power level of Rx noises.

Limitations of state-of-the-art (SOTA) full-duplex radios. Given the high



(a) Base chirp (preamble) (b) Non-base chirp (payload)

Figure 5.6: (a) Suppressing interference bins harms the effective modulation space for uplink. The uplink symbol is submerged if it lies close to interference bins (represented as blue zones). (b) The problem is more severe under payload self-interference.

sensitivity of a LoRa radio (*e.g.*, down to -137 dBm (82)), the self-interference from a gateway's downlink can still trigger packet detection and reception of the gateway even with the SI being suppressed close to the noise floor (*e.g.*, using a traditional SIC technique). It imposes extra requirements on the SIC of a full-duplex gateway to further suppress SI signals below the Rx sensitivity of a LoRa radio (*e.g.*, usually tens of dB lower than the noise floor (82)).

A recent work (146) incorporates new analog and digital SIC techniques that take account of LoRa signal features and use optimized circuits together with digital SIC to achieve a remarkable 117.7 dB SIC gain. The full-duplex gateway design is proven effective in mitigating the self-interference of a specific downlink busy signal consisting of the same base up-chirps, thereby reserving uplinks for end nodes and avoiding packet collisions (146). However, the previous work (146) cannot be applied to the SIC of in-band full-duplex LoRa communications in our scenario where the downlink carries LoRa data packets rather than pre-defined busy signals for media access control purposes as in (146).

In-band full-duplex LoRa communication faces two unique challenges that have yet to be addressed. Firstly, a normal downlink packet consists of not only base chirps in the preamble but also non-base chirps in the payload part. The existing LoRa SIC method (146) suppresses interference during demodulation by removing certain fre-

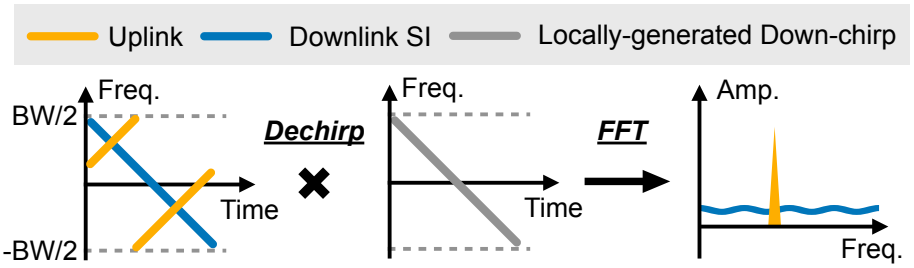
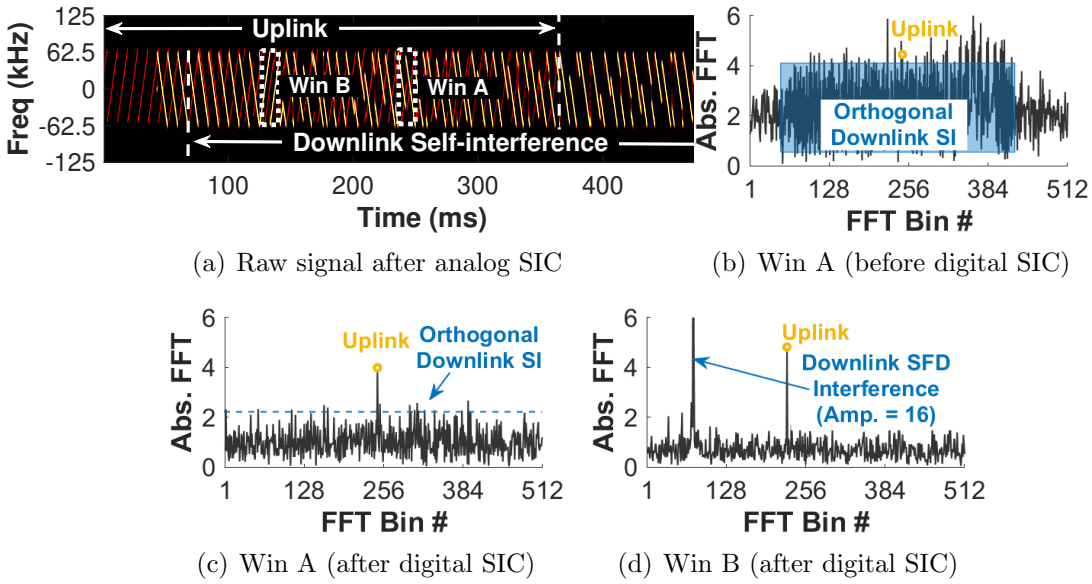


Figure 5.7: Uplink-downlink orthogonality: downlink self-interference can be separated with conjugate modulation.

quency bins, which would reduce the effective space for payload data modulation (see Figure 5.6(a)). Such a method is particularly difficult for the non-base chirps of LoRa payload because non-base chirps usually suffer more severe frequency misalignment and side lobe issues than base chirps (97). Given that downlink SI is much stronger than uplink signals, even the side-lobe bins of SI can overwhelm the target bin of a weak uplink. We have to remove multiple surrounding bins to effectively suppress interference, which can drastically reduce the modulation space of payload data (see Figure 5.6(b)). On the other hand, existing SIC methods (145; 146) require estimating channels from a preamble and applying channel parameters to the entire packet for digital cancellation. However, a LoRa packet has a long air-time (*e.g.*, hundreds of ms), while a preamble constitutes only a small portion of the packet (*e.g.*, <20 %). Channel parameters may change across a LoRa packet as the channel coherence time lasts only tens of ms (*i.e.*, shorter than a LoRa packet) (158). The channel estimated from a preamble may become outdated when applied to a payload and fail the digital cancellation. It is important yet challenging to acquire channel information from a packet at fine degrees to enable full-duplex LoRa communications.

5.4.2 SIC for Full-duplex Communication

FDLoRa implements an in-band full-duplex LoRa gateway by systematically mitigating the self-interference (SI) from downlink to uplink using both analog and digital



(a) Raw signal after analog SIC

(b) Win A (before digital SIC)

(c) Win A (after digital SIC)

(d) Win B (after digital SIC)

Figure 5.8: SIC for in-band full-duplex LoRa communication: (a) Spectra of Rx signal: downlink SI is 15 dB stronger than uplink; (b) Demodulation of an uplink symbol: orthogonal downlink SI overwhelms the uplink; (c) Digital cancellation suppresses downlink SI to noise floor; (d) Downlink SFD still interferes with uplink symbols.

SIC techniques. *FDLoRa* uses Tx/Rx antenna isolation and an analog SIC circuit to suppress strong downlink SI to an acceptable power level, allowing normal operations of Rx radio (*e.g.*, ADC) for sampling a weak uplink packet (146). Beyond the analog SIC, *FDLoRa* presents a new digital SIC solution that can effectively remove the SI from normal downlink LoRa packets to achieve in-band full-duplex communication. Unlike the existing digital SIC (*i.e.*, bin suppression (146)) that may damage the modulation space of uplink packets, *FDLoRa* cancels the SI from downlink without compromising (de)modulation performance or Rx sensitivity for uplink.

The basic idea of *FDLoRa* SIC solution can be described as follows: *FDLoRa* leverages the fact that the uplink and downlink in LoRaWAN can be configured with conjugated modulation schemes, *e.g.*, using normal up-chirps as the modulation basis for uplink but down-chirps (*i.e.*, the conjugate of up-chirps) for downlink. While

conjugated modulation schemes are used for uplink and downlink, the SI signals from downlink (*i.e.*, down-chirp) now become orthogonal to the uplink signals (*i.e.*, up-chirp). Such signal orthogonality allows *FDLoRa* to adopt a simple yet effective signal processing method (*e.g.*, dechirping) to efficiently cancel the downlink SI. For instance, as the signal received by the Rx antenna is a superimposition of down-chirps (*i.e.*, downlink SI) and up-chirps (*i.e.*, uplink signal), if we apply the demodulation method for up-chirps (*i.e.*, dechirping followed by FFT) to the superimposed signal as illustrated in Figure 5.7, down-chirps will not get demodulated because of the orthogonality between up-chirp and down-chirp. It can logically cancel the downlink SI during demodulation and only give symbols of the uplink packet as demodulation results. The (de)modulation performance for uplink will not be compromised when performing SIC.

However, it remains non-trivial to directly apply orthogonal uplink and downlink to an in-band full-duplex radio in practice to achieve digital domain SIC. The reason is that the downlink SI signals received by digital components of the radio, though being largely suppressed by analog SIC, may still be tens of dB stronger than uplink signals, and this large power disparity between uplink and downlink can destroy their signal orthogonality. For example, Figure 5.8(a) shows the spectrogram of signals received by our full-duplex radio after analog SIC but before digital SIC. We see brighter downlink SI signals than uplink signals (*i.e.*, around 15 dB difference in signal power). When we apply the demodulation method of uplink (*i.e.*, de-chirp with a standard down-chirp, then perform FFT) to the compound signals, we expect the dechirping operation spread the signal power of down-chirps (*i.e.*, downlink SI) but concentrate the energy of up-chirps into FFT bins that give us the demodulated symbols of the uplink. However, Figure 5.8(b) shows that the energy of the strong downlink SI after spreading into multiple bins is still comparable with (or even higher than) the accumulated energy of the uplink and interferes with the demodulation of

the uplink. It requires another digital cancellation procedure before demodulation that can further deduce downlink SI power to protect signal orthogonality between uplink and downlink for correction demodulation of an uplink.

To deal with the practical issues above, *FDLoRa* performs digital SIC in two steps: *FDLoRa* first employs a digital cancellation technique to remove downlink SI from the received compound signals of uplink and downlink, aiming at suppressing SI to the same level of Rx noises. After that, *FDLoRa* exploits the signal orthogonality between uplink and downlink to logically remove any residual downlink SI via a normal demodulation operation of the uplink. The method details are provided in the following.

