WIRELESS IOT COMMUNICATIONS AND NETWORKING: ENERGY EFFICIENCY, SPECTRAL EFFICIENCY, AND SECURITY

By

Hossein Pirayesh

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2023

ABSTRACT

While interest in Internet of Things (IoT) applications has surged in recent years, the broad diversity in their constraints, such as power consumption, channel bandwidth, link robustness, and packet latency, still challenges state-of-the-art technologies to enable efficient and ubiquitous wireless connectivity for IoT devices in many practical scenarios. In this thesis, we study three sets of primary constraints in developing IoT networks: energy efficiency, spectral efficiency, and physical-layer security.

First, this thesis introduces EE-IoT, an energy-efficient wireless communication scheme for IoT networks. The key enabler of EE-IoT is an asymmetric physical-layer design that allows low-complex and single-carrier IoT devices to communicate with multi-carrier-based wireless local-area network (WLAN) access point (AP) at a very low sampling rate, leading to a significant reduction of IoT devices' hardware complexity and power consumption. This thesis further introduces a practical design, termed WiFi-IoT, to enable a transparent coexistence of IoT devices and legacy Wi-Fi devices. WiFi-IoT design allows a multi-antenna AP to simultaneously serve Wi-Fi and IoT devices.

Second, to improve the spectral efficiency of dense IoT networks, this thesis presents a practical uplink distributed multiple-input multiple-output (MIMO) scheme, termed UD-MIMO, for WLANs that enables concurrent data transmissions from multiple users to multiple APs in the absence of fine-grained inter-node synchronization. UD-MIMO leverages new co-channel interference management and data decoding techniques that allow WLANs' APs to decode concurrent data packets from asynchronous users. This thesis further introduces a first-of-its-kind multi-antenna long-range (LoRa) gateway, termed MaLoRaGW, that enables multi-user MIMO LoRa communications, marking a significant advancement in LoRa network capabilities. MaLoRaGW features a joint design for uplink packet detection and downlink beamforming, enabling it to concurrently

serve multiple LoRa user devices in both uplink and downlink. The key component of MaLoRaGW is a baseband signal design for uplink packet collision recovery, accurate channel estimation, and implicit beamforming.

Third, this thesis proposes two jamming-resilient receiver architectures to secure delay-constrained IoT networks against jamming attacks. The proposed schemes leverage multi-antenna technology and new signal detection methods to suppress jamming signals and decode desired signals. It first introduces JammingBird, a multi-antenna receiver design for vehicular ad hoc networks that can tolerate high-power and in-band constant jamming attacks. JammingBird uses two MIMO-based modules, a jamming-resistant synchronizer and a jamming suppressor module, in its physical-layer design. Jamming-resistant synchronizer employs a spatial projection filter that alleviates the impact of jamming signal and allows JammingBird to conduct packet timing and frequency synchronizations. The jamming suppressor module leverages the spatial degrees of freedom provided by multiple antennas to cancel the jamming signal and recover the signal of interest. This thesis further proposes a MIMO-based receiver design to secure ZigBee communications against constant jamming attacks. The enabler of the proposed scheme is a learning-based jamming mitigation method, which can mitigate unknown interference using an optimized neural network.

This thesis provides details on the system implementation, experimental setup, and performance evaluation of the proposed schemes in real-world environments. It further delves into an in-depth analysis of the lessons learned and highlights the open issues in the design of efficient and secure wireless IoT communications and networks.

To my wife, *Mahsa*, my parents, *Mohsen* and *Mehrangiz*,
and my brother, *Parham*, with my gratitude for their love and support.

ACKNOWLEDGMENTS

I would like to extend my sincere appreciation and gratitude to my advisor, Prof. Huacheng Zeng, for his professional guidance, support, and mentorship throughout my Ph.D. studies. His invaluable insights, patience, and encouragement have been significant in shaping the direction of my research and have helped me to grow as a scholar.

I would also like to express my sincere gratitude to my committee members, Prof. Li Xiao, Prof. Tongtong Li, and Prof. Qiben Yan, for their valuable feedback, constructive criticism, and thoughtful suggestions. Their expertise and insight have been invaluable in shaping the development of my research and have challenged me to think critically and creatively about my work.

I am also deeply grateful to my collaborators, Pedram Kheirkhah Sangdeh, Shichen Zhang, and Adnan Quadri, for their invaluable contributions to my research. Their input and collaboration have broadened my perspective and helped me to approach my work with a more interdisciplinary and holistic perspective.

I would like to extend my acknowledgment to the faculty, staff, and fellow students in the Department of Computer Science and Engineering (CSE) at Michigan State University (MSU) for their support and assistance in the completion of my Ph.D. thesis. The intellectual and academic environment fostered by the CSE department at MSU has provided me with a stimulating and conducive atmosphere for conducting research and pursuing scholarly excellence.

I would also like to acknowledge the contributions of the University of Louisville (UofL) in my Ph.D. journey, where a part of this thesis was completed. The academic environment, research facilities, and diverse community at UofL have enriched my research and expanded my scholarly horizons. I am thankful for the opportunities and resources that UofL has provided, which have contributed to the quality of my thesis.

Finally, yet most importantly, I would like to thank my parents, wife, and brother for their

unwavering love, support, and sacrifices. Your constant belief in my abilities and encouragement have been a driving force behind my pursuit of this Ph.D. degree. I am grateful for your unconditional love and support, which have sustained me during challenging times.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
AP	Access Point
CFO	Carrier Frequency Offset
COTS	Commercial Off The Shelf
CP	Cyclic Prefix
CSI	Channel State Information
CSMA/CA	Carrier-Sense Multiple Access/Collision Avoidance
CSS	Chirp Spread Spectrum
CTO	Chirp Timing Offset
DSSS	Direct Sequence Spread Spectrum
EVM	Error Vector Magnitude
FFT	Fast Fourier Transform
IoT	Internet of Things
JMC	Jamming Mitigation Capability
JSR	Jamming to Signal Ratio
LoRa	Long Range
LoRaWAN	LoRa Wide Area Network
LTF	Long Training Field
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Square Error
MU MIMO	Multi-User Multiple Input Multiple Output

NB	Narrow Band
OFDM	Orthogonal Frequency Division Multiplexing
PER	Packet Error Rate
PHY	Physical
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SDMA	Space Division Multiple Access
SDoF	Spatial Degrees of Freedom
SDR	Software Defined Radio
STF	Short Training Field
SVD	Singular Value Decomposition
TDAM	Time Division Multiple Access
UE	User Equipment
USRP	Universal Software Radio Peripheral
VANET	Vehicular Ad Hoc Network
WLAN	Wireless Local Area Network
ZF	Zero Forcing

Chapter 1

Introduction

The Internet of Things (IoT) is the strategy of extending Internet connectivity beyond standard electronic devices (e.g., desktops, laptops, and smartphones) to any type of traditionally dumb physical devices and everyday objects. With the rapid proliferation of IoT applications under the driving forces of 5G and artificial intelligence, IoT services have penetrated every aspect of our lives. It is expected that by year 2025, the number of IoT devices will reach 21.5 billion and the global IoT market value will achieve $\$7.1$ trillion [1]. This massive number of connected IoT devices will enable a wide variety of applications, as shown in 1.1, such as smart cities [2], smart homes [3], healthcare devices [4], industries [5], and transportation systems [6].

While interest in IoT applications has surged in recent years, their broad diversity in constraints and requirements is challenging the capability of existing technologies to realize IoT networks in many practical scenarios. These constraints can be summarized along three axes: energy consumption, bandwidth, and reliability in over-the-air (OTA)/radio frequency (RF) and physical (PHY)-layer levels. Some IoT applications, such as smart parking and waste management services, need small packets to send, but they require high energy efficiency so that their batteries will last for a long time. Alternatively, some applications, such as virtual reality and multimedia systems, require high throughput communications, while their communications are constrained by channel bandwidth. Yet, wireless IoT networks are vulnerable to unintentional interference signal or jamming attacks. This is of particular importance of healthcare gadgets and vehicular networks

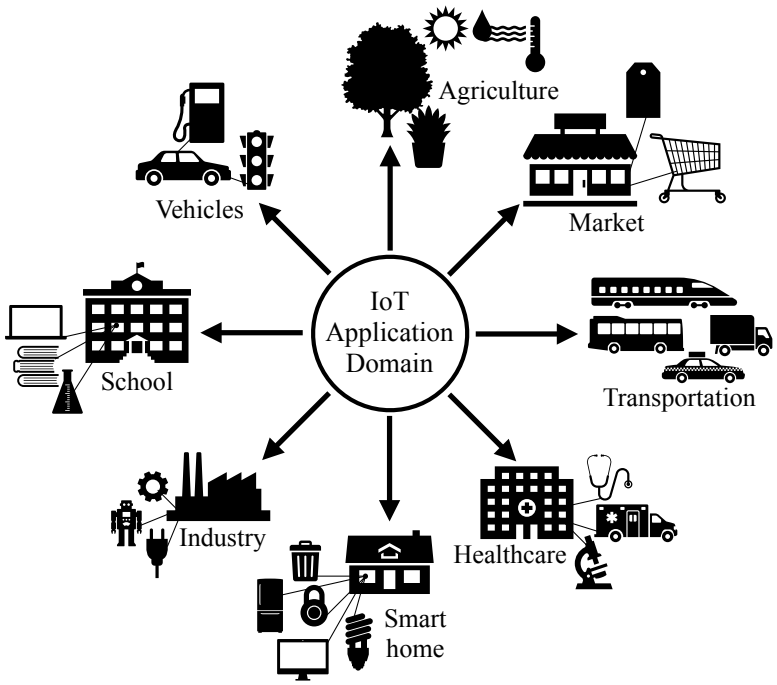


Figure 1.1: Examples of IoT applications.

Part I: Energy-Efficiency	Part II: Spectral-Efficiency	Part III: Physical-Layer Security
• Energy-efficient IoT communications for WLANs	• Uplink distributed MIMO communications for WLANs	• Securing VANET against high-power constant jamming attacks
• Transparent coexistence of IoT and Wi-Fi devices	• Joint uplink and downlink MU-MIMO transmission for LoRa	• Securing ZigBee communications against constant jamming attacks

Figure 1.2: The research scope of this thesis.

as the connection loss caused by the interference and jamming signal may lead to irreparable consequences.

In what follows, we will elaborate on the research scope of this thesis, and then briefly introduce our proposed schemes and summarize the key contributions of this thesis.

1.1 Research Scope

The research scope of this thesis, as shown in Fig. 1.2, is categorized in three parts: energy efficiency, spectral efficiency, and physical-layer security. While a large body of work in the

past studied these constraints, they mainly focus on algorithm and protocol design in isolation and often without taking the IoT application into account. These approaches have demonstrated a significant enhancement for connectivity of IoT devices, however the state-of-the-art is still far from satisfaction in practice. In this thesis, we present multiple schemes that address these constraints and enable efficient and ubiquitous wireless connectivity for IoT applications in real-world scenarios. The proposed solutions are of higher capabilities than the state-of-the-art technologies and can be integrated into the existing wireless infrastructure.

1.1.1 Energy-Efficiency

IoT devices are typically battery-powered and limited by their physical size. They are expected to work for many years (e.g., 10 years) without battery replacement. Energy-efficient wireless connection is therefore a key element for them to interact with the cyber-physical world. Enabling an energy-efficient IoT communication scheme for Wi-Fi networks is particularly of high importance due to the following two reasons. First, Wi-Fi is the dominant Internet service infrastructure in indoor environments. It also has extensive outdoor coverage in urban and suburban areas. By upgrading Wi-Fi access point (AP)'s air interface, the existing Wi-Fi infrastructure can be leveraged to offer energy-efficient IoT services in many scenarios. Second, Wi-Fi has demonstrated its success as an Internet provider for mobile devices. As expected, enabling IoT communication in Wi-Fi will dramatically offload the cellular IoT traffic, thereby mitigating the traffic congestion in cellular networks. Given its potential, a successful design of energy-efficient IoT design for Wi-Fi networks will accelerate the evolution of IoT ecosystems. This thesis introduces EE-IoT, an energy-efficient IoT communication scheme that allows low-complex non-OFDM IoT devices to communicate with an orthogonal frequency division multiplexing (OFDM)-based wireless local area network (WLAN) AP at a very low sampling rate, thereby leading to a significant reduction in

IoT devices' hardware complexity and power consumption. We have built a prototype of EE-IoT on a USRP2 wireless testbed and evaluated its performance in an office building environment. The experimental results show that an AP can serve 24 designed IoT devices simultaneously, and each IoT device can achieve more than 187 kbps in the downlink and more than 125 kbps in the uplink.

1.1.2 Spectral Efficiency

The proliferation of IoT applications has led to unprecedented demands for wireless services. The predicament many IoT networks face is that the increase of their capacity cannot catch up with the growth of wireless demands, especially given that frequency bands suitable for most IoT applications are limited. Such a predicament becomes particularly daunting in dense wireless environments such as conference rooms, football stadiums, cinemas, and airports. The stagnation can be attributed to the lack of practical interference management or collision recovery algorithms that can decode multiple data packets in the absence of inter-node/network synchronization.

This thesis proposes to improve the spectral efficiency of WLANs and LoRaWANs by jointly processing the signals from/to multiple users in the absence of fine-grained inter-node synchronization. First, we focus on WLANs that are widely deployed in indoor environments and regarded as a key component of the telecommunications infrastructure in our society. This thesis introduces UD-MIMO, a practical uplink distributed multiple-input multiple-output (MIMO) for WLANs, to enable concurrent uplink transmissions. Second, we introduce MaLoRaGW, the first-of-its-kind multi-antenna LoRa gateway that enables multi-user MIMO (MU-MIMO) LoRa communications in both uplink and downlink. We have built prototypes of UD-MIMO and MaLoRaGW on real-world wireless testbeds and demonstrated their compatibility with commercial-off-the-shelf (COTS) devices. Our experimental results show that, for a WLAN with 8 APs in a conference room, UD-MIMO offers $3.4\times$ throughput compared to interference-avoidance approach. We also evaluated

MaLoRaGW performance in three scenarios: lab, office building, and university campus. The results show that, compared to the state-of-the-art, the two-antenna MaLoRaGW increases uplink throughput by 10% and downlink throughput by 95%.

1.1.3 Physical-Layer Security

As IoT devices are mainly reliant on wireless connectivity for data transmissions/receptions, wireless security threats have become a big concern for confidentiality, integrity, and availability of IoT services. Compared to other security threats such as eavesdropping and data fabrication, IoT networks are particularly vulnerable to radio jamming attacks for the following reasons. First, jamming attacks are easy to launch. With the advances in software-defined radio, one can easily program a small $10 USB dongle device to a jammer that covers 20 MHz bandwidth below 6 GHz and up to 100 mW transmission power [7]. Such a USB dongle suffices to disrupt the Wi-Fi services in a home or office scenario. Other off-the-shelf SDR devices such as USRP [8] and WARP [9] are even more powerful and more flexible when using as a jamming emitter. The ease of launching jamming attacks makes it urgent to secure wireless IoT networks against intentional and unintentional jamming threats. Second, jamming attacks are mostly thwarted at the PHY layer but not at the MAC or network layer. When a wireless IoT network suffers from jamming attacks, its legitimate wireless signals are typically overwhelmed by irregular or sophisticated radio jamming signals, making it hard for legitimate IoT devices to decode data packets. Therefore, any strategies at the MAC layer or above are incapable of thwarting jamming threats, and innovative anti-jamming strategies are needed at the physical layer. Third, effective anti-jamming strategies for real-world wireless networks remain limited. Despite the significant advancement of wireless technologies, most of current IoT networks can be easily paralyzed by jamming attacks due to the lack of protection mechanisms. The vulnerability of existing IoT networks can be attributed to the

lack of effective anti-jamming mechanisms in practice. The jamming vulnerability of existing wireless networks also underscores the critical need and fundamental challenges in designing practical anti-jamming schemes [10].

In this thesis, we introduce practical anti-jamming strategies that secure IoT networks against jamming attacks. The proposed schemes secure ZigBee communications and vehicular networks (VANETs) against high-power constant jamming signals. Securing ZigBee and VANETs against jamming attacks is motivated by the following two reasons: First, jamming attacks are of particular importance in VANETs as the connection loss caused by the jamming signal may lead to car crashes and road fatalities. Second, ZigBee is used in many crucial IoT applications in real-world scenarios, such as healthcare devices, and therefore it is of great importance to secure its communications against jamming attacks. The proposed schemes take advantage of recent advances in MIMO technology to mitigate unknown jamming signals and decode the data packets. The experimental results show that the proposed schemes can decode the desired packets in the face of a 20 dB stronger jamming signal.

1.2 Contributions

The contributions of this Ph.D. thesis are summarized as follows.

- This thesis introduces EE-IoT [11] to enable an energy-efficiency IoT communications for WLANs. The key enabler of EE-IoT is an asymmetric PHY design that allows a broadband OFDM-based access point to communicate with multiple QAM-based (non-OFDM) IoT devices at a very low sampling rate. We built EE-IoT on a USRP-based wireless testbed and evaluated its performance in an office building environment. The experimental results show that an AP can serve 24 IoT devices simultaneously and each IoT device can achieve more

than 187 kbps in the downlink and more than 125 kbps in the uplink.

- This thesis develops EE-IoT scheme and introduces a practical design, termed WiFi-IoT [12], to enable a transparent coexistence of IoT and legacy Wi-Fi devices. We have built a prototype of the proposed schemes on a USRP-based wireless testbed and evaluated its performance in an indoor environment. Our experimental results show that an AP with two antennas can serve one Wi-Fi device and 24 IoT devices simultaneously in both downlink and uplink. The Wi-Fi device can achieve more than 36 Mbps in the uplink and more than 24 Mbps in the downlink. IoT devices can also achieve more than 375 kbps in both uplink and downlink transmissions.

- This thesis introduces UD-MIMO [13], a practical uplink distributed MIMO scheme for WLANs, to enable concurrent uplink transmissions in the absence of fine-grained inter-node synchronization. The enabling technique behind UD-MIMO is a practical solution to decoding uplink packets from asynchronous users. We implemented UD-MIMO on a USRP-based testbed and validated its compatibility with commodity Wi-Fi dongles. The experimental results show that UD-MIMO offers $3.4\times$ throughput compared to the interference-avoidance approach for a WLAN with 8 APs in a conference room.

- This thesis introduces MaLoRaGW [14], the first-of-its-kind multi-antenna LoRa GateWay that enables MU-MIMO LoRa communications in both uplink and downlink. The key component of MaLoRaGW is a joint baseband PHY design for uplink packet detection and downlink beamforming. We have built a prototype of two-antenna MaLoRaGW on a USRP device and extensively evaluated its performance with commercial LoRa dongles. Our experimental results show that, compared to the state-of-the-art, the two-antenna MaLoRaGW increases the throughput by 10% and reduces the packet error rate (PER) by 40% in uplink.

In downlink, it improves the throughput by 95% while maintaining a similar PER.

- This thesis proposes JammingBird [15], a MIMO-based receiver structure, that secures VANETs against constant jamming attacks. The key enablers of JammingBird are a jamming-resilient synchronization module and a jamming suppression module. The proposed receiver is capable of decoding desired packets under high-power constant jamming attacks, regardless of the PHY-layer technology employed by the jammer. We built JammingBird on a USRP-based wireless testbed and evaluated its performance in real-world outdoor scenarios. Our experimental results demonstrated that JammingBird can decode the desired packets in the face of 25 dB stronger jamming signal.

- This thesis introduces a learning-based receiver design that secures ZigBee communications against jamming attacks [16]. We design a neural network that serves as a linear spatial filter to suppress constant jamming attacks while not requiring any knowledge of the jamming signal. We built a prototype of ZigBee receiver on a wireless testbed to validate our design in real-world wireless environments and evaluated its performance in the presence of a malicious device that emits different types of radio jamming signals. Experimental results show that the proposed scheme is capable of decoding its packets in the face of 20 dB stronger jamming.

1.3 Organization

In the first part of this thesis, we study energy efficiency constraint in IoT networks. Chapter 2 presents EE-IoT in detail and how it enables energy-efficient IoT communications for WLANs. Chapter 3 presents the proposed transparent coexistence of EE-IoT and legacy Wi-Fi devices in detail. In the second part of this thesis, we study spectral efficiency constraint in IoT networks. Chapter 4 presents UD-MIMO, an uplink distributed MIMO for WLANs, which enables concurrent

uplink transmissions in the absence of fine-grained inter-node synchronization. Chapter 5 introduces MaLoRaGW, the first-of-its-kind multi-antenna LoRa GateWay that enables MU-MIMO LoRa communications in both uplink and downlink. In the third part of this thesis, we study PHY-layer security constraint in IoT networks. Chapter 6 presents JammingBird, which secures VANETs against constant jamming attacks. Chapter 7 presents the proposed learning-based receiver design to secure ZigBee communications against constant jamming attacks. Chapter 8 concludes this thesis.

Chapter 2

Energy-Efficient IoT Communications

While Narrow-Band Internet of Things (NB-IoT) has been standardized by 3GPP to provide wireless Internet access for IoT devices, this service is expected to come with a monthly fee (e.g., $1 or $2 per month per device). As the number of IoT devices tends to be large, the service charge will impose a considerable financial burden on the end users. In this chapter, we propose an Energy-Efficient IoT communication scheme by taking advantage of the existing WiFi infrastructure that is widely available in home, office, campus, and city environments. EE-IoT will not only avoid monthly service charge for the end users but also maintain a low power consumption for IoT devices. The key component of EE-IoT is an asymmetric PHY design, which enables an OFDM-based broadband AP to communicate with multiple QAM-based narrowband IoT devices at a low sampling rate (250 ksps) in both uplink and downlink. The trick in our design is that, instead of using the same carrier frequency as the AP, each IoT device tunes its carrier frequency to a particular subcarrier of the AP's OFDM signal, making it possible to encode/decode the data on that subcarrier at a low sampling rate. Based on this new PHY, we propose a medium access control (MAC) protocol to enable EE-IoT in WLANs. We have built a prototype of EE-IoT on a USRP2 wireless testbed and evaluated its performance in an office building environment. Experimental results show that an AP can serve 24 IoT devices simultaneously and each IoT device can achieve more than 187 kbps in the downlink and more than 125 kbps in the uplink.

2.1 Introduction

The Internet of Things is the network of physical devices with sensing, communicating, and actuating capabilities to integrate physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions. As IoT devices are typically battery-powered and limited by their physical size, energy-efficient wireless connection is a key element for them to communicate over the Internet. Narrowband IoT (NB-IoT), which is a Low Power Wide Area Network (LPWAN) radio technology standard [17], has been developed by 3GPP to enable a wide range of cellular services for IoT devices.

While NB-IoT has been standardized as a part of LTE, there are two concerns about its commercial applications. First, similar to cellular services for mobile phones, NB-IoT services will not come free. Users have to pay monthly fee to enjoy the NB-IoT services (e.g., $1 or $2 per month per device). Although this fee is not much compared to one's mobile phone bill, it easily becomes significant if one has many IoT devices on demand for Internet service. The monthly charge of NB-IoT services imposes a severe financial burden on the end users. Second, cellular networks are already very crowded. Serving extra billions of IoT devices may result in traffic congestion in cellular networks, especially considering the fact that the licensed spectrum bands suitable for energy-efficient IoT communications (below 6 GHz) are limited.

In this chapter, we study IoT communications in WLANs. This study is motivated by the following observations. First, WiFi is the dominant Internet service provider in indoor environments. It also has a large outdoor coverage in urban and suburban areas. By upgrading WiFi AP's air interface, the existing WiFi infrastructure can be leveraged to provide wireless Internet service for a large portion of IoT devices and avoid the monthly fee. Second, WiFi has demonstrated its success as an Internet provider for mobile devices. It is estimated that by 2020, WiFi will carry

38.1 exabytes traffic per month, continuing to exceed the monthly traffic in cellular networks (30.6 exabytes). As expected, enabling IoT communications in WLANs will dramatically offload the cellular IoT traffic, thereby mitigating the traffic congestion in cellular networks. Given its potential, a successful design of practical IoT communication scheme for WLANs will not only alleviate the above two concerns on NB-IoT services but also boost the prosperity and evolution of the IoT ecosystem.