Fine-grained digital cancellation. Digital cancellation uses a digital domain signal processing method to remove the signals of downlink SI from the received raw signals of an in-band full-duplex radio. Without loss of generality, we denote the raw signals ($y(t)$) digitized by a full-duplex radio as below.

$$y(t) = SI(t) + U(t) + n(t) \tag{5.5}$$

where $SI(t)$ denotes the downlink self-interference signal, $U(t)$ is the signal of an uplink packet, and $n(t)$ denotes Rx noises. The goal of digital cancellation is to remove $SI(t)$ from $y(t)$. Note that $SI(t)$ is a copy of the signal of a downlink packet that is sent out by the Tx antenna, traversing through the air and re-entering the Rx antenna of the same radio. In practice, a full-duplex radio has prior knowledge about the transmitted downlink signals (*i.e.*, denoted by $x(t)$), and thus we can model $SI(t)$ as $SI(t) = h_{tx-rx} \cdot x(t)$, where h_{tx-rx} is the channel from Tx antenna to Rx antenna. To perform effective digital cancellation, *FDLoRa* takes a three-step procedure: (1) use the known signal of downlink SI (*i.e.*, $x(t)$) to estimate channel h_{tx-rx} from $y(t)$; (2) use the estimated channel (*i.e.*, $\hat{h}_{tx-rx}$) to rebuild a received downlink SI as $\hat{SI}(t) = \hat{h}_{tx-rx} \cdot x(t)$; and (3) subtract $\hat{SI}(t)$ from $y(t)$.

To handle potential channel changes over a long duration of a LoRa packet (*e.g.*, hundreds of ms), *FDLoRa* proposes a fine-grained channel estimation method to estimate the channel of downlink SI (*i.e.*, h_{tx-rx}) on a per-chirp basis. In particular, as the Tx and Rx of a full-duplex radio share the same clock, the transmitted downlink signals and received SI do not have carrier frequency offsets (CFOs). We can correlate $y(t)$ with the known downlink signal ($x(t)$) to accurately locate the start of downlink SI in received signals. Then, we estimate h_{tx-rx}, reconstruct and cancel SI signals chirp-by-chirp. Recall that $SI(t)$ is orthogonal with but stronger than uplink signals $U(t)$ in $y(t)$. We can multiply the i^{th} chirp of $SI(t)$ with the conjugate of the corresponding chirp from $x(t)$ to estimate channel h_{tx-rx} as follows:

$$\begin{aligned} y^i(t) \cdot [x^i(t)]^* &= SI^i(t) \cdot [x^i(t)]^* + U^i(t) \cdot [x^i(t)]^* \\ &\approx h^i_{tx-rx} x^i(t) \cdot [x^i(t)]^* = h^i_{tx-rx} \end{aligned} \tag{5.6}$$

where $(\cdot)^i$ denotes the i^{th} chirp and $(\cdot)^*$ denotes a complex conjugate. Equation 5.6 can remove the common down-chirp signal shared by $SI^i(t)$ and $x^i(t)$, spread the uplink chirp $U^i(t)$ to noise levels and produce a tone-frequency of the SI. We can apply FFT to accumulate signal power of the tone-frequency to a high peak from which $\hat{h}^i_{tx-rx}$ can be extracted. After traversing the chirps in $SI(t)$ and $x(t)$, we can use the extracted fine-grained channel information $\hat{h}_{tx-rx} = \{\hat{h}^i_{tx-rx}\}$ to rebuild $\hat{SI}(t)$. Next, we align $\hat{SI}(t)$ with $SI(t)$ and finally subtract it from $y(t)$ for digital cancellation of the SI signals.

Figure 5.8(c) shows the demodulation results of the same signal from Figure 5.8(b) after a fine-grained digital cancellation. Comparing the two figures, we can see that the strong downlink SI shown in Figure 5.8(b) has been successfully suppressed to a noise level.

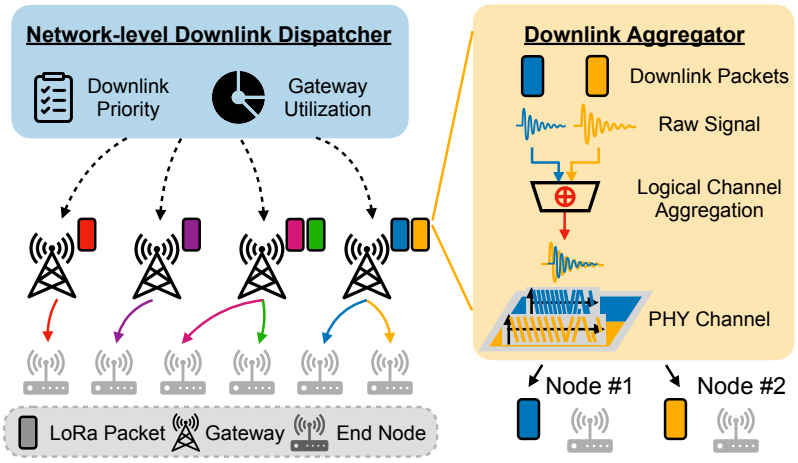


Figure 5.9: The downlink framework of *FDLoRa*.

5.5 *FDLoRa* Downlink Framework

Handling downlink SFD interference. With the above digital cancellation method, we have effectively suppressed the power strength of downlink SI. The residual downlink SI, if any, can next be logically mitigated during the demodulation of the uplink by leveraging signal orthogonality between the uplink and the downlink. However, as a practical downlink LoRa packet also contains 2.25 up-chirps as the Start Frame Delimiter (SFD) of the packet, the SFD chirps are not orthogonal to the data chirps of uplink packets and may interfere with the uplink demodulation when the downlink SI is not completely removed, as shown in Figure 5.8(d). Our study reveals that although such interference from downlink SFD only impacts the demodulation of 2∼4 symbols of an uplink packet, it can lead to 8% more packet loss on uplink reception (see Figure 5.15).

FDLoRa handles downlink SFD interference from the residual SI signals through two techniques. Firstly, as the transmitted downlink signals are known to a full-duplex radio, *FDLoRa* can precisely detect the start of downlink SI in received signals and track the location of downlink SFD based on the prior knowledge about the frame structure of a downlink packet. With the located downlink SFD information,

FDLoRa can accurately predict the FFT bins of a downlink SFD chirp in an affected demodulation window of the uplink as shown in Figure 5.8(d), and thus interpret those bins as SFD interference and extract other FFT peaks as the demodulated symbol of uplink. Secondly, we observe that symbols from the same uplink packet usually have the same FFT amplitude and differ from the amplitude of downlink SFD chirps. As downlink SFD only affects 2∼4 uplink symbols, *FDLoRa* can extract the amplitude of uplink symbols from the demodulation windows not affected by downlink SFD and use the amplitude as a feature to differentiate uplink symbols from downlink SFD in those affected windows. *FDLoRa* jointly employs the two techniques to reliably demodulate symbols of an uplink without compromising Rx sensitivity.

To unlock the potential of full-duplex LoRa gateways, *FDLoRa* introduces a novel downlink framework that utilizes multiple logical channels of all available gateways for concurrent downlink transmissions. We aim to address the limitations of the current LoRa downlink, *i.e.*, each gateway only supports a single downlink channel, and the network server only selects a single gateway for downlink transmission.

To breakthrough the bottleneck, as shown in Figure 5.9, our downlink framework adopts a two-tier structure to scale concurrent downlink transmissions over multiple logical channels (§5.5.1) through multiple gateways (§5.5.2). First, *FDLoRa* includes a *downlink aggregator* that empowers a gateway to serve multiple nodes by transmitting concurrent downlink packets over multiple logical channels. Second, *FDLoRa* employs a *network-level downlink dispatcher*, which coordinates and dispatches concurrent downlink packets through multiple gateways. In the following, we present the design of *FDLoRa* downlink framework in detail.

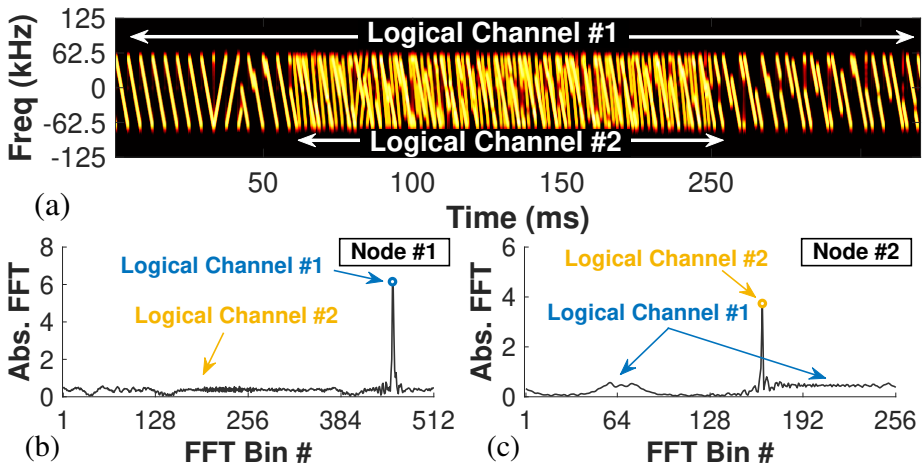


Figure 5.10: Concurrent downlink over logical channels: (a) Spectra of aggregated logical channels; (b,c) LoRa node #1 and node #2 can extract and receive over different logical channels from the aggregated PHY signal.

5.5.1 Concurrent Downlink over Multiple Logical Channels

Logical channel aggregation. *FDLoRa* leverages the orthogonality of LoRa logical channels to enable concurrent downlink transmissions from one gateway to multiple nodes. *FDLoRa* develops novel techniques to aggregate multiple logical channels in a unified Tx raw signal while maintaining compatibility with COTS LoRa nodes. This allows *FDLoRa* to send concurrent downlink packets over multiple logical channels to LoRa nodes without any hardware or software extensions to the nodes. As illustrated in Figure 5.9, *FDLoRa* takes the concurrent downlink packets from the network server as input, and then generates a unified raw signal containing multiple downlink packets. A unique challenge here is that the concurrent downlink packets have diverse demands for delay and power, as they target different nodes with varying Rx windows and link qualities. To address this, *FDLoRa* assigns each packet with unique PHY features before aggregation to meet the delay and power requirements. Formally, we represent the aggregated Tx raw signal as follows:

$$x(t) = \sum_{i=1}^{N} \alpha_i \cdot P_i(t - t_i) \tag{5.7}$$

where $P_i(t)$ denotes the raw signal of the i^{th} downlink packet, α_i and t_i represent the transmit power and time delay of $P_i(t)$, respectively.