To design a practical IoT communication scheme for WLANs, the challenge lies in preserving the energy efficiency of IoT devices. As most IoT devices are battery-powered and expected to work for many years without battery replacement, it is critical to minimize their power consumption for wireless communications. Simply embedding WiFi client chipset into IoT devices is not a plausible solution as it consumes too much energy for communications. To address this challenge, we propose EE-IoT, an energy-efficient IoT communication scheme for WLANs. The key component of our EE-IoT scheme is an asymmetric PHY design, which enables seamless uplink and downlink data transmissions between OFDM-based broadband WLAN AP and multiple QAM-based (non-OFDM) narrowband IoT devices. The asymmetric PHY is designed based on intrinsic properties of OFDM modulation and frequency mixer. In this asymmetric PHY, the AP preserves its legacy hardware architecture to transmit/receive OFDM-modulated broadband signals. But for each IoT device, instead of receiving/transmitting the broadband OFDM-modulated signal from/to the AP, it only receives/transmits narrowband signal on a single subcarrier by tuning its carrier frequency to that subcarrier. By doing so, the IoT devices can use a much lower sampling rate (250 ksps) for signal transmission/reception and do not require computation-intensive FFT/IFFT operations in their baseband signal processing, thereby leading to a significant reduction in their hardware complexity and power consumption. Based on the asymmetric PHY, we propose a semi-centralized MAC protocol for EE-IoT. As shown in Fig. 2.1, the reduction of analog-to-digital converter (ADC)

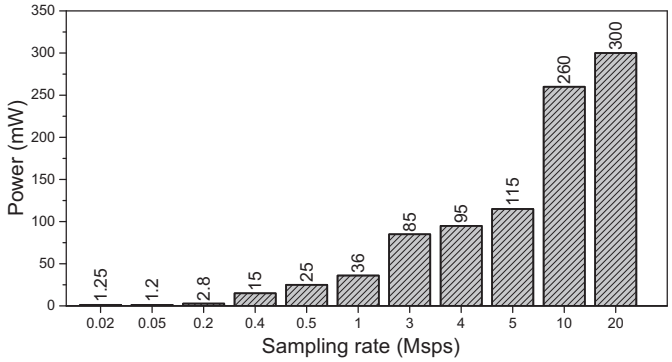


Figure 2.1: Power consumption versus sampling rate of ADC in wireless communication systems [20].

sampling rate (from 20 Msps to 250 ksps) can save about 300 mW (97%) power for an IoT device. Moreover, the elimination of FFT/IFFT operation can save another 41 mW power [18]. As ADC and FFT/IFFT are two of the most power-hungry components of a wireless transceiver [19], such an asymmetric PHY will lead to a significant power reduction for IoT devices.

We have built a prototype of EE-IoT on a GNURadio-USRP2 wireless testbed and have evaluated its performance in an office building wireless environment. Experimental results show that EE-IoT can serve 24 IoT devices simultaneously in a 802.11-based OFDM frame in both uplink and downlink, and each IoT device can achieve more than 187 kbps in downlink and more than 125 kbps in uplink.

2.2 Related Work

NB-IoT in Cellular Networks. As our society evolves to smart era, NB-IoT has attracted tremendous research efforts in both industry and academia. While there are many research results of NB-IoT (see, e.g., [17, 21–23]), most of them are limited to cellular networks. Our work focuses on IoT communications in WiFi networks, and thus differs from NB-IoT essentially.

NarrowBand WiFi Communications. Recently, there are some pioneering research efforts

from the industry to explore the feasibility of narrowband WiFi communications. In [24], Bluetooth Low Energy (BLE) was studied in 802.11ax WLANs to support IoT applications. In [25], an overlay narrow-band IoT communication approach was studied in 802.1ax WLANs. In [26], narrow-band IoT communications in WiFi networks were studied and evaluated using simulation from MAC layer protocol perspective. However, these results remain in conceptual discussion and theoretical exploration without considering practical issues in real implementation. Our work differs from these efforts significantly.

Cross-Technology Communications. Another research line in relevance to this work is WiFi and ZigBee cross-technology communications [27, 28]. However, these efforts aim to enable cross communications between different types of wireless devices without hardware modification. Furthermore, the existing results can enable only one-way communication (from WiFi transmitter to ZigBee receiver). This work differs from this research line fundamentally.

2.3 Problem Statement

Network Setting. We consider a WLAN as shown in Fig. 2.2, where an AP serves both standard WiFi devices and IoT devices. To serve two types of devices, the AP uses time division multiplexing. That is, the AP serves two types of devices in different time slots. Thus, there is no interference or interaction between WiFi devices and IoT devices. We focus on the wireless communications between the AP and the IoT devices.

Design Objective. In such a WLAN, our objective is to design an energy-efficient scheme to enable uplink and downlink communications between the AP and the IoT devices. We note that the term "energy-efficient" is to emphasize the low power consumption for the IoT devices, which will be achieved through reducing their sampling rate and baseband signal processing complexity.

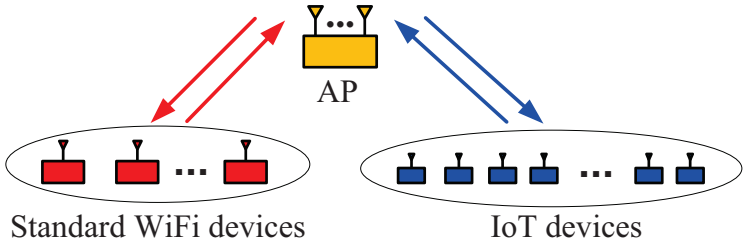


Figure 2.2: A WLAN with standard WiFi devices and IoT devices.

Since the AP typically has sufficient power supply, its energy consumption is not considered in our design. In addition, we aim to preserve the AP's hardware architecture as much as possible when extending its service from standard WiFi devices to IoT devices.

2.4 Mathematical Foundation of EE-IoT

In WLANs, the AP uses OFDM modulation for data transmission. Denote N as the number of FFT/IFFT points, which is also the number of subcarriers (e.g., $N = 64$ in 802.11ac). Denote $[X(0),\ X(1), \cdots, X(N-1)]$ as the frequency-domain data sequence of an OFDM symbol. Then, the time-domain data sequence of the OFDM symbol, which we denote as $[x(0),\ x(1), \cdots, x(N-1)]$, can be obtained through IFFT operation as follows:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j\frac{2\pi}{N}nk}. \tag{2.1}$$

For notational convenience, we reorganize the frequency-domain sequence $[X(0),\ X(1), \cdots, X(N-1)]$ by defining $X_s(k)$ as follows:

$$X_s(k) = \begin{cases} X(k), & 0 \le k \le N/2; \\ X(k+N), & -N/2+1 \le k \le -1. \end{cases} \tag{2.2}$$

Then, the IFFT operation in (2.1) can be rewritten as follows:

$$x(n) = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} X_s(k) \cdot e^{j\frac{2\pi}{N}nk}. \tag{2.3}$$

At the AP, the time-domain discrete data sequence $[x(0), x(1), \cdots, x(N-1)]$ is then converted to continuous signal by digital-to-analog converter (DAC). The resulting baseband waveform, which we denote as $x_b(t)$, can be written as:

$$x_b(t) = A \sum_{k=-N/2+1}^{N/2} X_s(k) \cdot e^{j2\pi k\Delta f t}\ , \tag{2.4}$$

where Δf is OFDM subcarrier spacing bandwidth (e.g., $\Delta f = 312.5$ kHz in 802.11ac) and A is a constant that denotes signal amplitude.

Then, the baseband waveform is up-converted to RF signal with carrier frequency f_c. The resulting radio signal can be written as:

$$x_r(t) = A \sum_{k=-N/2+1}^{N/2} X_s(k) \cdot e^{j2\pi k\Delta f t} \cdot e^{2\pi f_c t}\ , \tag{2.5}$$

which can be rewritten as:

$$x_r(t) = A \sum_{k=-N/2+1}^{N/2} X_s(k) \cdot e^{j2\pi (f_c + k\Delta f) t}\ . \tag{2.6}$$

Suppose that the AP uses a single subcarrier, say subcarrier k, in the OFDM symbol to send data to an IoT device. That is, the AP puts payload on subcarrier k and puts zeros to other subcarriers.

Then, the transmitted radio signal at the AP can be written as:

$$x_r(t) = A \cdot X_s(k) \cdot e^{j2\pi(f_c+k\Delta f)t} \ . \tag{2.7}$$

Assume that the signal on the subcarrier experiences flat fading from the AP to the IoT device, which is true in practice as a single subcarrier is narrowband (312.5 kHz). Then, the received radio signal at the IoT device can be written as:

$$y_r(t) = B \cdot X_s(k) \cdot e^{j2\pi(f_c+k\Delta f)t} \ , \tag{2.8}$$

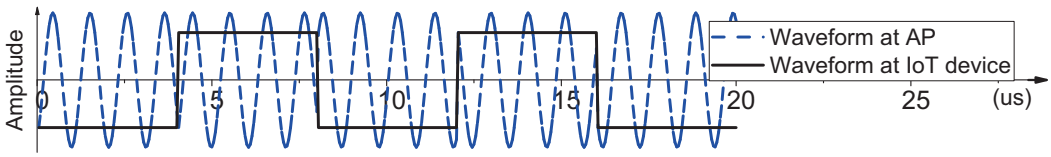
where B is a constant complex number that characterizes path loss and flat fading coefficient.

Equation (2.8) indicates that, if the IoT device wants to decode the signal on subcarrier k, it does not require OFDM demodulation. Instead, it can use center/carrier frequency $f_c + k\Delta f$ to down-convert the radio signal. The down-converted baseband signal, which we denote as $y(t)$, can be written as:

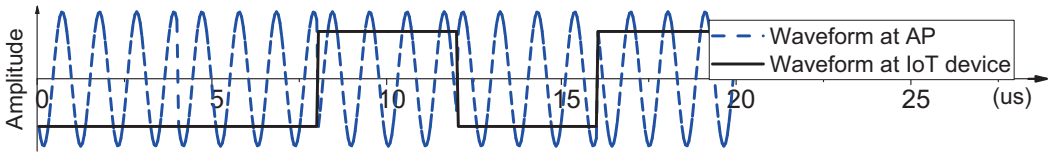
$$\begin{aligned} y(t) &= B \cdot X_s(k) \cdot e^{j2\pi(f_c+k\Delta f)t} \cdot e^{-j[2\pi(f_c+k\Delta f)t-\phi]} \ , \\ &= Be^{j\phi} \cdot X_s(k) \ , \end{aligned} \tag{2.9}$$

where ϕ is the phase offset between radio carrier signal and the local clock signal generated by IoT device's oscillator.

The result in (2.9) has the following two implications. First, the received baseband signal at the IoT device is a constant over the time duration of an OFDM symbol. This means that the required sampling rate at the IoT device is the OFDM symbol rate. For instance, Fig. 2.3 shows the baseband waveforms at the AP and the IoT device when a single subcarrier is used for data transmission. It



(a) Real part of waveforms at Tx and Rx.



(b) Imaginary part of waveforms at Tx and Rx.

Figure 2.3: The baseband signal waveform at OFDM AP versus the baseband signal waveform at non-OFDM IoT device.

is evident to see that 250 ksps sampling rate is sufficient for the IoT device to decode the signal on the subcarrier. Second, $Be^{j\phi}$ in (2.9) represents the compound channel effect on baseband signal, which is a complex constant in block fading channel. Hence, it is easy for the IoT device to decode the signal of interest if the reference signals are embedded on the transmitter side. These two observations lay the mathematical foundation for our design of an asymmetric PHY for EE-IoT.

2.5 PHY Design for EE-IoT: Downlink

In this section, we design a practical PHY to enable downlink data transmission from an OFDM-based broadband AP to multiple (non-OFDM) narrowband IoT devices. In what follows, we first present our proposed frame format for data transmission and then present the PHY design for single-user case. Finally, we extend our PHY design to multi-user case.

2.5.1 Frame Format

Fig. 2.4 depicts our proposed frame format. We elaborate each field of the frame as follows:

- *Preamble field:* The preamble field is designed for synchronization and channel estimation.

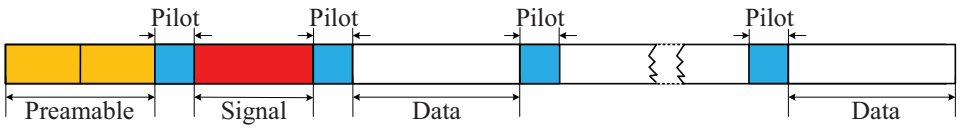


Figure 2.4: Physical-layer frame format.

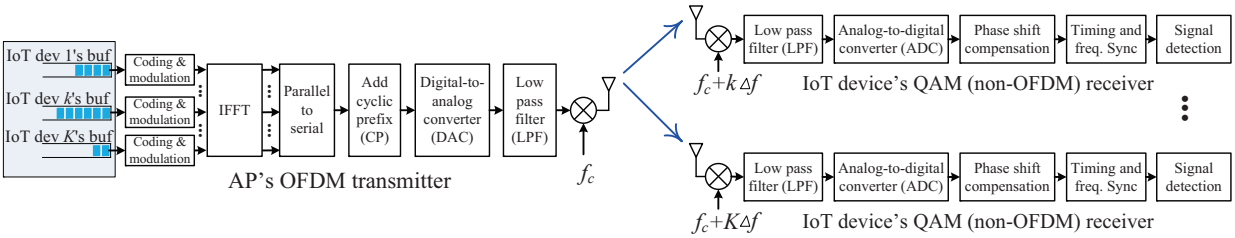


Figure 2.5: Downlink PHY design for data transmission from a broadband AP to K narrowband IoT devices.

It consists of two identical Zadoff-Chu sequences of M_p symbols (e.g., $M_p = 12$ in our experiment).

- *Signal field:* The signal field is used to define the modulation and coding scheme (MCS) used in the data field as well as the total length of the frame. The MCS type of the symbols in this field is fixed (e.g., BPSK and 1/2 coding rate). The number of symbols in this field, which we denote as M_s, can be flexibly defined.

- *Pilot field:* The pilot field is one reference symbol, which is used to correct phase offset for signal detection at the receiver.

- *Data field:* The data field is used to carry payloads. The number of symbols in this field, denoted as M_d, can be user-defined (e.g., $M_d = 50$ in our experiment).

2.5.2 PHY Design: Single-User Case

Fig. 2.5 presents our downlink PHY design for an OFDM-based AP to serve a single IoT device. As shown in Fig. 2.5, The AP preserves its legacy architecture. Specifically, the AP does not require

hardware modification to serve IoT device. The only manipulation is that the AP uses a single subcarrier (say subcarrier k) for data transmission and leaves other subcarriers unused. This is easy to be done through upper-layer control.

We now focus on the PHY design for the IoT device. Fig. 2.5 shows its PHY modules, which we elaborate as follows:

Local Carrier Frequency. To decode the signal from the AP, the IoT device tunes its local carrier frequency to $f_c + k\Delta f$, where f_c is the carrier frequency used at the AP and k is the index of the subcarrier used for payload. Note that the index of subcarrier is reordered in (2.2) and thus k is in the range of $[-N/2+1,\ N/2]$.

Bandwidth of LPF. For the down-converted signal from the frequency mixer, the IoT device uses a low pass filter (LPF) to suppress the noise. Configuration of its bandwidth is a trade-off problem. On the one hand, a large bandwidth will cause less signal distortion but bring more noise. On the other hand, a small bandwidth will cause more signal distortion but filter out more noise. Therefore, the optimal bandwidth of the LPF is determined by the SNR. In our experiment, we set the bandwidth of the LPF to $10 \times \Delta f = 3.125$ MHz.

Sampling Rate of ADC. On the transmitter side, the AP sends one QAM symbol on a single subcarrier in one OFDM symbol. Thus, the symbol rate at the IoT device is equal to the OFDM symbol rate at the AP. For a legacy AP, its sampling rate is 20 Msps and each OFDM symbol has 80 samples (64 samples in data part and 16 samples in cyclic prefix (CP) part). Hence, the QAM symbol rate at the IoT device is 250 ksps (20 Msps divided by 80). Therefore, the required sampling rate of the ADC is 250 ksps.

Phase Shift Compensation. The analytical results in Section 2.4 show that, by using an appropriate local carrier frequency, the IoT device can perfectly decode the signal on a subcarrier in the OFDM symbol from the AP. However, the analytical study in Section 2.4 did not consider the

CP in the OFDM symbols. In practice, the CP is attached at the beginning of each OFDM symbol, which is $16 \times 0.05 = 0.8$ μs in 802.11 WLANs. As the time period of the signal on subcarrier k is $0.05 \times 64/k = 3.2/k$ μs, the phase shift caused by the CP is $2\pi \times \frac{0.8}{3.2/k} = 2\pi \times \frac{k}{4}$ radians. To decode the signal at the IoT device, the phase shift caused by CP must be compensated for each OFDM symbol.

For the phase shift compensation module in Fig. 2.5, denote $y_{\text{in}}(n)$ as its input data symbol sequence; denote $y_{\text{out}}(n)$ as its output data symbol sequence. Then, the operation of this module can be written as:

$$y_{\text{out}}(n) = y_{\text{in}}(n) \cdot e^{j2\pi \frac{nk}{4}} \ , \tag{2.10}$$

where n is the time-domain symbol index, and k is the index of the subcarrier that is used for the IoT device. Note that the initial phase can be arbitrarily selected as it will be tackled by the signal detection module.

Timing and Frequency Synchronization. The purpose of timing synchronization is to search for the bursty signal frames in the received signal stream at the IoT device. To reduce the computational complexity, we propose a three-step strategy for timing synchronization, which combines energy detection (low complexity), auto-correlation of the preamble signal (coarse search), and cross-correlation of the preamble signal (fine search). Specifically, in step 1, we detect the energy of the received signal. If the detected energy is below a pre-defined threshold, then the search procedure stops with a false output; otherwise, we go to step 2. In step 2, we auto-correlate the received signal stream with a distance M_p, with the aim of identifying the two identical pieces of Zadoff-Chu signals in the preamble. If the auto-correlation value is smaller than a pre-defined threshold, the search procedure stops with a false output; otherwise, we identify a small search area and go to step 3. In step 3, we cross-correlate the received signal with a local copy of the

preamble in the identified small search area. If the cross-correlation value is smaller than a pre-defined threshold, then the search procedure stops with a false output; otherwise, a signal frame is successfully found at the position with the maximum cross-correlation value.

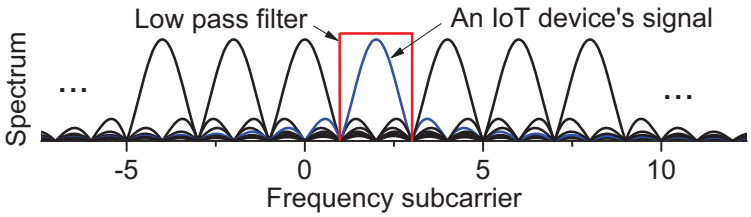
Once a signal frame is found, we then conduct frequency synchronization. The purpose of frequency synchronization is to estimate the frequency offset and compensate for it. We take advantage of the two identical pieces of Zadoff-Chu signals in the preamble to estimate the frequency offset. Mathematically, the phase offset per symbol caused by the frequency offset, which we denote as θ, can be written as:

$$\theta = \frac{1}{M_p} \cdot \angle\Big(\sum_{n=P}^{P+M_p-1} y(n) y(n+M_p)^* \Big), \tag{2.11}$$

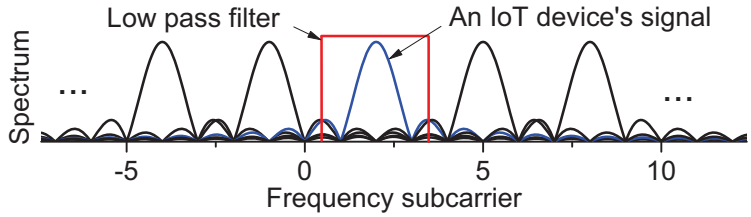
where $y(n)$ is the received baseband signal stream, P is the beginning position of a frame (from timing synchronization), $(\cdot)^*$ is conjugate operator, $\angle(\cdot)$ is the angle of a complex number. After the frequency offset estimation, we can compensate for the frequency offset for the sampled signal. Note that QAM signal detection is not as susceptible to frequency offset as OFDM signal detection and, therefore, the frequency synchronization accuracy at IoT device is not that demanding as that in OFDM systems.

Signal Detection for Single-User Case. For the wireless channel between the AP and the IoT device, the delay spread is much less than the time duration of an OFDM symbol. Moreover, the bandwidth of the IoT device's LPF is much larger than the signal bandwidth (Δf). Therefore, the compound channel from the transmitter to the receiver can be modeled as a flat fading channel.

To detect the signals that have experienced flat fading channel, we take advantage of the distributed pilot signals (see Fig. 2.4). We can first use the pilot signals to estimate the channel coefficient and then use the calculated channel coefficient to equalize the channel for signal detection.



(a) Signal spectrum when $K = 24$.



(b) Signal spectrum when $K = 16$.

Figure 2.6: Transmit signal spectrum at the AP when different numbers of subcarriers are used for downlink transmission.

2.5.3 Downlink PHY Design: Multi-User Case

Number of IoT Devices (K). The previous design is focused on the case where the AP uses a single subcarrier to serve one IoT device. We now extend the downlink PHY design to the case where the AP uses K subcarriers to serve K IoT devices. Suppose that the OFDM modulation at the AP has 64 subcarriers and 48 of them are used for possible data transmission. Fig. 2.6(a) and (b) illustrate the transmit signal spectrum at the AP when $K = 24$ and $K = 16$, respectively.

Inter-Subcarrier Interference. For an IoT device, it intends to receive the signal on a particular subcarrier. The signals on the other subcarriers constitute inter-subcarrier interference for this IoT device. As shown in Fig. 2.6, the signals on all the subcarriers overlap in the spectral domain. How to manage the inter-subcarrier interference is a critical problem for an IoT device to decode its desired signal.

A natural approach to address the inter-subcarrier problem is by reducing the spectrum utilization. In other words, by decreasing the number of data-carrying subcarriers at the AP (i.e., K), the inter-subcarrier interference at the IoT devices will be alleviated. However, the decrease of K will

result in a low spectrum efficiency. A trade-off between spectrum efficiency and each IoT device's throughput will be studied using experimental results in Section 2.9.

Inter-Symbol Interference. For a given K, each IoT device uses a low pass filter with bandwidth $24\Delta f/K$ to suppress inter-subcarrier interference, as illustrated in Fig. 2.6. As the bandwidth of the filter is smaller than that of the desired signal, the use of such a filter will inevitably cause inter-symbol interference. Although there are many sophisticated techniques to combat inter-symbol interference (e.g., Viterbi sequence detector [29]), these techniques pursue the optimal performance and thus have a high computational complexity. As IoT devices are limited by their computational capability, those sophisticated techniques may not be suited for IoT devices. In light of this, we propose a low-complexity approach to combat inter-symbol interference by taking advantage of the Zadoff-Chu sequences in the preamble.

Signal Detection for Multi-User Case. Our signal detection approach consists of two steps: channel estimation and channel equalization. We first introduce the mathematical modeling and then present the proposed channel estimation and equalization methods.

Mathematical Modeling: The compound channel consists of over-the-air channel and the RF circuit response. While the delay spread of the over-the-air channel is very small compared to the time duration of an OFDM symbol, the delay spread of the LPF at the IoT devices is significant. Thus, we model the channel as a multi-tap channel with tap coefficients $[h_{-L}, \cdots, h_{-1}, h_0, h_1, \cdots, h_L]$.

With a bit abuse of notation, we denote $\{x(n)\}$ as the transmit signal on that subcarrier at the AP; and denote $\{y(n)\}$ as the received signal at the IoT device. Then, the transfer function can be written as:

$$y(n) = \sum_{l=-L}^{L} h_l \cdot x(n-l) + w(n) \ , \tag{2.12}$$

where $w(n)$ is the combination of noise and residual inter-subcarrier interference.