Reception at COTS LoRa nodes. Concurrent downlink reception of the aggregated packets over logical channels requires no modification to COTS LoRa nodes. Figure 5.10(a) illustrates the aggregated downlink packets over two concurrent logical channels targeting at two COTS nodes configured with SF9 and SF8, respectively. The received signal of node #1 can be represented as:

$$y_1(t) = h_1\alpha_1 \underbrace{S(f_{sym1}, k_1, t)}_{\text{Signal of logical ch.\#1}} + h_1\alpha_2 \underbrace{S(f_{sym2}, k_2, t)}_{\text{Signal of logical ch.\#2}} \tag{5.8}$$

where $0 \leqslant t \leqslant T$ and h_1 denotes the channel from the gateway to node #1. Node #1 monitors incoming chirps with SF9 (*i.e.*, slope $k = k_1$) to extract the target chirps from the aggregated downlink signal, with the demodulation results as below (according to Equation 3.3 and 5.4).

$$y_1^{demod}(t) = h_1\alpha_1 \underbrace{e^{j2\pi f_{sym1}t}}_{\text{Single tone}} + h_1\alpha_2 \underbrace{e^{j2\pi[\frac{1}{2}(k_2-k_1)t+f_{sym2}]t}}_{\text{Time-varying frequency}} \tag{5.9}$$

After performing FFT, the signal from logical channel #1 (single tone) is accumulated to a distinct energy peak. In contrast, the signal from logical channel #2 (Time-varying frequency) disperses across the spectrum (see Figure 5.10(b)). We can observe similar demodulation results for node #2 on logical channel #2 in Figure 5.10(c). The experiment results show that by leveraging the logical channel orthogonality, COTS LoRa nodes can extract and receive their individual downlink packets from the aggregated downlink signal without any hardware or software modification.

Downlink quality prediction and power allocation. While the LoRa nodes can receive their individual downlink packets from the aggregated signal, this approach entails practical challenges. Firstly, transmitting multiple packets would

inevitably divide the transmit power of a gateway, potentially affecting the communication range and reliability. Second, signals from concurrent logical channels can spread across the spectrum rather than being perfectly canceled after dechirping as illustrated in Figure 5.10(b, c). The increased noise floor can negatively affect the reception of the target packet for each LoRa node.

To ensure downlink reliability, we adopt a novel downlink quality prediction method for concurrent logical channels. *FDLoRa* rehearses the aggregated physical layer signal arriving at each target node based on their link quality estimation. Next, it predicts the downlink demodulation performance of LoRa nodes and computes the minimum transmit power required for each downlink packet (*i.e.*, $\{\alpha_i^{min}\}$).

Based on the downlink quality prediction, our goal is to maximize the number of concurrent downlink transmissions while meeting the transmit power requirements as presented in (5.10) below. λ_i equals 1 if the packet P_i can be transmitted, else it equals 0. A denotes the maximum transmit power of the gateway.

$$\begin{aligned} &\max \quad \sum_{i=1}^{N} \lambda_i \\ &s.t. \quad \sum_{i=1}^{N} \lambda_i \alpha_i^{min} \leqslant A, \quad \lambda_i \in \{0,1\} \end{aligned} \tag{5.10}$$

This problem can be formulated as a *Knapsack Problem* (159), which is known to be NP-complete. In practice, there are at most 6 logical channels (*i.e.*, SF 7∼12 (151)). The maximum Tx power of the LoRa gateway is 20 dBm (144), and we set the minimum power unit as 1 dBm. In practice, we can efficiently solve this problem in $O(NA)$, where $N \leqslant 6$, $A = 20\,\text{dBm}$, with a *Dynamic Programming* algorithm (160).

5.5.2 Coordinating All Available Gateways

Beyond concurrent downlink transmission of a single gateway, *FDLoRa* also utilizes all available gateways to further enhance downlink performance. However, scaling to more gateways is a non-trivial task. Firstly, dispatching concurrent downlink transmissions to multiple gateways is more complex. Traditional LoRaWAN selects one gateway with the highest SNR for downlink transmission (153). This approach cannot be applied to schedule more concurrent downlink transmissions of multiple gateways. Moreover, utilizing multiple gateways simultaneously for downlink communications could potentially cause *inter-gateway interference*.

Dispatching concurrent downlink packets. *FDLoRa* introduces a *network-level downlink dispatcher* to optimize gateway utilization and maximize network-level downlink concurrency. Measurement studies (29) have shown that due to the long communication range most LoRa nodes can be simultaneously covered by multiple gateways, while a few LoRa nodes can only connect to a single gateway due to signal fading and blockage. To address practical LoRa deployments, our dispatching algorithm is designed as follows. Firstly, we identify the downlink packets whose target nodes can only connect to one gateway, and assign them to the corresponding gateways. Secondly, we arrange the other downlink packets that have multiple optional gateways. We allocate the downlink packets with orthogonal logical channels to the same gateway to maximize the network-level downlink concurrency. Finally, since the second step may lead to unbalanced allocation, we revisit the gateway availability and balance the workload if there are idle gateways. The algorithm traverses the downlink packets two times and can be completed in $O(N)$, which is evaluated in Figure 5.16(b).

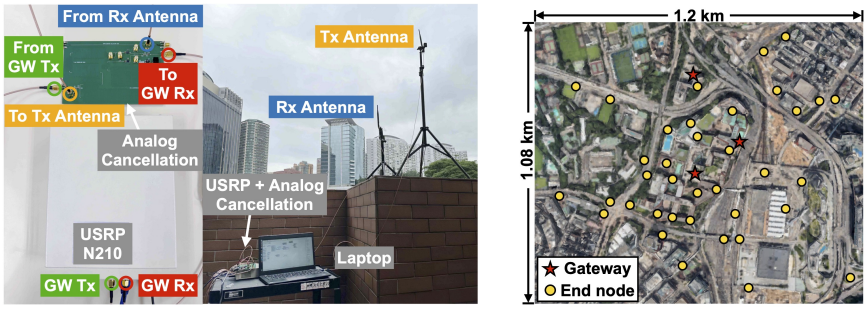
Resolving inter-gateway interference. Utilizing multiple gateways simultaneously faces inter-gateway interference problem. A gateway can potentially receive

the downlink signal transmitted by other gateways. The uplink packet reception can be influenced if the interference from other gateways unfortunately overlaps with its own uplink. Analog cancellation circuits can help, but it is not feasible as gateways are geographically separated. To mitigate the impact of inter-gateway interference in concurrent downlink transmissions, *FDLoRa* identifies downlink packets that could potentially cause inter-gateway interference (*i.e.*, using the same logical channel) and assigns them to gateways that are as far apart as possible, thereby exploiting the wireless signal attenuation between gateways. As illustrated in Figure 5.5(a), the wireless signal attenuation acts as an analog SIC in the air to suppress the inter-gateway interference. *FDLoRa* then applies the digital methods described in §5.4.2 to further cancel the residual interference signal. Note that such a digital cancellation method only requires the data sharing of concurrent downlink packets and their meta-information (but not raw PHY samples) among neighbor gateways which can be easily handled by the backhaul network between gateways. The requirement of signal attenuation between two gateways is to attenuate the Tx noise of one gateway to the Rx noise floor of the other gateway (see Figure 5.5(a)). Otherwise, the conflicted packets should be sent at different times.

5.6 Evaluation

5.6.1 Methodology

Implementation. We implement *FDLoRa* on software-defined radios (USRP N210) based on the `gr-lora` project (86). Each full-duplex LoRa gateway is integrated with an analog cancellation circuit (146) between USRP and Tx/Rx antennas as shown in Figure 5.11(a). The Tx/Rx antennas are vertically separated for antenna isolation. We use COTS LoRa nodes with Semtech SX1276 radio (82) as transceivers and Arduino Uno boards for node configuration. We build a testbed consisting of 50



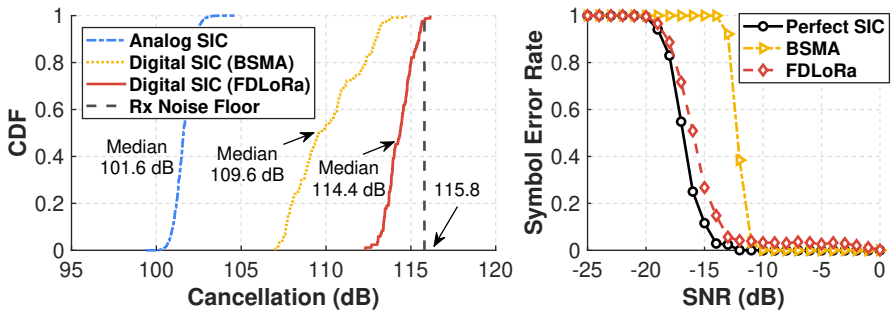
(a) Gateway implementation (b) Campus-scale testbed

Figure 5.11: (a) USRP gateway connects with analog cancellation circuit; (b) Deployment map of our testbed.

LoRa nodes and 3 gateways, where each gateway is connected to a laptop running *FDLoRa*. The gateways are connected to a network server for gateway coordination. We conduct experiments on our campus, covering an area of 1.08 km × 1.2 km in a typical urban environment (see Figure 5.11(b)). All nodes operate in the 915 MHz ISM band. We use the Arduino-LMIC library (161) to configure the LoRa nodes to apply down-chirp as the modulation basis for downlink packets. The LoRaWAN benchmarks are implemented with RAK7268 COTS LoRa gateways (144).

Experiment Setup. We collect data traces for over three months from >200 links in our testbed. The traces cover various channel conditions in urban settings, including indoor and outdoor, low SNR, and high SNR. All evaluations with ⩽50 nodes and ⩽3 gateways are conducted via real-world experiments in the testbed. In addition, we also perform large-scale trace-driven evaluations based on the data traces collected in the testbed. Our evaluation primarily focuses on the following questions: (1) How does *FDLoRa* perform in full-duplex LoRa communication? (2) How many concurrent downlink transmissions can be supported by *FDLoRa*? and (3) How does *FDLoRa* perform in supporting practical IoT applications?