Channel Estimation: To estimate the channel tap coefficients, we take advantage of the two identical Zadoff-Chu sequences in the frame preamble. Denote $[z(0),\, z(1),\, \cdots,\, z(M_p)]$ as the Zadoff-Chu sequence in the preamble. Denote $[z_i(0),\, z_i(1),\, \cdots,\, z_i(M_p)]$ as a cyclically shifted version of this Zadoff-Chu sequence. That is,

$$z_i(n) = z((n-i)\%M_p), \quad 0 \le n \le M_p, \tag{2.13}$$

where $\%$ is modulus operator.

Denote $[y(0),\ y(1), \cdots, y(2M_p-1)]$ as the received signal sequence in the frame preamble. With a local copy of the Zadoff-Chu sequence, the estimated channel tap coefficients, which we denote as $\hat{h}_l$, can be written as:

$$\hat{h}_l = \frac{1}{M_p} \sum_{n=0}^{M_p-1} y(n+l+M_p/2) \cdot z_{M_p/2}(n)^*, \quad -L \le l \le L. \tag{2.14}$$

For this channel estimation method, we have the following result: If the noise and the inter-subcarrier interference are negligible and $L \le M_p/4$, then the channel tap coefficients can be perfectly estimated, i.e., $\hat{h}_l = h_l$ for $-L \le l \le L$. This result stems from the nice property that the auto-correlation of a Zadoff-Chu sequence with a cyclically shifted version of itself is zero.

Channel Equalization: After channel estimation, we then equalize the channel for signal detection. Note that, while there are many signal detection methods pursuing the optimal detection performance [29], the proposed equalization method aims to preserve the low computational complexity of IoT devices by leveraging the observed channel characteristics for approximation.

Based on (2.12), we have:

$$\begin{aligned}\hat{x}(n) &= \frac{1}{h_0}\left(y(n) - \sum_{l=-L\cdots L}^{l\neq 0} h_l \cdot \hat{x}(n-l) - w(n)\right) \\ &\overset{(a)}{\approx} \frac{1}{h_0}\left(y(n) - \sum_{l=-L\cdots L}^{l\neq 0} \frac{h_l}{h_0} \cdot y(n-l) - w(n)\right) \\ &\overset{(b)}{\approx} \frac{1}{h_0}\left(y(n) - \sum_{l=-L\cdots L}^{l\neq 0} \frac{h_l}{h_0} \cdot y(n-l)\right), \end{aligned} \tag{2.15}$$

where (a) follows from our observation that $|h_l| \ll |h_0|$ for $-L \leq l \leq L$ and $l \neq 0$, and (b) follows from that we ignore the noise.

After channel equalization, we then use the pilot signals in the frame to estimate the phase offset for each segment of data symbols and compensate for the phase offset correspondingly.

Computational Complexity: The proposed signal detection method has a linear computational complexity with the length of the frame. Specifically, its computational complexity is $O(LF)$, where F is the number of symbols in the frame.

2.6 PHY Design for EE-IoT: Uplink

In this section, we present our uplink PHY design for the data transmission from multiple (K) IoT devices to an AP. From the signal processing perspective, the uplink PHY design is a converse of the downlink PHY design. However, the challenge in the uplink PHY design is different from that in the downlink PHY design. In the uplink, if the QAM-modulated signals from IoT devices are perfectly synchronized in both time and frequency domains, the AP can decode the QAM signals on its subcarriers as if the signals were from a single OFDM transmitter. Hence, the real challenge in the uplink PHY design is the timing and frequency synchronization among the signals from

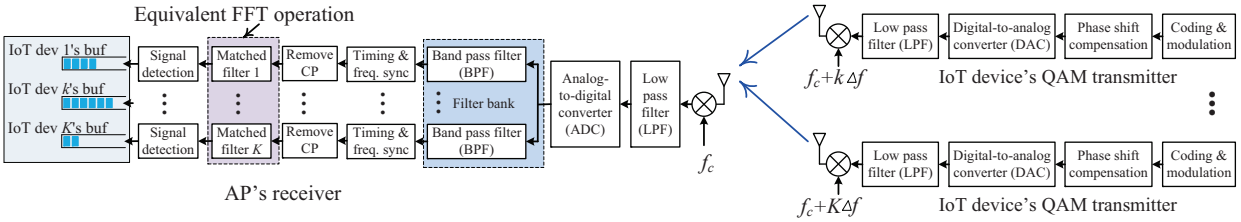


Figure 2.7: Uplink PHY design for data transmission from K narrow-band IoT devices to a broadband AP.

different IoT devices. In what follows, we present a PHY design to enable uplink data transmission between an AP and a set of asynchronous IoT devices.

2.6.1 Transmitter PHY for IoT devices

In the uplink data transmission, we use the same frame format as shown in Fig. 2.4. The two identical Zadoff-Chu sequences in the preamble will be used for synchronization and channel estimation, and the pilot signals will be used for phase offset correction. Fig. 2.7 depicts the uplink PHY design for the IoT devices, which we elaborate as follows.

Phase Shift Compensation. As we showed in the downlink PHY design, the CP in an OFDM symbol will introduce an extra delay for the signal on each subcarrier. This extra delay causes a phase shift for the signal on each subcarrier. For subcarrier k, the phase shift is $2\pi\frac{k}{4}$ radians, as we explained previously. Such a phase shift should be pre-compensated in order for the AP to decode the signal. Hence, the baseband signal processing module "Phase shift compensation" is designed for this purpose. Denote $x_{\text{in}}(n)$ and $x_{\text{out}}(n)$ as its input and output signal streams, respectively. Then, the function of this module can be written as:

$$x_{\text{out}}(n) = x_{\text{in}}(n) \cdot e^{-j2\pi\frac{nk}{4}} \ , \tag{2.16}$$

where n is the time-domain symbol index and k is the index of the subcarrier that is used for the

IoT device.

Settling Time of DAC. While the sampling rate of the DAC in the IoT device is 250 ksps, there is a requirement for the settling time of the DAC to maintain the rectangular shape of its output waveform. In this design, the settling time of the DAC should be less than the time duration of the CP, which is 0.8 μs. Note that such a requirement is very mild and can be met by many low-end DACs on the market.

Low Pass Filter. Similar to the LPF design in the downlink, the LPF design in the uplink is also a trade-off between signal distortion and noise suppression. In the uplink, the IoT device serves as a transmitter, where the noise is less significant compared to a receiver. Therefore, we set the bandwidth of the LPF to $15 \times \Delta f \approx 5$ MHz.

2.6.2 Receiver PHY for AP

As the IoT devices are driven by independent clock sources, their transmit signals are asynchronous when arriving at the AP. How to address the synchronization problem is the challenging task in the design of AP's receiver PHY. It is worth pointing out that the synchronization problem here is different from that in MU-MIMO in OFDM communications. This is because the IoT devices only have low-complexity transceivers that work at low clock rate (250 kHz). Sophisticated MAC protocols (e.g., Timing Advance [30]) cannot be applied to IoT devices to achieve timing and frequency synchronization on the transmitter side. Hence, the synchronization challenge has to be tackled at the PHY layer on the receiver side (on AP side).

To address the synchronization problem, we borrow the idea of filter bank from the SC-FDMA uplink in LTE networks, where a LTE base station decodes signals from multiple asynchronous user equipments (UEs) [31]. Specifically, as shown in Fig. 2.7, the AP first uses a bank of bandpass filters to separate the signals from different IoT devices. With the bandpass filters, the AP can

estimate and correct the synchronization errors independently for the signal from each IoT device. A shortcoming of this method is that perfect signal separation is not possible even with ideal brick-wall bandpass filters due to the frequency leakage among the adjacent subcarriers. A natural approach to addressing this shortcoming is to decrease the number of data-carrying subcarriers (i.e., K). As illustrated in Fig. 2.6, decreasing K can significantly reduce the inter-subcarrier interference. The impact of K on the performance of each IoT device in the uplink will be investigated using experimental results in Section 2.9. In what follows, we outline the key modules in each of the AP's signal paths.

Bandpass Filter. The number of bandpass filters that are used at the AP is equal to the number of IoT devices (i.e., K). For each bandpass filter, we set its normalized center frequency to $k/32$ and set its normalized bandwidth to $K/48$, where k is the index of the subcarrier used by the target IoT device. Note that we use the normalized frequency and bandwidth because the filters are applied in the digital domain at the AP.

Timing and Frequency Synchronization. The timing and frequency synchronization will be done in the same way as that in the downlink. Specifically, for timing synchronization, we exploit the auto-correlation property of the two identical Zadoff-Chu sequences in the preamble for coarse timing synchronization, and then exploit their cross-correlation property for fine timing synchronization. For frequency synchronization, we autocorrelate the two identical Zadoff-Chu sequences in the preamble to estimate the carrier frequency offset (CFO), and then compensate for the frequency offset in the digital domain.

Matched Filters (FFT). The bank of matched filters is equivalent to FFT operation at the AP.

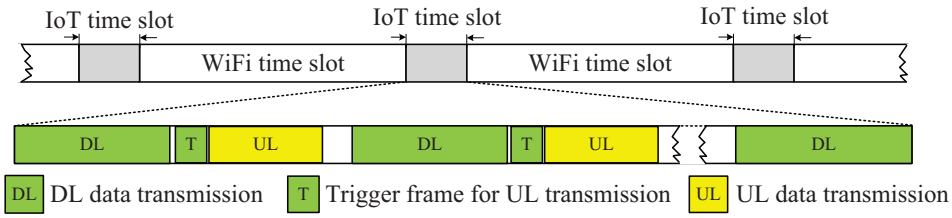


Figure 2.8: Resource allocation in the IoT communication protocol.

Specifically, the matched filter on path k can be written as:

$$Y_k = \sum_{n=0}^{N-1} y_k(n) e^{-j\frac{2\pi}{N}nk}, \tag{2.17}$$

where Y_k is the output data of the matched filter and $y_k(n)$, $0 \leq n \leq N-1$, is the input data sequence of the matched filter. It is easy to see that the matched filters are actually the same as FFT operation in math. The only difference is that different matched filters have different input data sequences for signal isolation.

Signal Detection. The signal processing module in the uplink is similar to its counterpart in the downlink. The main purpose of this module is to cancel inter-symbol interference and equalize the channel for signal recovery. The signal detection method that was proposed for the downlink multi-user case can be directly used here for the uplink signal detection. Furthermore, since the AP is not limited by power consumption and computational capability, more advanced signal detection methods (e.g., soft-decision Viterbi decoder [29]) can be employed to improve the detection performance.

2.7 MAC Protocol Design for EE-IoT

In this section, we outline our proposed MAC protocol for the communications between the AP and IoT devices in the WLAN as shown in Fig. 2.2.

Protocol Overview. As shown in Fig. 2.2, the AP needs to serve both standard WiFi and IoT devices. To do so, we propose a time division multiplexing scheme as shown in Fig. 2.8. The AP periodically reserves a time slot for the communications between itself and the IoT devices. During the WiFi time slot, the IoT devices can switch to sleep mode to reduce their power consumption. During the IoT time slot, the AP can silence the standard WiFi devices by broadcasting a network allocation vector (NAV) packet. The duration of an IoT time slot can be either fixed or adaptively set, depending on the system requirement.

Channel Assignment. In an IoT time slot, our PHY design can support K (e.g., $K = 16$ or $K = 24$) parallel independent channels for uplink and downlink data transmissions between AP and IoT devices. For a new or wake-up IoT device, it first listens to each of the K channels and selects the one with least traffic as its initial channel. After the selection, it will stick to this channel unless the AP assigns it to another channel. On the AP side, it maintains a list of active IoT devices. With the global information, it can perform an optimization procedure to adjust the channel assignment so as to improve the channel efficiency.

Downlink Transmission. The proposed MAC protocol is a semi-centralized protocol, where the AP is the controller for resource allocation. As such, it has full degree of freedom for downlink transmission scheduling on the K channels. In the downlink channels, the AP can periodically broadcast beacon frames that contain all the information about the network. The AP can also inform the IoT devices of its decision for channel re-assignment.