Baselines. We compare *FDLoRa* with four baselines on both full-duplex gateway and downlink transmission. (1) *LoRaWAN* (27) transmits one downlink packet



(a) Self-interference cancellation (b) Weak uplink reception

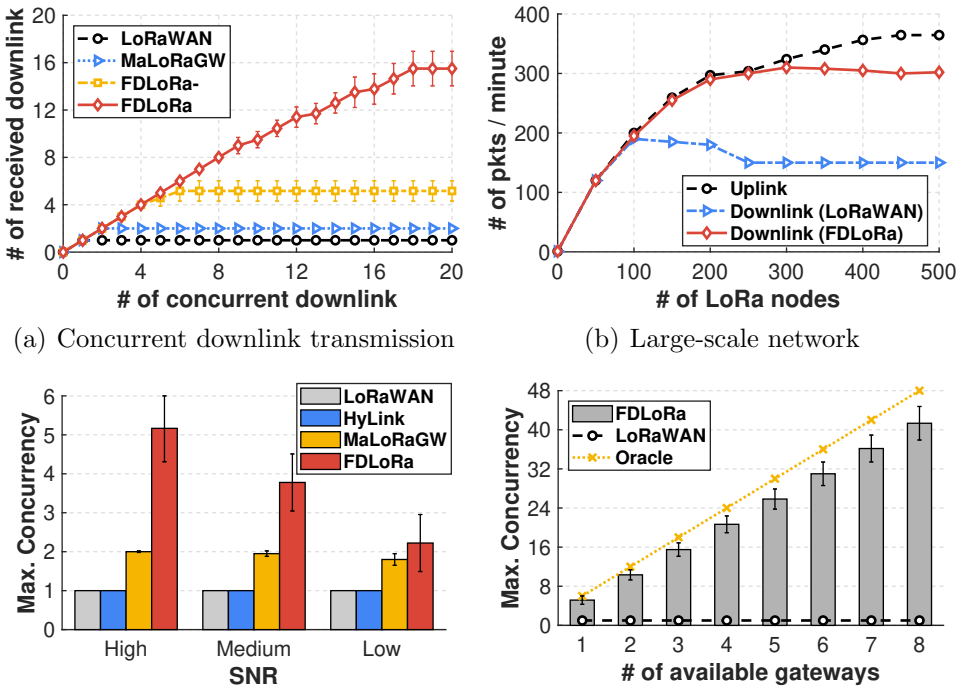
Figure 5.12: Performance of full-duplex communication.

through a single gateway; (2) *MaLoRaGW* (162) adopts two-antenna gateway to transmit concurrent downlinks by beamforming; (3) *HyLink* (163) modulates multiple chirps in the same packet to enhance link throughput; and (4) *BSMA* (146) applies digital cancellation methods designed for full-duplex busy signal transmission.

5.6.2 Basic Performance

Full-duplex LoRa communication. We first evaluate *FDLoRa* in supporting full-duplex LoRa communication. We set up one gateway to transmit downlink packets and receive uplink simultaneously. Figure 5.12(a) presents the full-duplex SIC performance over 20 seconds. We see that antenna isolation and analog cancellation combined provide 101.6 dB SIC. The digital cancellation of BSMA (146) provides additional 8 dB for downlink packets, which is insufficient for receiving weak uplink. *FDLoRa* utilizes a fine-grained digital cancellation method that achieves 12.8 dB and thereby takes the downlink SI to the noise floor.

We further examine the uplink reception sensitivity of *FDLoRa* (below the noise floor) under full-duplex downlink transmission. We set up one LoRa node at SF10 under different SNR conditions and analyzed >100 packets for each SNR condition. The gateway receives uplink while transmitting downlink packets. We also



(a) Concurrent downlink transmission

(b) Large-scale network

(c) Comparison with SOTA

(d) Scaling with more gateways

Figure 5.13: Performance of downlink scalability: (a) different number of concurrent transmissions; (b) downlink packet rate in large-scale networks; (c) comparison with state-of-the-art; (d) concurrency with more gateways.

test the uplink-only scenario as the perfect SIC for comparison. As illustrated in Figure 5.12(b), the perfect SIC performs best as it has no self-interference from downlink. The bin suppression method (146) does not work well because payload symbols of SI have strong side lobe bins that overpower the weak uplink. *FDLoRa* can receive the weak uplink at ultra-low SNR (*e.g.*, −14.5 dB for SF10) by leveraging the signal orthogonality and calibrating the SFD interference. Compared to the uplink-only gateway, the full-duplex communication in *FDLoRa* only leads to 1.1 dB SNR loss on uplink reception while achieving <20% symbol error rate.

Gains on downlink concurrency. We next evaluate the performance of *FDLoRa* in supporting concurrent downlink transmissions. We control a varying number of LoRa nodes to receive concurrently on different logical channels. We compare

FDLoRa against LoRaWAN and MaLoRaGW (162). LoRaWAN uses three COTS gateways, and MaLoRaGW adopts a two-antenna gateway for downlink transmission, respectively. Both baselines are coordinated by a network server. The nodes use SF9, as the baselines can only transmit downlink packets on a single logical channel. *FDLoRa* contains three full-duplex gateways. To fully evaluate the gain of full-duplex communication for unlocking the downlink spectrum and available gateways, we implement a version of *FDLoRa* without full-duplex techniques, which utilizes only one gateway for downlink transmission, represented as *FDLoRa-* in the evaluation. The nodes use SF7∼12, as *FDLoRa* can transmit concurrent downlink packets on multiple logical channels.

Figure 5.13(a) illustrates the number of received downlink packets under different number of concurrent downlink packets. The number of received downlinks first increases and then reaches the maximum downlink concurrency and stops growing. As expected, LoRaWAN supports at most one downlink packet at a time. MaLoRaGW supports two concurrent downlinks but is still limited by the half-duplex design. In contrast, *FDLoRa-* can transmit an average of >5 concurrent packets over multiple logical channels through a single gateway. The maximum concurrency increases to >15 when unlocking all gateways for downlink with full-duplex communications of *FDLoRa*. The downlink concurrency gain of *FDLoRa* is mainly because of its full-duplex communication capability. By breaking the conflict between uplink and downlink transmissions, it can fully utilize the available gateways and logical channels to enhance downlink concurrency.

Downlink in large-scale networks. This experiment uses ACK as a representative downlink application to evaluate *FDLoRa* in large-scale networks. We deploy 50 LoRa nodes in our testbed area to compare *FDLoRa* against LoRaWAN (27). Both strategies consist of three gateways to cover the LoRa nodes. Each node transmits a *ConfirmedDataUp* message that requires downlink ACK with a duty-cycle

$\leqslant$1%. We modify the duty-cycle of LoRa nodes to emulate a larger number of nodes (*e.g.*, one node with 5% duty-cycle represents 5 nodes with 1% duty-cycle). The nodes randomly select the channel for each transmission. For a fair comparison, we use the standard LoRa decoder for uplink reception.

Figure 5.13(b) shows the packet rate for both uplink and downlink. We observe that although the uplink packet rate grows continuously as the number of nodes increases, the downlink packet rate of standard LoRaWAN stops at 100 LoRa nodes and is limited to <200 packets per minute. This significant asymmetry between the uplink and downlink prevents more nodes from receiving effective downlink services. In contrast, *FDLoRa* fills this gap by unlocking downlink logical channels of available gateways. Results show that *FDLoRa* improves the downlink packet rate to 300 packets per minute. It is only when the number of LoRa nodes exceeds 300 that the downlink begins to lag behind the uplink.

Comparison with state-of-the-art. This experiment evaluates *FDLoRa* against current leading LoRa downlink methods. We examine two state-of-the-art (SOTA) strategies (*i.e.*, HyLink (163) and MaLoRaGW (162)). To ensure a fair comparison, we focus on the maximum number of concurrent downlink transmissions each gateway can support. For HyLink (163), we implement both the transmitter and receiver with USRPs as it has to access the PHY raw signal. For MaLoRaGW (162), we employ two synchronized USRPs to facilitate 2×2 MIMO beamforming for the transmitter, while the reception is handled by COTS LoRa nodes. For *FDLoRa*, we aggregate multiple downlink logical channels and transmit with USRP, with COTS LoRa nodes serving as receivers. A standard LoRaWAN setup is also implemented for baseline comparison. The baselines use SF9 since they can only transmit downlink on a single logical channel. *FDLoRa* uses SF7∼12. We test three SNR scenarios: high (>5 dB), medium (-5∼5 dB), and low (<-5 dB), with the average communication range of <30 m, ∼100 m and >200 m.

Figure 5.13(c) displays the maximum downlink concurrency per gateway achieved by four strategies at different SNR conditions. As expected, LoRaWAN is limited to a single downlink transmission. HyLink enhances link throughput by encoding parallel chirps within a single packet payload, yet it fails to accommodate multiple clients simultaneously. MaLoRaGW achieves up to 2× concurrent transmissions at high SNR, with a slight reduction as SNR worsens. The reason is that the concurrency of multi-user MIMO is limited by the number of Tx antennas and the precision of channel estimation. *FDLoRa* outperforms the other strategies, achieving on average 5× and 2.2× concurrent transmissions at high and low SNR conditions, respectively. This superior performance is because *FDLoRa* exploits logical channel orthogonality of LoRa to enhance downlink capabilities, which is feasible with the gateway's single Tx chain and antenna. We notice that the downlink concurrency drops as the communication range increases. The reason is that low-SNR links require higher transmit power, but the budget is restricted by the maximum transmit power. Since the low-SF packets require higher transmit power to reach the same communication range as the high-SF ones, *FDLoRa* prioritizes the high-SF logical channels (*e.g.*, SF10∼12) to achieve higher concurrency for low-SNR links.

Scaling with more gateways. This experiment assesses the maximum number of concurrent downlink transmissions that *FDLoRa* can support with an increased number of gateways. We increment the number of gateways from 1 to 8 and configure various LoRa nodes to receive on different logical channels. We employ trace-driven evaluation to conduct experiments involving more than three gateways. To achieve this, we aggregate multiple downlink packet traces with random time offsets to simulate real-world network traffic. These synthesized traces are then replayed to assess the performance of *FDLoRa* in scenarios with over three gateways. We compare *FDLoRa* with two benchmarks: (1) LoRaWAN, which transmits one downlink packet through a single gateway, and (2) Oracle, which explicitly receives all the downlink

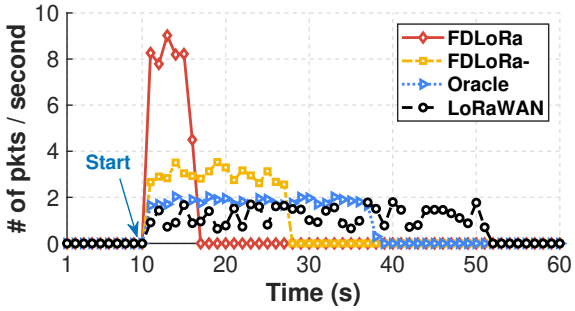


Figure 5.14: Practical application with bursty downlink traffic.

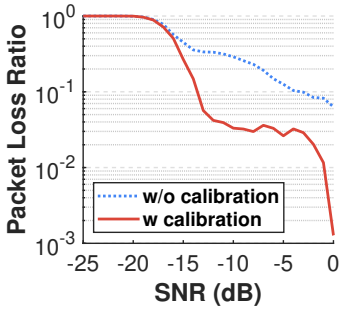


Figure 5.15: Downlink SFD interference.

logical channels.

We measure the maximum number of concurrent downlink transmissions and present the results in Figure 5.13(d). The maximum concurrency of LoRaWAN is limited to 1 due to the resource conflict of half-duplex gateways, which restricts the allocation of additional gateways for downlink transmissions. However, the number of available downlink logical channels (*i.e.*, Oracle) continues to increase. *FDLoRa* closely aligns with Oracle, *i.e.*, within 86%. Specifically, when the number of available gateways increases to 8, *FDLoRa* supports 41 concurrent downlink transmissions. The results indicate that *FDLoRa* can effectively utilize all available gateways to enhance downlink transmission.

Practical application with bursty downlink traffic. This experiment evaluates *FDLoRa* within a smart lamppost system (37) that experiences bursty downlink traffic. The objective is to switch on/off 50 lampposts in the least amount of time. We deploy 50 LoRa nodes for device control and transmit commands via three gateways. We compare *FDLoRa* against two benchmarks: (1) LoRaWAN: COTS gateways with a LoRaWAN network server. (2) Oracle: LoRaWAN with an Oracle scheduler that explicitly schedules downlinks to achieve optimal time slot utilization.

We record the downlink packet rate obtained in each time slot and display the results in Figure 5.14. The transmission starts at the 10-second mark. We see

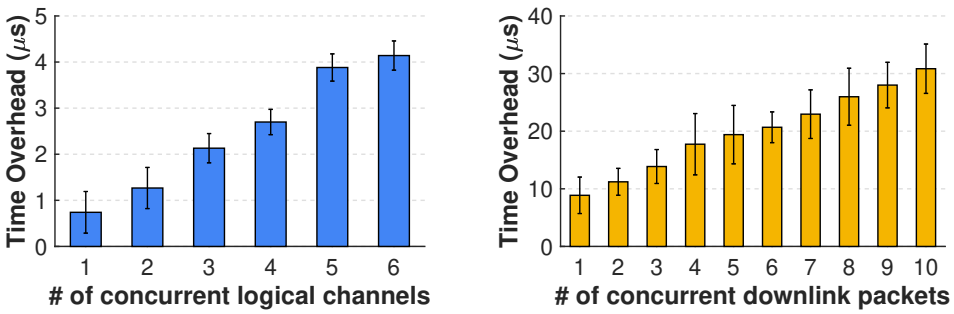
that LoRaWAN requires >40 seconds to complete the task. Even when enhanced with an Oracle scheduler for optimal time slot utilization, LoRaWAN still requires >27 seconds, with a limited downlink packet rate of <1.9 packets/s. In contrast, *FDLoRa-* improves the rate to 3 packets/s by enabling concurrent logical channels through a single gateway. The full-duplex communication of *FDLoRa* further boosts the rate to approximately 8 packets/s by unlocking multiple gateways for concurrent transmission, which is 5.7× higher than Oracle LoRaWAN. *FDLoRa-* and *FDLoRa* take <17 and <6 seconds to accomplish this task, respectively.

5.6.3 Microbenchmarks

Impact of downlink SFD interference. This experiment evaluates the impact of downlink SFD on weak uplink reception. Specifically, we deploy a LoRa node at different SNR conditions and measure the packet loss ratio of *FDLoRa* without downlink SFD calibration. The packets are considered lost when the symbol error rate is more than 20%. We also implement a full version of *FDLoRa* for comparison.

Figure 5.15 shows the packet loss ratio under different SNR conditions. We see that the downlink SFD (including 2.25 interference chirps) leads to about 8 % more packet loss. The reason is that the strong SFD interference introduces more symbol errors for weak uplink reception. *FDLoRa* effectively mitigates this issue by calibrating downlink SFD interference during demodulation.

Time overhead. This experiment investigates the time overhead of *FDLoRa* for logical channel power allocation and dispatching concurrent downlink packets. We first test the power allocation algorithm for concurrent logical channels with a time complexity of $O(NA)$. This algorithm is executed on a laptop. We measure the runtime under different number of concurrent logical channels, specifically for $N = 1 \sim 6$. As shown in Figure 5.16(a), the experimental overhead observed is slightly higher than linear $O(N)$. Next, we evaluate the network-level downlink



(a) Power allocation (b) Dispatch concurrent downlink

Figure 5.16: Time overhead of downlink framework.

dispatcher. We also measure the runtime on a laptop under different number of concurrent downlink packets. The results demonstrate a linear $O(N)$ overhead as depicted in Figure 5.16(b). Notably, the overheads of the two algorithms under normal downlink traffic are below 5 and 40 μs, respectively. This duration is much shorter than the airtime of LoRa packets (*e.g.*, hundreds of ms) and can be easily handled by the gateway/server processors.

5.7 Discussion

Table 5.1 contrasts *FDLoRa* with the state-of-the-art.

Full-duplex radio. Self-interference cancellation for full-duplex radios employs a combination of antenna isolation, analog cancellation, and digital cancellation (145; 156; 164; 155; 165; 166; 167). Many studies (145; 167) focused on full-duplex communication for WiFi systems. Regarding full-duplex LoRa gateways, BSMA (146) has developed analog cancellation circuits and a bin suppression technique to enable full-duplex downlink busy signal transmission for channel access control. However, the transmit signal in (146) is a pre-defined busy signal consisting of only base up-chirps. Its digital domain solution faces limitations when applied to standard downlink packets (*e.g.*, data transfer, ACK). To support full-duplex uplink and downlink communications, *FDLoRa* borrows from the analog cancellation circuits (146)

System Name	Full Duplex	# of gateway	Concur. per gateway	Method.
LoRaWAN (27)	✗	1	1	Single logical ch. & single gateway
HyLink (163)	✗	1	1	Parallel-CSS
MaLoRaGW (162)	✗	1	2	MU-MIMO (2 Tx)
BSMA (146)	Busy Signal	1	1	Media access control (for uplink)
FDLoRa	Uplink & Downlink	All	**5.16**	Full-duplex comm. & Multi. logical ch. & Multi. gateways

Table 5.1: Comparisons between *FDLoRa* and SoTA.

and introduces innovative digital SIC methods that outperform existing approaches.

LoRa downlink. The newly emerging IoT applications are calling for robust LoRa downlink to support efficient device management, real-time response, and high throughput data transfer. LoRaWAN (27) sequentially transmits the downlink packets and thus suffers from significant downlink-uplink asymmetry. A recent work (153) proposes a downlink gateway selection strategy but is still restricted by the sequential transmit methodology. HyLink (163) enhances link throughput by modulating parallel chirps but can only serve a single client. MaLoRaGW (162) requires a multi-antenna gateway to support concurrent downlink via beamforming. However, existing works either limit to one client (163) or narrowly focus on concurrency within a single logical channel (162). Built on top of the full-duplex gateway, *FDLoRa* unlocks the available spectrum and resources for downlink transmission, enabling concurrent downlink transmission through multiple logical channels per gateway and across more gateways.

5.8 Limitations and Future Work

Overheads of *FDLoRa*. The key advantage of *FDLoRa* is that it fully unlocks the spectrum and gateway utilization for downlink with full-duplex gateways. By leveraging uplink-downlink orthogonality, *FDLoRa* implements a novel digital SIC method that outperforms prior works. Importantly, the (de)modulation of down-

link packets based on down-chirp is supported by COTS LoRa nodes (82; 161) and gateways (144), allowing *FDLoRa* to operate with COTS LoRa end devices without requiring any modifications. Despite these benefits, *FDLoRa* introduces higher computational and energy overheads at the gateway and server sides, associated with additional analog circuits, digital SIC methods, and the management of downlink packets and logical channels. A comprehensive evaluation of these overheads will be carried out in future work. Additionally, *FDLoRa* experiences slight SNR losses ($\sim 1.1\,\mathrm{dB}$) as a result of full-duplex communication. Nonetheless, we believe these trade-offs are justified by the significant improvements in downlink scalability and overall system performance.

Achieving true downlink-uplink symmetry. The goal of *FDLoRa* is to address the downlink-uplink asymmetry in LoRaWAN. *FDLoRa* mitigates this gap by scaling downlink communications from one to multiple logical channels. However, the gap is not entirely closed due to the hardware constraints of commodity LoRa gateways, which utilize 9 Rx chains for uplink but only one Tx chain for downlink. This physical asymmetry presents fundamental challenges for achieving true downlink-uplink symmetry in LoRaWAN. We will explore additional possibilities to enhance downlink communications for newly emerging IoT applications using LoRa.

Further enhancing downlink communication. Building on *FDLoRa*, our future work will focus on developing software-based gateway solutions and new downlink protocols. We will also conduct more comprehensive evaluations of *FDLoRa*, including fine-grained and quantitative system overhead analysis, real-time performance enhancements, multi-gateway deployment issues, *etc.*.

5.9 Conclusion

This chapter presents *FDLoRa*, a collaborative design of LoRa gateway and downlink framework that revolutionizes the sequential downlink paradigm in existing LoRaWAN. Unlike traditional LoRaWAN that utilizes half-duplex gateways, *FDLoRa* presents a new in-band full-duplex gateway design. To harness the capabilities of full-duplex gateways, *FDLoRa* develops a novel downlink framework that scales concurrent downlink packets to massive logical channels of all available gateways. This breakthrough addresses the downlink-uplink asymmetry in current LoRaWAN and accommodates the emerging demands of new applications on increasing downlink transmissions.

Chapter 6

Conclusions

This dissertation systematically addresses critical challenges in deploying a scalable LoRa network for massive edge IoT devices. By developing practical techniques for scaling flexible network capacity, resolving inter-logical channel interference in large-scale deployments, and enabling concurrent downlink communication for diverse IoT applications, my work significantly enhances the scalability and efficiency of LoRa infrastructure and corresponding IoT applications. Unlike prior solutions that introduce new modulation schemes for LoRa (71; 140; 163), our approach fully exploits the logical channels already defined by the LoRaWAN standard and natively supported by all COTS LoRa devices. This design ensures full compliance with existing end-node standards, requiring no changes on the device side. However, achieving scalability gains does necessitate upgrading COTS LoRa gateways. Looking ahead, we plan to modularize the key components of our solution by clearly specifying inputs, outputs, required modifications, and resource utilization, thereby accelerating standardization and driving industrial adoption.