Uplink Transmission. As there are K parallel channels that can be used for uplink transmission, it is important to coordinate the IoT devices on those channels for uplink transmission. Thus, we have designed a special frame (called trigger frame for uplink transmission) for this purpose, as shown in Fig. 2.8. Specifically, an IoT device keeps listening its channel for downlink data transmission; it performs possible uplink data transmission only if it receives a *trigger frame* from

the AP. For each individual channel, carrier-sense multiple access/collision avoidance (CSMA/CA) is used to control the channel access among the IoT devices on this channel.

Uplink Power Control. A power control mechanism has been implemented for uplink data transmission. For each IoT device, it estimates the signal strength of the downlink trigger frame, based on which it adjusts its transmit power for uplink data transmission. By doing so, the AP will receive relatively similar signal power from the IoT devices on different channels. This mechanism improves the performance of AP's signal detection.

2.8 Implementation

To evaluate the practicality and performance of EE-IoT in real-world wireless environments, we have prototyped the proposed PHY design and MAC protocol on a wireless testbed that consists of USRP2 and GNU-Radio software package.

Frame Parameters. The frame format in Fig. 2.4 is used for data transmission in both uplink and downlink, with $M_p = 12$, $M_s = 50$ and $M_d = 50$. The total number of symbols in a frame is set to 1044. On the AP side, one symbol actually refers to one OFDM symbol in legacy 802.11 standard. Specifically, each OFDM symbol has 64 subcarriers, where 48 of them may be used for data transmission. The length of CP is 16 samples. The length of one OFDM symbol is 80 samples and its time duration is 4 μs.

Prototype of AP. We have built an AP using a USRP2 device and a laptop. We have implemented the proposed AP's PHY in Figures 2.5 and 2.7 in GNU-Radio on the laptop, which will control the USRP2 device to work in the way as designed. The maximum transmit power of the AP is set to 20 dBm. The sampling rate is set to 20 Msps.

Prototype of IoT devices. We have built three IoT devices using three independent USRP2

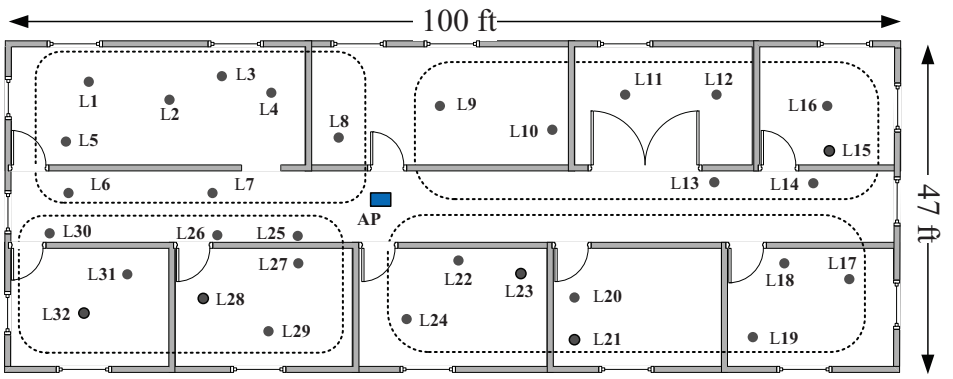


Figure 2.9: The floor plan for EE-IoT evaluation.

devices and laptops. We have implemented their PHY in Figures 2.5 and 2.7. The maximum transmit power of the IoT device is set to 0 dBm. While the symbol rate is 250 kHz, we use $4\times$ oversampling rate and therefore set the sampling rate to 1 Msps.

We also implemented a special device using USRP2 that can mimic up to 21 IoT devices. This device is used only for test purpose to emulate the inter-subcarrier interference in the uplink. Its performance will not be measured.

Prototype of MAC Protocol. We have implemented simplified version of the proposed MAC protocol, including downlink and uplink data transmission as well as uplink power control, with a fixed set of IoT devices. Time-sharing with standard WiFi devices, channel assignment at the AP, and uplink CSMA/CA among the IoT devices are not considered in our implementation.

2.9 Performance Evaluation

In this section, we evaluate the performance of EE-IoT.

2.9.1 Experimental Setup and Performance Metrics

Experimental Setup. We measure the performance of EE-IoT in an office building as shown in Fig. 2.9. The AP is placed at the spot marked "AP". The four IoT devices (three IoT devices and one special device to mimic multiple IoT devices in the uplink) are placed at 4 out of the 32 locations. Particularly, these four IoT devices are always placed in four different areas marked by dashed boxes. The purpose of this setting is to more authentically emulate the real network scenarios.

Performance Metrics. We use two performance metrics to assess the performance of EE-IoT. The first one is error vector magnitude (EVM), which is widely used in WiFi device tests. EVM quantifies the normalized error magnitude between the measured constellation and the ideal constellation. Mathematically, it can be written as:

$$\text{EVM (dB)} = 10\log_{10}\Big(\frac{\mathbb{E}(|x-\hat{x}|^2)}{\mathbb{E}(|x|^2)}\Big), \tag{2.18}$$

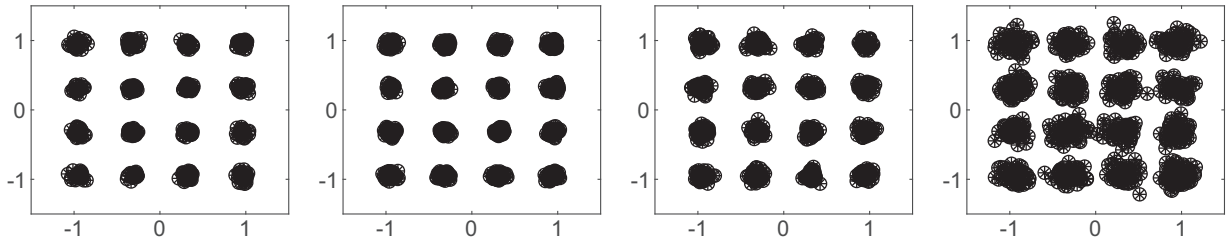
where x is the original signal at the transmitter and $\hat{x}$ is the estimated signal at the receiver. The second performance metric that we use is an IoT device's data rate. Different from EVM, which will be directly measured from the experimental results, the data rate will be estimated based on the MCS table specified in 802.11 standard, shown in Table 2.1, as:

$$r = \frac{M}{N} \times \text{BW} \times \gamma(\text{EVM}), \tag{2.19}$$

where M is the number of subcarriers used for data transmissions, N is the length of one OFDM symbol (including CP), and BW represents the channel bandwidth. $\gamma(\text{EVM})$ is the average number of bits carried by one symbol and its values are given in Table 2.1. Specifically, for an IoT device, its uplink and downlink data rate is estimated by: $r = \frac{1}{2} \times 250 \times \gamma(\text{EVM})$ kbps, where $\frac{1}{2}$ means

Table 2.1: EVM specification in IEEE 802.11ac standards [32].

EVM (dB)	(inf -5)	[-5 -10)	[-10 -13)	[-13 -16)	[-16 -19)	[-19 -22)	[-22 -25)	[-25 -27)	[-27 -30)	[-30 -32)	[-32 -inf)
Modulation	N/A	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM	64QAM	256QAM	256QAM
Coding rate	N/A	1/2	1/2	3/4	1/2	3/4	2/3	3/4	5/6	3/4	5/6
γ(EVM)	0	0.5	1	1.5	2	3	4	4.5	5	6	20/3



(a) $K = 1$ (b) $K = 12$ (c) $K = 16$ (d) $K = 24$

Figure 2.10: The constellation of the decoded signal in the downlink.

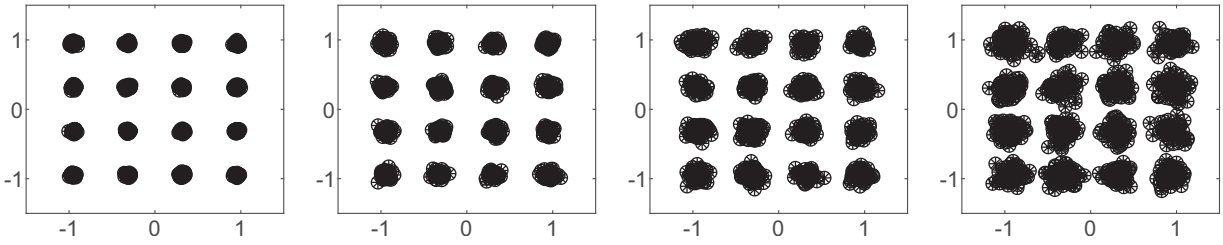
that one half time for downlink transmission and the other half for uplink transmission, $M = 1$, $N = 80$ samples, and $\mathrm{BW} = 20$ MHz.

2.9.2 A Case Study

We use a case study to show the details of downlink and uplink data transmission for the IoT device placed at Location 1 in Fig. 2.9.

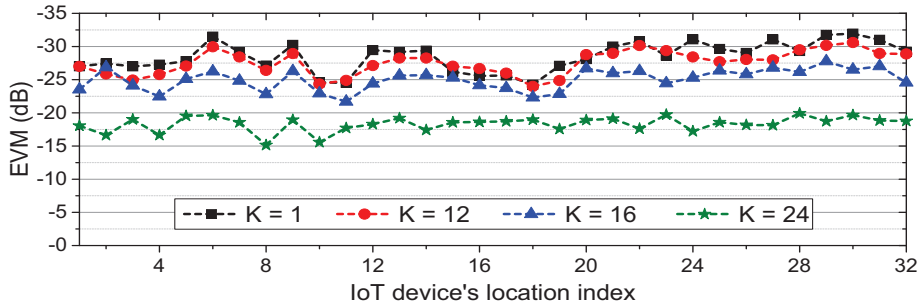
Downlink. Fig. 2.10 shows the constellation of the decoded signals at the IoT device when 16QAM is used. It is evident to see that the 16QAM can be successfully demodulated by this IoT device. Specifically, the measured EVM is -27.0 dB when $K = 1$, -26.9 dB when $K = 12$, -23.5 dB when $K = 16$, and -18.0 dB when $K = 24$.

Uplink. Fig. 2.11 shows the constellation of the decoded signals (from the IoT device at Location 1) at the AP when 16QAM is used. We can see that 16QAM can be successfully demodulated at the AP. Specifically, the measured EVM is -32.1 dB when $K = 1$, -25.4 dB when $K = 12$, -21.6 dB when $K = 16$, and -17.6 dB when $K = 24$.

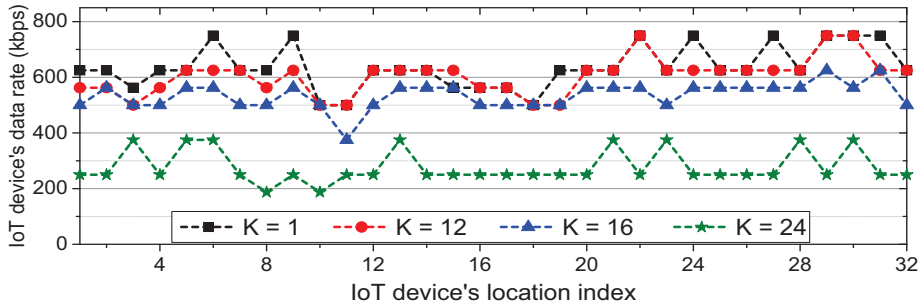


(a) $K = 1$ (b) $K = 12$ (c) $K = 16$ (d) $K = 24$

Figure 2.11: The constellation of the decoded signal in the uplink.



(a) One IoT device's EVM



(b) One IoT device's data rate

Figure 2.12: Measured EVM and calculated data rate for one IoT device in the downlink.

2.9.3 Complete Experimental Results

We now present the measured experimental results for one IoT device when it is placed at each of the 32 locations.

Downlink. Fig. 2.12(a) shows the measured EVM at one IoT device in the downlink when it is placed throughout the 32 locations. From the figure we can see that the maximum EVM is less than -24.2 dB when $K = 1$, less than -23.9 dB when $K = 12$, less than -21.7 dB when $K = 16$, and less than -15.1 dB when $K = 24$. We can also see that the measured EVM degrades

as K increases. This is because the inter-subcarrier interference becomes more significant as K increases, as we illustrated in Fig. 2.6.