To advance the scalability and utility of LoRa networks, future research must extend beyond gateway-centric enhancements and address full-stack challenges encountered in real-world deployments. First, with the growing adoption of LoRa in emerging domains such as satellite IoT communications (106), particularly in Low

Earth Orbit (LEO) scenarios, new application-specific challenges arise—most notably, the Doppler frequency shifts induced by satellite motion. Addressing these dynamics necessitates rethinking LoRa network design to accommodate non-terrestrial and mobile environments, moving beyond the assumptions of static, ground-based deployments. Second, the evolution of LoRa systems calls for more adaptive and tightly integrated architectures capable of coordinating with local base stations and interacting efficiently with network servers to optimize performance. Recent advances in gateway collaboration (168) suggest a promising shift toward cloud-centric architectures; however, such designs often impose substantial backhaul bandwidth overheads. As part of future work, I aim to explore a more deployable, hierarchical edge-cloud coordination framework that balances scalability with practical deployment constraints. These emerging applications and architectural shifts may pave the way for a new generation of LoRa systems that are not only technically robust but also scalable and adaptable to complex, heterogeneous environments.

References

[1] Semtech. (2022, Oct.) Sx1301. "https://www.semtech.com/products\/wireless-rf/lora-core/sx1301".

[2] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, "Concurrent Interference Cancellation: Decoding Multi-Packet Collisions in LoRa," in *ACM SIGCOMM 2021*, 2021, pp. 503–515.

[3] F. Yu, X. Zheng, L. Liu, and H. Ma, "Enabling concurrency for non-orthogonal lora channels," in *ACM MobiCom 2023*, 2023, pp. 1–15.

[4] S. Yu, X. Xia, N. Hou, Y. Zheng, and T. Gu, "Revolutionizing lora gateway with xgate: Scalable concurrent transmission across massive logical channels," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 482–496.

[5] R. Zhao, F. Zhu, Y. Feng, S. Peng, X. Tian, H. Yu, and X. Wang, "Ofdma-enabled wi-fi backscatter," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–15.

[6] M. Dunna, M. Meng, P.-H. Wang, C. Zhang, P. P. Mercier, and D. Bharadia, "Syncscatter: Enabling wifi like synchronization and range for wifi backscatter communication." in *NSDI*, 2021, pp. 923–937.

[7] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for cots rfid tags," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 500–511.

[8] L. Kong, L. He, Y. Gu, M.-Y. Wu, and T. He, "A parallel identification protocol for rfid systems," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 154–162.

[9] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "Lora backscatter: Enabling the vision of ubiquitous connectivity," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 3, pp. 1–24, 2017.

[10] M. Katanbaf, A. Weinand, and V. Talla, "Simplifying backscatter deployment: Full-duplex lora backscatter." in *NSDI*, 2021, pp. 955–972.

[11] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, "Saiyan: Design and implementation of a low-power demodulator for lora backscatter systems," in *Proc. USENIX NSDI*, 2022, pp. 437–451.

[12] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "Plora: A passive long-range data network from ambient lora transmissions," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 147–160.

[13] J. Jiang, Z. Xu, F. Dang, and J. Wang, "Long-range ambient lora backscatter with parallel decoding," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 684–696.

[14] M. Hessar, A. Najafi *et al.*, "Netscatter: Enabling large-scale backscatter networks," in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI'19)*, 2019.

[15] Y. Song, L. Lu, J. Wang, C. Zhang, H. Zheng, S. Yang, J. Han, and J. Li, "μmote: Enabling passive chirp de-spreading and μw-level long-range downlink for backscatter devices," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, 2023.

[16] S. Li, H. Zheng, C. Zhang, Y. Song, S. Yang, M. Chen, L. Lu, and M. Li, "Passive {DSSS}: Empowering the downlink communication for backscatter systems," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 913–928.

[17] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, "Fm backscatter: Enabling connected cities and smart fabrics." in *NSDI*, vol. 17, no. 10.5555, 2017, pp. 3 154 630–3 154 650.

[18] X. Li, C. Feng, X. Wang, Y. Zhang, Y. Xie, and X. Chen, "{RF-Bouncer}: A programmable dual-band metasurface for sub-6 wireless networks," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 389–404.

[19] Y. Sangar and B. Krishnaswamy, "Wichronos: energy-efficient modulation for long-range, large-scale wireless networks," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[20] X. Xia, Y. Zheng, and T. Gu, "Litenap: Downclocking lora reception," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2632–2645, 2021.

[21] H. Yang, H. Liu, C. Luo, Y. Wu, W. Li, A. Y. Zomaya, L. Song, and W. Xu, "Vehicle-key: A secret key establishment scheme for lora-enabled iov communications," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 787–797.

[22] J.-L. Wang, S. Tong, X.-Y. Li, Z. Yang, F. Xiao, and Y.-H. Liu, "On the scalability of internet of things systems," *Journal of Computer Science and Technology*, pp. 1–13, 2025.

[23] D. Mu, Y. Chen, J. Shi, and M. Sha, "Runtime control of lora spreading factor for campus shuttle monitoring," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, 2020, pp. 1–11.

[24] Y. Li, J. Yang, and J. Wang, "Dylora: Towards energy efficient dynamic lora transmission control," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2312–2320.

[25] Y. Cheng, H. Saputra, L. M. Goh, and Y. Wu, "Secure smart metering based on lora technology," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, 2018, pp. 1–8.

[26] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A survey on lora for smart agriculture: Current trends and future perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2022.

[27] L. Alliance. (2022, Oct.) Lorawan specification. "https://lora-alliance.org/about-lorawan".

[28] V. M. Research. (2025) Lora and lorawan iot market size and forecast. "https://www.verifiedmarketresearch.com/product/lora-and-lorawan-iot-market/".

[29] S. Tong, J. Wang, J. Yang, Y. Liu, and J. Zhang, "Citywide lora network deployment and operation: Measurements, analysis, and implications," in *ACM SenSys 2023*, 2023, pp. 362–375.

[30] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *ACM SIGCOMM 2017*, 2017, pp. 309–321.

[31] X. Xia, Y. Zheng, and T. Gu, "FTrack: parallel decoding for LoRa transmissions," in *ACM SenSys 2019*, 2019, pp. 192–204.

[32] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in LPWANs," in *ACM MobiSys 2020*. Toronto Ontario Canada: ACM, Jun. 2020, pp. 234–246.

[33] X. Xia, N. Hou, Y. Zheng, and T. Gu, "Pcube: scaling lora concurrent transmissions with reception diversities," in *ACM MobiCom 2021*. Association for Computing Machinery, 2021, pp. 670–683.

[34] M. Mishra, D. Koch, M. O. Shahid, B. Krishnaswamy, K. Chintalapudi, and S. Banerjee, "{OpenLoRa}: Validating {LoRa} implementations through an extensible and open-sourced framework," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 1165–1183.

[35] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *ACM SIGCOMM 2008*, 2008, pp. 159–170.

[36] Y. Chen, K. Yang, Z. An, B. Holder, L. Paloutzian, K. Bali, and W. Du, "Marlp: Time-series forecasting control for agricultural managed aquifer recharge," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'24)*, 2024.

[37] A. Gehlot, S. S. Alshamrani, R. Singh, M. Rashid, S. V. Akram, A. S. AlGhamdi, and F. R. Albogamy, "Internet of things and long-range-based smart lampposts for illuminating smart cities," *Sustainability*, vol. 13, no. 11, p. 6398, 2021.

[38] N. Sornin, "Lorawan®: Firmware updates over-the-air," *Semtech Technical Paper*, vol. 200, pp. 93 012–8790, 2020.

[39] Z. Sun, T. Ni, H. Yang, K. Liu, Y. Zhang, T. Gu, and W. Xu, "Flora: Energy-efficient, reliable, and beamforming-assisted over-the-air firmware update in lora networks," in *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, 2023, pp. 14–26.

[40] W. S. Jeon and D. G. Jeong, "Adaptive uplink rate control for confirmed class a transmission in lora networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 361–10 374, 2020.

[41] 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)*. IEEE, 2019, pp. 1–7.

[42] Semtech. (2024, Mar.) Understanding adr. "https://lora-developers.semtech.com/documentation/tech-papers-and-guides/understanding-adr".

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

[44] A. W. Services. (2024, Mar.) Queue downlink messages to send to lorawan devices. "https://docs.aws.amazon.com/iot/latest/developerguide/connect-iot-lorawan-downlink-queue.html".

[45] S. Yu, X. Xia, N. Hou, Y. Zheng, and T. Gu, "Xgate: Scaling lora communications to massive logical channels," *IEEE Transactions on Networking*, 2025.

[46] S. Yu, Z. Zhang, X. Xia, Y. Zheng, and J. Wang, "Are lora logical channels really orthogonal? practically orthogonalizing massive logical channels," in *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*, 2025, pp. 56–69.

[47] ——, "Resolving inter-logical channel interference for large-scale lora deployments," *IEEE Transactions on Mobile Computing*, 2025.

[48] S. Yu, X. Xia, Z. Zhang, N. Hou, and Y. Zheng, "Fdlora: Tackling downlink-uplink asymmetry with full-duplex lora gateways," in *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, 2024, pp. 281–294.

[49] ——, "Fdlora: Scaling downlink concurrent transmissions with full-duplex lora gateways," *IEEE Transactions on Mobile Computing*, 2025.

[50] A. Gadre, R. Narayanan, A. Luong, A. Rowe, B. Iannucci, and S. Kumar, "Frequency configuration for low-power wide-area networks in a heartbeat." in *USENIX NSDI*, 2020, pp. 339–352.

[51] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 56–68.

[52] J. Du, Y. Ren, Z. Zhu, C. Li, Z. Cao, Q. Ma, and Y. Liu, "Srlora: Neural-enhanced lora weak signal decoding with multi-gateway super resolution," in *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2023, pp. 270–279.

[53] C. Li, H. Guo, S. Tong, Z. Cao, M. Zhang, Q. Yang, L. Xiao, J. Wang, and Y. Liu, "Nelora: Neural-enhanced demodulation for low-power wans," *GetMobile: Mobile Computing and Communications*, vol. 26, no. 3, pp. 34–38, 2022.

[54] S. Tong, Y. He, Y. Liu, and J. Wang, "De-spreading over the air: long-range ctc for diverse receivers with lora," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 42–54.

[55] R. Trüb, R. Da Forno, A. Biri, J. Beutel, and L. Thiele, "Lsr: Energy-efficient multi-modulation communication for inhomogeneous wireless iot networks," *ACM Transactions on Internet of Things*, vol. 4, no. 2, pp. 1–36, 2023.

[56] C. Shao and O. Muta, "Tonari: Reactive detection of close physical contact using unlicensed lpwan signals," *ACM Transactions on Internet of Things*, vol. 5, no. 2, pp. 1–30, 2024.

[57] N. Hou, Y. Wang, X. Xia, S. Yu, Y. Zheng, and T. Gu, "Molora: Intelligent mobile antenna system for enhanced lora reception in urban environments," in *ACM SenSys*, 2025.

[58] Q. Chen, X. Xia, Z. Li, and Y. Zheng, "Collaborative transmission over intermediate links in duty-cycle wsns," in *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2021, pp. 843–850.

[59] Q. Meng, H. Wang, C. Zhang, Y. Song, S. Li, L. Lu, and H. Zhu, "Processor-sharing internet of things architecture for large-scale deployment," in *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, 2024, pp. 211–224.

[60] H. Wang, Y. Song, Q. Meng, Z. Gao, C. Zhang, and L. Lu, "Sisyphus: Redefining low power for lora receiver," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 1177–1191.

[61] Y. Ren, G. Li, Y. Liu, Y. Dong, and Z. Cao, "Aeroecho: Towards agricultural low-power wide-area backscatter with aerial excitation source," in *IEEE INFOCOM*, 2025.

[62] J. Liu, W. Xu, S. Jha, and W. Hu, "Nephalai: towards lpwan c-ran with physical layer compression," in *ACM MobiCom 2020*, 2020, pp. 1–12.

[63] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *ACM/IEEE IPSN 2018*. IEEE, 2018, pp. 60–71.

[64] N. Hou, X. Xia, and Y. Zheng, "Don't miss weak packets: Boosting lora reception with antenna diversities," *ACM Trans. Sen. Netw.*, vol. 19, no. 2, feb 2023.

[65] S. Tong, Z. Shen, Y. Liu, and J. Wang, "Combating link dynamics for reliable lora connection in urban settings," in *ACM MobiCom 2021*, New York, NY, USA, 2021, p. 642–655.

[66] X. Xia, Q. Chen, N. Hou, Y. Zheng, and M. Li, "Xcopy: Boosting weak links for reliable lora communication," in *ACM MobiCom 2023*. Association for Computing Machinery, 2023, pp. 1–15.

[67] S. Tong, Z. Xu, and J. Wang, "CoLoRa: Enabling Multi-Packet Reception in LoRa," in *IEEE INFOCOM 2020*, Jul. 2020, pp. 2303–2311, iSSN: 2641-9874.

[68] X. Wang, L. Kong, L. He, and G. Chen, "mlora: A multi-packet reception protocol in lora networks," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–11.

[69] P. Xie, Y. Li, Z. Xu, Q. Chen, Y. Liu, and J. Wang, "Push the limit of lpwans with concurrent transmissions," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[70] C. Li, Z. Cao, and L. Xiao, "Curvealoha: Non-linear chirps enabled high throughput random channel access for lora," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 520–529.

[71] Z. Xu, S. Tong, P. Xie, and J. Wang, "Fliplora: Resolving collisions with up-down quasi-orthogonality," in *IEEE SECON*, 2020, pp. 1–9.

[72] Z. Xu, P. Xie, and J. Wang, "Pyramid: Real-time lora collision decoding with peak tracking," in *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–9.

[73] C. Li, X. Guo, L. Shangguan, Z. Cao, and K. Jamieson, "CurvingLoRa to boost LoRa network throughput via concurrent transmission," in *USENIX NSDI 2022*. Renton, WA: USENIX Association, Apr. 2022, pp. 879–895.

[74] T. Xu and M. 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)*. IEEE, 2020, pp. 1415–1420.

[75] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.

[76] J. Chan, A. Wang, A. Krishnamurthy, and S. Gollakota, "Deepsense: Enabling carrier sense in low-power wide area networks using deep learning," *arXiv preprint arXiv:1904.10607*, 2019.

[77] A. Gamage, J. Liando, C. Gu, R. Tan, M. Li, and O. Seller, "Lmac: Efficient carrier-sense multiple access for lora," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–27, 2023.

[78] S. Suji Prasad, M. Thangatamilan, M. Suresh, H. Panchal, C. A. Rajan, C. Sagana, B. Gunapriya, A. Sharma, T. Panchal, and K. K. Sadasivuni, "An efficient lora-based smart agriculture management and monitoring system using wireless sensor networks," *International Journal of Ambient Energy*, vol. 43, no. 1, pp. 5447–5450, 2022.

[79] P. Kumari, R. Mishra, H. P. Gupta, T. Dutta, and S. K. Das, "An energy efficient smart metering system using edge computing in lora network," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 4, pp. 786–798, 2021.

[80] Semtech. (2022, Dec) New abi research white paper. "https://www.semtech.com/company/press/new-abi-research-white-paper-highlights-growth-of-lora-and-the-lorawan-open-protocol".

[81] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, pp. 1–35, 2019.

[82] Semtech. (2022, Dec.) Sx1276. "https://www.semtech.com/products\/wireless-rf/lora-connect/sx1276".

[83] ——. (2023, Jan.) Lora gateway chips and reference designs. "https://www.semtech.com/products/wireless-rf/lora-core".

[84] F. Yu, X. Zheng, L. Liu, and H. Ma, "Loradar: An efficient lora channel occupancy acquirer based on cross-channel scanning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 540–549.

[85] N. Hou, X. Xia, and Y. Zheng, "Jamming of lora phy and countermeasure," in *IEEE INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[86] Gr-LoRa GitHub community. (2021, Jul) gr-lora projects. "https://github.com/rpp0/gr-lora".

[87] A. Gadre, F. Yi, A. Rowe, B. Iannucci, and S. Kumar, "Quick (and dirty) aggregate queries on low-power wans," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 277–288.

[88] L. Shen, Q. Yang, Y. Zheng, and M. Li, "Autoiot: Llm-driven automated natural language programming for aiot applications," in *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, 2025.

[89] R. Li, Z. Zhang, X. Xia, N. Hou, W. Chai, S. Yu, Y. Zheng, and T. Gu, "From interference mitigation to toleration: Pathway to practical spatial reuse in lpwans," in *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, 2025.

[90] A. Bansal, A. Gadre, V. Singh, A. Rowe, B. Iannucci, and S. Kumar, "Owll: Accurate lora localization using the tv whitespaces," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*, 2021, pp. 148–162.

[91] K. Yang and W. Du, "A low-density parity-check coding scheme for lora networking," *ACM Transactions on Sensor Networks*, 2024.

[92] A. Gadre, R. Narayanan, and S. Kumar, "Maintaining uav stability using low-power wans," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 738–740.

[93] Y. Ren, P. Cai, J. Jiang, J. Du, and Z. Cao, "Prism: High-throughput lora backscatter with non-linear chirps," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[94] J. Jiang, J. Wang, Y. Liu, Y. Chen, and Y. Liu, "Wicloak: Protect location privacy of wifi devices," in *ACM/IEEE IPSN*, 2024, pp. 101–112.

[95] J. Jiang, J. Wang, Y. Chen, S. Tong, P. Xie, Y. Liu, and Y. Liu, "Willow: Practical wifi backscatter localization with parallel tags," in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024, pp. 265–277.

[96] N. Hou, X. Xia, and Y. Zheng, "Cloaklora: A covert channel over lora phy," *IEEE/ACM Transactions on Networking*, vol. 31, no. 3, pp. 1159–1172, 2022.

[97] N. Hou, X. Xia, Y. Wang, and Y. Zheng, "One shot for all: Quick and accurate data aggregation for lpwans," *IEEE/ACM Transactions on Networking*, 2024.

[98] H. Yang, Z. Sun, H. Liu, X. Xia, Y. Zhang, T. Gu, G. Hancke, and W. Xu, "Chirpkey: a chirp-level information-based key generation scheme for lora networks via perturbed compressed sensing," in *IEEE INFOCOM*. IEEE, 2023, pp. 1–10.

[99] L. Shen, Q. Yang, K. Cui, Y. Zheng, X.-Y. Wei, J. Liu, and J. Han, "Fedconv: A learning-on-model paradigm for heterogeneous federated clients," in *ACM MobiSys*, 2024, pp. 398–411.

[100] J. Cao, J. Chen, C. Lin, Y. Liu, K. Wang, and Z. Li, "Practical gaze tracking on any surface with your phone," *IEEE Transactions on Mobile Computing*, 2024.

[101] L. Shen, Q. Yang, X. Huang, Z. Ma, and Y. Zheng, "Gpiot: Tailoring small language models for iot program synthesis and development," in *ACM SenSys*, 2025.

[102] Y. Qu, Z. Wei, Z. Qin, T. Wu, J. Ma, H. Dai, and C. Dong, "Collaborative service provisioning for uav-assisted mobile edge computing," *Chinese Journal of Electronics*, vol. 33, no. 6, pp. 1504–1514, 2024.

[103] X. Huang, L. Shen, Z. Ma, and Y. Zheng, "Towards privacy-preserving and personalized smart homes via tailored small language models," *arXiv preprint arXiv:2507.08878*, 2025.

[104] K. Yang, Y. Chen, and W. Du, "Orchloc: In-orchard localization via a single lora gateway and generative diffusion model-based fingerprinting," in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024, pp. 304–317.

[105] Y. Ren, W. Sun, J. Du, H. Zeng, Y. Dong, M. Zhang, S. Chen, Y. Liu, T. Li, and Z. Cao, "Demeter: Reliable cross-soil lpwan with low-cost signal polarization alignment," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 230–245.

[106] Y. Ren, A. Gamage, L. Liu, M. Li, S. Chen, Y. Dong, and Z. Cao, "Sateriot: High-performance ground-space networking for rural iot," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 755–769.

[107] A. Gadre, Z. Machester, and S. Kumar, "Adapting lora ground stations for low-latency imaging and inference from lora-enabled cubesats," *ACM Transactions on Sensor Networks*, vol. 20, no. 5, pp. 1–30, 2024.

[108] Y. Chen, J. Hu, J. Zhao, and G. Min, "Qos-aware computation offloading in leo satellite edge computing for iot: A game-theoretical approach," *Chinese Journal of Electronics*, vol. 33, no. 4, pp. 875–885, 2024.

[109] Semtech. (2024, Nov.) Lora. "https://www.semtech.com/lora".

[110] F. Yu, X. Zheng, Y. Ma, L. Liu, and H. Ma, "Resolve cross-channel interference for lora," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2024, pp. 1027–1038.