Fig. 2.12(b) shows the data rate of the IoT device in the downlink. The results show that the achievable downlink data rate for this IoT device is greater than 500 kbps when $K \leq 12$, greater than 375 kbps when $K = 16$, and greater than 187 kbps when $K = 24$.

Uplink. Fig. 2.13(a) shows the measured EVM at the AP in the uplink when the target IoT device is placed throughout the 32 locations. From the figure we can see that the EVM is less than -23.3 dB when $K = 1$, less than -19.1 dB when $K = 12$, less than -15.3 dB when $K = 16$, and less than -10.1 dB when $K = 24$. Similar to the observation in the downlink, the EVM degrades as K increases in the uplink. This is caused by the inter-subcarrier interference as well.

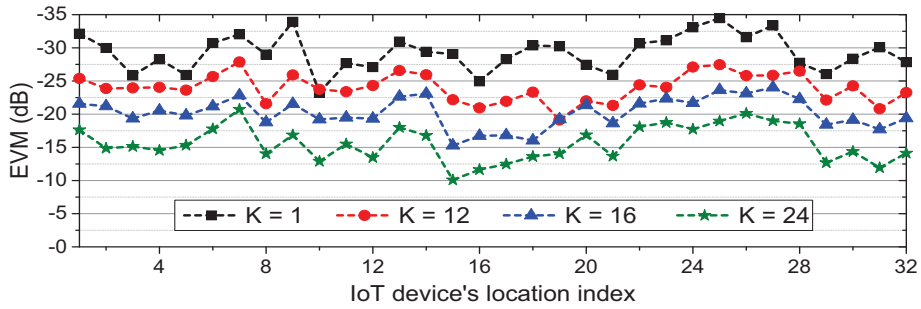
Fig. 2.13(b) shows the data rate for the target IoT device in the uplink. The results show that the achievable uplink data rate for this IoT device is greater than 500 kbps when $K = 1$, greater than 375 kbps when $K = 12$, greater than 187.5 kbps when $K = 12$, and greater than 125 kbps when $K = 24$.

2.9.4 Observations

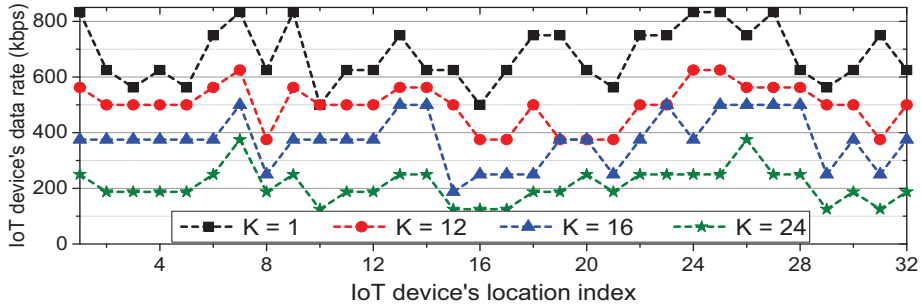
We summarize our observations as follows: A broadband AP can communicate with 24 narrowband IoT devices simultaneously. In a typical office building environment, the IoT devices can achieve more than 187 kbps in the downlink and more than 125 kbps in the uplink.

2.10 Chapter Summary

In this chapter, we proposed EE-IoT, an energy-efficient IoT communication scheme for WLANs. Compared to its counterpart, NB-IoT, EE-IoT takes advantage of the widely existing WiFi infrastructure to provide wireless Internet access for IoT devices, and thus will not incur additional



(a) One IoT device's EVM



(b) One IoT device's data rate

Figure 2.13: Measured EVM and calculated data rate for one IoT device in the uplink.

service fee to the end users. The key component of EE-IoT is an asymmetric PHY design, which enables an OFDM-based broadband AP to communicate with multiple (non-OFDM) narrowband IoT devices. In this asymmetric PHY, instead of using the same carrier frequency as the AP, an IoT device aligns its carrier frequency to a particular subcarrier of the AP's OFDM signals. Such a carrier frequency setting makes it possible for the IoT device to transmit/receive signal on a single subcarrier at a low sampling rate (250 ksps). We have evaluated the performance of EE-IoT in an office building environment. Experimental results show that an AP can serve 24 IoT devices simultaneously and each IoT device can achieve more than 187 kbps in the downlink and more than 125 kbps in the uplink.

Chapter 3

Coexistence of Energy-Efficient IoT Devices

In this chapter, we propose a practical design, termed WiFi-IoT, to enable a transparent coexistence of EE-IoT and Wi-Fi devices. WiFi-IoT enables a multi-antenna AP to serve Wi-Fi and IoT devices simultaneously, leading to an efficient utilization of spectrum. As we showed in chapter 2, EE-IoT enables an OFDM-based broadband AP to communicate with many QAM-based (non-OFDM) narrow-band IoT devices at a much low sampling rate (250 ksps). In the proposed EE-IoT scheme, presented in Chapter 2, AP schedules Wi-Fi and IoT devices into different time slots. Such a TDMA-based approach will avoid their mutual interference and create interference-free environments for their respective communications. However, since a considerable portion of airtime will be allocated to IoT devices, this approach tends to sacrifice the quality of service (QoS) for Wi-Fi devices. To enable transparent coexistence of Wi-Fi and IoT devices , we propose a SDMA-based approach that allows a multi-antenna AP to serve broadband Wi-Fi devices and narrow-band IoT devices simultaneously. The key component of our approach is a lightweight interference cancellation method, which can effectively mitigate the mutual interference in practice by leveraging the spatial degrees of freedom provided by AP's multiple antennas. We have built a prototype of WiFi-IoT on a GNURadio-USRP2 wireless testbed and evaluated its performance in an office building environment. Experimental results show that, using WiFi-IoT, an AP with two antennas can serve one Wi-Fi device and 24 IoT devices simultaneously in both uplink and downlink, with each IoT device achieving more than 375 kbps.

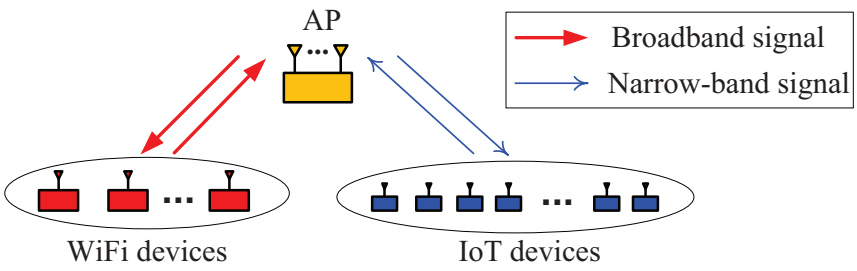


Figure 3.1: A future WLAN with both Wi-Fi and IoT devices.

3.1 Introduction

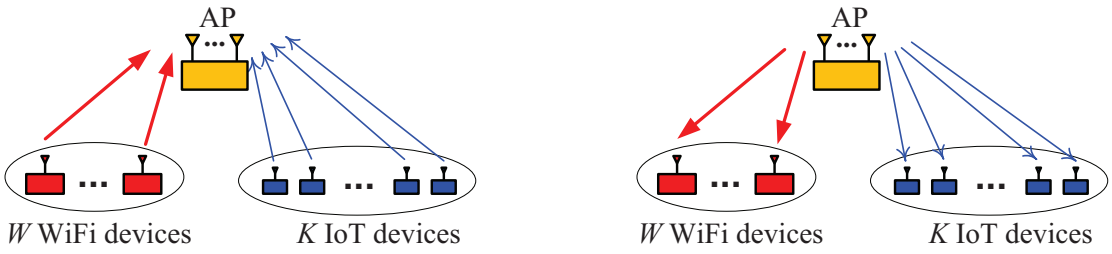
In this chapter, we propose a practical design, termed WiFi-IoT, to enable a transparent coexistence of EE-IoT and Wi-Fi devices, as illustrated in Fig. 3.1. To harmonize the coexistence of broadband Wi-Fi devices and narrow-band IoT devices, a straightforward approach is that the AP schedules Wi-Fi and IoT devices into different time slots, as discussed in . Such a TDMA-based approach will avoid their mutual interference and create interference-free environments for their respective communications. However, since a considerable portion of airtime will be allocated to IoT devices, this approach tends to sacrifice the QoS for Wi-Fi devices. To enable transparent coexistence of Wi-Fi and IoT devices and maximize the spectral efficiency, we propose a spatial division multiple access (SDMA)-based approach that allows a multi-antenna AP to serve broadband Wi-Fi devices and narrow-band IoT devices simultaneously. The key component of our approach is a lightweight interference cancellation method, which can effectively mitigate the mutual interference in practice by leveraging the spatial degrees of freedom provided by AP's multiple antennas. Specifically, in the uplink, we construct a spatial linear filter at the AP to decode the signals from each individual Wi-Fi/IoT device in the presence of cross-technology interference. In contrast to existing signal detection methods such as zero-forcing (ZF) and minimum mean square error (MMSE), our method does not require channel estimation and turns out to be very robust in practice. In the downlink, we construct beamforming filters for the AP to enable concurrent data transmissions. Different

from existing beamforming techniques, which require channel state information (CSI) for the construction of beamforming filters, our technique simply uses the decoding filters obtained in the uplink as the beamforming filters. The elimination of the need for CSI not only simplifies the system complexity, but it also reduces the airtime overhead induced by channel feedback. Leveraging these two interference cancellation techniques, WiFi-IoT is capable of serving Wi-Fi and IoT devices on the same spectrum simultaneously.

We have built a prototype of WiFi-IoT on a GNURadio-USRP2 wireless testbed and evaluated its performance in an office building environment. Experimental results show that, using WiFi-IoT, an AP with two antennas can serve one Wi-Fi device and 24 IoT devices simultaneously in both uplink and downlink, with each IoT device achieving more than 375 kbps. Our prototype provides a reference design to the community as an alternative solution to supporting energy-efficient IoT communication and sheds light on the integration of energy-efficient IoT communication in future Wi-Fi standards. We note that our design targets the stationary or semi-stationary environments such as smart home with smoke detection sensors, door opening sensors, or smart meters. The design of IoT communications in highly dynamic environments with frequent roaming is beyond the scope of this work.

3.2 Coexistence of Wi-Fi and IoT Communications

In chapter 2, we have presented a TDMA-based protocol to enable the coexistence of Wi-Fi and IoT devices. Specifically, the AP schedules Wi-Fi and IoT devices into different time slots so the mutual interference can be avoided in the time domain. In this chapter, we propose a more efficient coexistence scheme by taking advantage of the AP's multiple antennas to enable the spectrum sharing between the Wi-Fi and IoT devices in the spatial domain. Such a SDMA-based approach



(a) Uplink transmission (b) Downlink transmission

Figure 3.2: Coexistence of W Wi-Fi devices and K IoT devices in uplink and downlink. Each Wi-Fi device sends/receives broadband signals to/from the AP using OFDM modulation, whereas each IoT device sends/receives narrow-band signals to/from the AP using a single OFDM subcarrier.

will allow a multi-antenna AP to serve Wi-Fi and IoT devices simultaneously, thereby improving the spectral efficiency and scheduling flexibility. It is noteworthy that this work focuses on the design of efficient solutions to enable the coexistence of Wi-Fi and IoT devices. The cross-technology interference from ZigBee, Bluetooth, or other ISM devices is not considered in our design. This type of interference can be handled by other existing designs [33–37].

3.2.1 Basic Idea and Overview

The principle of our coexistence scheme is similar to that of MU-MIMO. In the uplink as shown in Fig. 3.2(a), the AP receives a blend of signals from all Wi-Fi and IoT devices. To decode those signals, the AP constructs a spatial filter (also called decoding filter or detection filter) that can cancel out the inter-user interference and recover the desired signal. Specifically, to decode the signals from a Wi-Fi device, the AP constructs a decoding filter for each of the subcarriers. Such a decoding filter will cancel the interference from other Wi-Fi devices and all the IoT devices. To decode the signals from an IoT device, the AP constructs a spatial filter that can cancel out the interference from all the Wi-Fi devices. Since different IoT devices use different subcarriers (radio frequencies), the signals from different IoT devices will not interfere each other.