[111] C. Li and Z. Cao, "Lora networking techniques for large-scale and long-term iot: A down-to-top survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–36, 2022.

[112] Y. Ren, L. Liu, C. Li, Z. Cao, and S. Chen, "Is lorawan really wide? fine-grained lora link-level measurement in an urban environment," in *2022 IEEE 30th International Conference on Network Protocols (ICNP)*. IEEE, 2022, pp. 1–12.

[113] K. Yang, Y. Chen, X. Chen, and W. Du, "Link quality modeling for lora networks in orchards," in *IPSN 2023*, 2023, pp. 27–39.

[114] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on lora networking: Research problems, current solutions, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2019.

[115] M. Yang, J. Zhang, A. Gadre, Z. Liu, S. Kumar, and V. Sekar, "Joltik: enabling energy-efficient" future-proof" analytics on low-power wide-area networks," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[116] F. Yu, X. Zheng, L. Liu, and H. Ma, "LoRadar: An Efficient LoRa Channel Occupancy Acquirer based on Cross-channel Scanning," in *IEEE INFOCOM 2022*. London, United Kingdom: IEEE, May 2022, pp. 540–549.

[117] R. Li, X. Zheng, Y. Wang, L. Liu, and H. Ma, "Polarscheduler: Dynamic transmission control for floating lora networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 550–559.

[118] Y. Wang, X. Zheng, L. Liu, and H. Ma, "Polartracker: Attitude-aware channel access for floating low power wide area networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1807–1821, 2022.

[119] J. Luo, Z. Xu, J. Lin, C. Chen, and R. Xiong, "Ch-mac: Achieving low-latency reliable communication via coding and hopping in lpwan," *ACM Transactions on Internet of Things*, vol. 4, no. 4, pp. 1–25, 2023.

[120] A.-U.-H. Ahmar, E. Aras, T. D. Nguyen, S. Michiels, W. Joosen, and D. Hughes, "Design of a robust mac protocol for lora," *ACM Transactions on Internet of Things*, vol. 4, no. 1, pp. 1–25, 2023.

[121] J. Wang, S. Tong, Z. Xu, and P. Xie, "Real-time concurrent lora transmissions based on peak tracking," *IEEE Transactions on Mobile Computing*, 2024.

[122] G. Deepthy and R. Susan, "Analysis of successive interference cancellation in cdma systems," in *2012 Second International Conference on Advanced Computing & Communication Technologies*. IEEE, 2012, pp. 481–485.

[123] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: Carving out mac layer opportunities," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 346–357, 2012.

[124] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a ds/cdma system," *IEEE journal on selected areas in communications*, vol. 12, no. 5, pp. 796–807, 1994.

[125] J. G. Andrews and T. H.-Y. Meng, "Performance of multicarrier cdma with successive interference cancellation in a multipath fading channel," *IEEE Transactions on Communications*, vol. 52, no. 5, pp. 811–822, 2004.

[126] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.

[127] Z. Xu, S. Tong, P. Xie, and J. Wang, "From demodulation to decoding: Toward complete lora phy understanding and implementation," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–27, 2023.

[128] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using spectrogram and cnn," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[129] X. Wang, L. Kong, Z. Wu, L. Cheng, C. Xu, and G. Chen, "Slora: Towards secure lora communications with fine-grained physical layer features," in *ACM SenSys 2020*, 2020, pp. 258–270.

[130] J. Du, Y. Liu, Y. Ren, L. Liu, and Z. Cao, "Loratrimmer: Optimal energy condensation with chirp trimming for lora weak signal decoding," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024.

[131] K. Sabharwal, S. Ramesh, J. Wang, D. M. Divakaran, and M. C. Chan, "Enhancing lora reception with generative models: Channel-aware denoising of loraphy signals," in *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, 2024, pp. 507–520.

[132] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Aloba: rethinking on-off keying modulation for ambient lora backscatter," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 192–204.

[133] J. Jiang, J. Wang, Y. Chen, Y. Liu, and Y. Liu, "Locra: Enable practical long-range backscatter localization for low-cost tags," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, 2023, pp. 317–329.

[134] L. Liu, Y. Yao, Z. Cao, and M. Zhang, "Deeplora: Learning accurate path loss model for long distance links in lpwan," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[135] K. Yang and W. Du, "Lldpc: A low-density parity-check coding scheme for lora networks," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 193–206.

[136] Z. Zhao, W. Gao, W. Du, G. Min, W. Mao, and M. Singhal, "Towards energy-fairness in lora networks," *IEEE Transactions on Mobile Computing*, 2022.

[137] Z. Wang, L. Kong, L. Shangguan, L. He, K. Xu, Y. Cao, H. Yu, Q. Xiang, J. Yu, T. Ma *et al.*, "Ligbee: Symbol-level cross-technology communication from lora to zigbee," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[138] C. Gu, L. Jiang, R. Tan, M. Li, and J. Huang, "Attack-aware synchronization-free data timestamping in lorawan," *ACM Transactions on Sensor Networks (TOSN)*, vol. 18, no. 1, pp. 1–31, 2021.

[139] D. Guo, C. Gu, L. Jiang, W. Luo, and R. Tan, "Illoc: In-hall localization with standard lorawan uplink frames," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–26, 2022.

[140] C. Li, Y. Ren, S. Tong, S. I. Siam, M. Zhang, J. Wang, Y. Liu, and Z. Cao, "Chirptransformer: Versatile lora encoding for low-power wide-area iot," in *ACM MobiSys 2024*, 2024, pp. 479–491.

[141] K. Cui, Q. Yang, L. Shen, Y. Zheng, F. Xiao, and J. Han, "Towards isac-empowered mmwave radars by capturing modulated vibrations," *IEEE Transactions on Mobile Computing*, 2024.

[142] Q. Yang and Y. Zheng, "Aquahelper: Underwater sos transmission and detection in swimming pools," in *ACM SenSys*, 2023, pp. 294–307.

[143] D. Technology. (2024, Mar.) lps8n. "https://www.dragino.com/products/lora-lorawan-gateway/item/200-lps8n.html".

[144] R. Technology. (2024, Mar.) Rak7268. "https://docs.rakwireless.com/Product-Categories/WisGate/RAK7268/Overview".

[145] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 375–386.

[146] R. Subbaraman, Y. Guntupalli, S. Jain, R. Kumar, K. Chintalapudi, and D. Bharadia, "Bsma: scalable lora networks using full duplex gateways," in *Proceedings of the 28th Annual International Conference on Mobile*

Computing And Networking, ser. MobiCom '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 676–689. [Online]. Available: https://doi.org/10.1145/3495243.3560544

[147] Z. Sun, H. Yang, K. Liu, Z. Yin, Z. Li, and W. Xu, "Recent advances in lora: A comprehensive survey," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–44, 2022.

[148] Z. Xu, S. Tong, P. Xie, and J. Wang, "From demodulation to decoding: Toward complete lora phy understanding and implementation," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–27, 2023.

[149] Semtech. (2024, Mar.) Understanding lorawan architecture. "https://blog.semtech.com/understanding-the-lorawan-architecture".

[150] T. T. Industries. (2024, Mar.) Duty cycle. "https://www.thethingsnetwork.org/docs/lorawan/duty-cycle".

[151] L. Alliance. (2021) Rp2-1.0.3 lorawan regional parameters. "https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters".

[152] R. Technology. (2024, May) Rak7285. "https://docs.rakwireless.com/Product-Categories/WisGate/RAK7285/Overview".

[153] S. Abboud, N. El Rachkidy, A. Guitton, and H. Safa, "Gateway selection for downlink communication in lorawan," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[154] R. T. Limited. (2024, Apr) Revolutionizing high-density connectivity with the wisgate edge ultra, a full-duplex gateway for lorawan. "https://news.rakwireless.com/high-density-connectivity-with-the-wisgate-edge-ultra-a-full-duplex-lorawan-gateway/".

[155] T. Chen, M. Baraani Dastjerdi, J. Zhou, H. Krishnaswamy, and G. Zussman, "Wideband full-duplex wireless via frequency-domain equalization: Design and experimentation," in *The 25th annual international conference on mobile computing and networking*, 2019, pp. 1–16.

[156] D. Korpi, J. Tamminen, M. Turunen, T. Huusari, Y.-S. Choi, L. Anttila, S. Talwar, and M. Valkama, "Full-duplex mobile device: Pushing the limits," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 80–87, 2016.

[157] Semtech. (2024, Mar.) Antenna vertical isolation calculator. "https://calc.commscope.com/qvisolation.aspx".

[158] V. Va and R. W. Heath, "Basic relationship between channel coherence time and beamwidth in vehicular channels," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE, 2015, pp. 1–5.

[159] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:28836720

[160] Wikipedia, "Knapsack problem — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=Knapsack%20problem&oldid=1223595758, 2024, [Online; accessed 28-May-2024].

[161] IBM. (2022, Apr) Mcci lorawan lmic library. "https://github.com/mcci-catena/arduino-lmic".

[162] H. Pirayesh, S. Zhang, P. K. Sangdeh, and H. Zeng, "Maloragw: Multi-user mimo transmission for lora," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 179–192.

[163] X. Xia, Q. Chen, N. Hou, and Y. Zheng, "Hylink: Towards high throughput lpwans with lora compatible communication," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 578–591.

[164] J. Tamminen, M. Turunen, D. Korpi, T. Huusari, Y.-S. Choi, S. Talwar, and M. Valkama, "Digitally-controlled rf self-interference canceller for full-duplex radios," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 783–787.

[165] M. Duarte, A. Sabharwal, V. Aggarwal, R. Jana, K. K. Ramakrishnan, C. W. Rice, and N. Shankaranarayanan, "Design and characterization of a full-duplex multiantenna system for wifi networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1160–1177, 2013.

[166] E. Everett, A. Sahai, and A. Sabharwal, "Passive self-interference suppression for full-duplex infrastructure nodes," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 680–694, 2014.

[167] D. Bharadia and S. Katti, "Full duplex {MIMO} radios," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 359–372.

[168] M. O. Shahid, D. Koch, J. Raghuram, B. Krishnaswamy, K. Chintalapudi, and S. Banerjee, "Cloud-LoRa: Enabling cloud radio access LoRa networks using reinforcement learning based Bandwidth-Adaptive compression," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. Santa Clara, CA: USENIX Association, 2024, pp. 1959–1976.