In the downlink as shown in Fig. 3.2(b), to enable concurrent data transmissions, the AP

pre-cancels the interference using beamforming technique on the transmitter side, so that each Wi-Fi/IoT device will receive its desired signal without any interference. For each Wi-Fi device, the AP constructs a spatial filter (also called beamforming filter) for each of its subcarriers in the OFDM modulation. Such a beamforming filter will steer the signal power to the target Wi-Fi device while nullify the signal power at other devices. Similarly, for each IoT device, the AP constructs a beamforming filter for signal precoding. This beamforming filter will steer the signal power to the target IoT device and nullify the signal at other devices. With the beamforming at the AP, the Wi-Fi and IoT devices will only receive their desired signals and therefore are capable of decoding their respective data packets in the downlink.

While the principle is straightforward, a big question is how to construct the decoding filters in the uplink and the beamforming filters in the downlink. We will answer this question shortly. Before we answer this question, we would like to offer some discussions on the proposed coexistence scheme.

Spatial Degrees of Freedom (SDoF). The proposed coexistence scheme can be interpreted using the concept of SDoF in the information theory. For the AP with M antennas, it has M SDoF, each of which can be used to support one data stream transmission for either Wi-Fi or IoT device. For the network as shown in Fig. 3.2, the W Wi-Fi devices will consume W SDoF at the AP, and the K IoT devices will consume one SDoF at the AP. This is because the K IoT devices use different subcarriers for data transmission and therefore occupy only one spatial direction. To ensure that the AP has enough SDoF for multi-user detection in the uplink and beamforming in the downlink, we have the following constraints: $W + 1 \leq M$. It is worth pointing out that we assume the channels between the AP and the Wi-Fi/IoT devices have full rank. If the channels are deficient in rank, then the number of Wi-Fi/IoT devices that the AP can simultaneously serve will decrease correspondingly.

Heterogeneous versus Homogeneous MU-MIMO. The proposed coexistence scheme can be regarded as a heterogeneous MU-MIMO transmission where the users are Wi-Fi and IoT devices. Compared to homogeneous (conventional) MU-MIMO, heterogeneous MU-MIMO faces two challenges in the design of decoding filters in the uplink and beamforming filters in the downlink.

First, in the uplink transmission of homogeneous MU-MIMO, the user devices are typically well synchronized in both time and frequency domain. As a result, the uplink channel between the AP and each user device can be estimated at the AP, and the estimated channel can be used to decode the signals. However, in the uplink transmission of heterogeneous MU-MIMO (see Fig. 3.2(a)), it is very hard to achieve the time synchronization (at the level of 50 ns) among the Wi-Fi and IoT devices, because the IoT devices operate at a much lower clock frequency. As a consequence, the AP cannot estimate the uplink channels, which are needed for the conventional signal detection methods (e.g., ZF and MMSE). To address this challenge, we propose a channel-agnostic method for the signal detection. Unlike conventional signal detection methods that require CSI, our proposed method does not require CSI. Instead, it constructs the decoding filters for signal detection directly based on the corrupted reference signals.

Second, in the downlink transmission of heterogeneous MU-MIMO (see Fig. 3.2(b)), the acquisition of downlink channels for the design of beamforming filters is a costly task as it entails a large amount of airtime overhead, especially considering the MAC-level coordination for channel feedback among the Wi-Fi and IoT devices. To reduce the airtime overhead, we propose a lightweight beamforming method, which takes advantage of wireless channel reciprocity and directly uses the decoding filters in the uplink as the beamforming filters in the downlink.

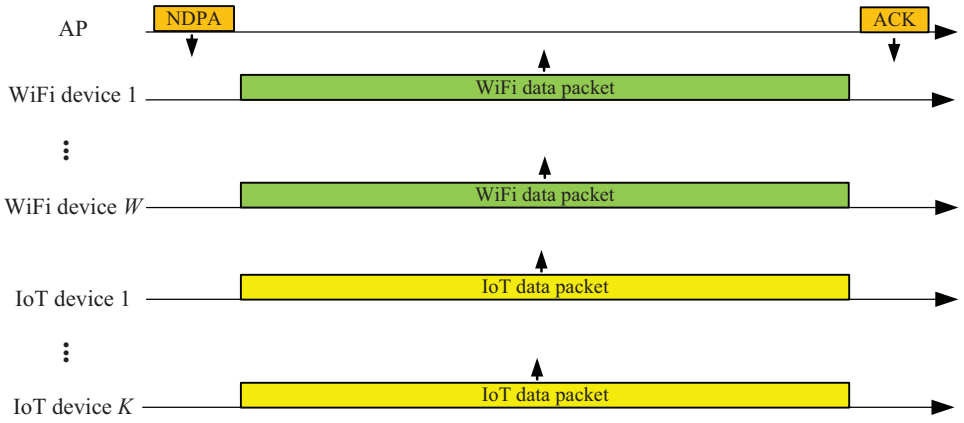


Figure 3.3: Uplink MAC protocol for data transmission of coexisting Wi-Fi and IoT devices.

3.2.2 Uplink Transmission

In this section, we first present a MAC protocol for uplink transmission, and then present the construction of decoding filters for the AP to decode the signals from the Wi-Fi and IoT devices, respectively.

Uplink MAC Protocol. Fig. 3.3 shows the proposed MAC protocol to enable the concurrent uplink transmissions for Wi-Fi and IoT devices. In this protocol, the AP first broadcasts an NDPA (null data packet announcement) frame to notify the Wi-Fi and IoT devices of the uplink data transmission. It contains the address of the AP and the selected devices. Upon receipt of the NDPA frame, the Wi-Fi and IoT devices send their data packets to the AP simultaneously. After the AP receives and decodes the uplink data packets, it responds with an ACK/NACK packet to inform each Wi-Fi/IoT device if its data packet has been successfully decoded. When an IoT device has no data packet for transmission, it will simply switch to the sleep mode to reduce its power consumption. The IoT device only needs to listen the beacon packets, which is broadcasted by the AP every 100 ms. Then, the AP can wake up an IoT device by setting commands in its next beacon packet. When the IoT device itself has data packets coming to its radio buffer for transmission, it will wake up and obtain the parameters for transmission by listening the beacon packets.

Decoding Wi-Fi Signal at AP. In the proposed protocol, the AP needs to decode the mixed signals from the Wi-Fi and IoT devices. To do so, we propose a heuristic signal detection method. In our method, the AP decodes the signal from each device separately. When decoding the signal from one device, it simply treats the signals from other devices as interference and constructs a spatial filter to cancel the interference and recover the desired signal by leveraging the reference signals embedded in each frame (packet). Mathematically, to decode the signal on subcarrier k from Wi-Fi device i, the AP constructs a spatial filter $\mathbf{G}_i(k) \in \mathbb{C}^{M\times 1}$ as follows:

$$\mathbf{G}_i(k) = \Big[\sum_{(l,k')\in\mathcal{R}_k} \mathbf{Y}(l,k')\mathbf{Y}(l,k')^{\mathsf{H}} \Big]^{+} \Big[\sum_{(l,k')\in\mathcal{R}_k} \mathbf{Y}(l,k')\bar{X}_i(l,k')^{\mathsf{H}} \Big], \tag{3.1}$$

where $\mathbf{Y}(l,k') \in \mathbb{C}^{M\times 1}$ is the AP's received frequency-domain signals in OFDM symbol l on subcarrier k', which includes the signals from all Wi-Fi and IoT devices. $\bar{X}_i(l,k')$, $(l,k') \in \mathcal{R}_k$, is the set of reference signals (e.g., L-STF and L-LTF [32]) in the frame that are used to construct subcarrier k's decoding filter. $(\cdot)^{\mathsf{H}}$ is conjugate transpose operator. $(\cdot)^{+}$ is pseudo-inverse operator. After constructing the decoding filter, the AP estimates the signals from Wi-Fi device i in the face of interference from other Wi-Fi/IoT devices by: $\hat{X}_i(l,k) = \mathbf{G}_i(k)^{\mathsf{H}}\mathbf{Y}(l,k)$, $\forall l,k$, where $\hat{X}_i(l,k)$ is the estimated signal from Wi-Fi device i.

Decoding IoT Signal at AP. A similar method has been used at the AP to decode the signal from each IoT device in the presence of interference from Wi-Fi devices. However, (3.1) cannot be directly used to decode IoT signals. This is because the IoT signal is not an OFDM signal, but a narrow-band signal. Hence, the AP's IoT Rx-PHY does not have FFT operations. It uses matched filters to convolute the signals from each IoT device (see Fig. 2.7). To decode the IoT signal, we make some minor changes in the construction of the decoding filter.

With a bit abuse of notation, we denote k as the index of IoT devices and l as the index of

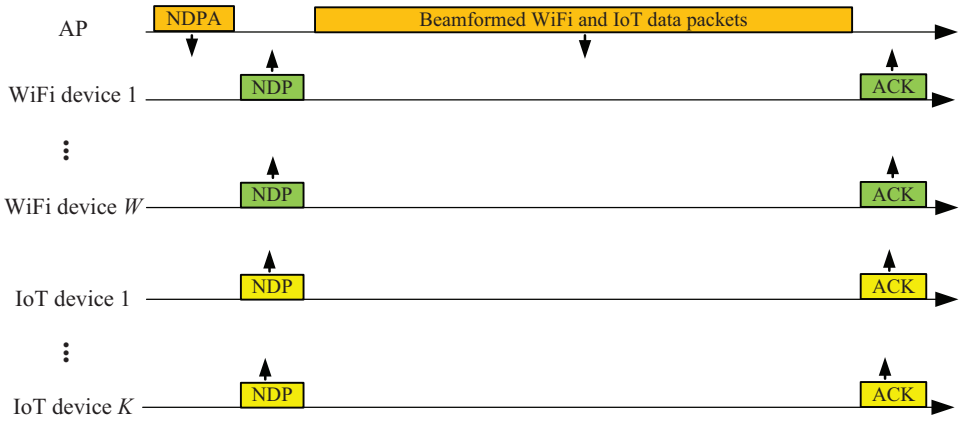


Figure 3.4: Downlink MAC protocol for data transmission of coexisting Wi-Fi and IoT devices.

data symbol from the IoT device. Denote $\mathbf{y}(l,k) \in \mathbb{C}^{M\times 1}$ as the AP's received baseband signal from its IoT-Rx PHY (i.e., the output signal of the matched filters on the left-hand side of Fig. 2.7). Denote $\bar{X}(l,k), l \in \mathcal{L}_{\text{ref}}$, as the set of reference signals in the IoT frame (i.e., the preamble of the IoT frame in Fig. 2.4). Denote $\mathbf{G}(k)$ as the spatial filter constructed to decode the signal from IoT device k. Then, we construct $\mathbf{G}(k) \in \mathbb{C}^{M\times 1}$ as follows:

$$\mathbf{G}(k) = \Big[\sum_{l\in\mathcal{L}_{\text{ref}}} \mathbf{y}(l,k)\mathbf{y}(l,k)^{\mathsf{H}}\Big]^{+}\Big[\sum_{l\in\mathcal{L}_{\text{ref}}} \mathbf{y}(l,k)\bar{X}(l,k)^{\mathsf{H}}\Big]. \tag{3.2}$$

After constructing the decoding filter, the AP estimates the signals from IoT device k by: $\hat{X}(l,k) = \mathbf{G}(k)^{\mathsf{H}}\mathbf{y}(l,k), \forall l,k$.

3.2.3 Downlink Transmission

Similar to the previous section, we first propose a MAC protocol for downlink transmission and then present the construction procedure of beamforming filters for the AP.

Downlink MAC Protocol. Fig. 3.4 shows the proposed MAC protocol enabling the concurrent downlink transmissions for Wi-Fi and IoT devices. The protocol has the following steps: (i) The

AP first broadcasts an NDPA packet to inform the Wi-Fi and IoT devices of downlink transmission. It contains the address of the AP and selected Wi-Fi and IoT devices. (ii) Upon receipt of the NDPA packet, each of the Wi-Fi and IoT devices responds with an NDP packet immediately. This legacy NDP packet serves two purposes: confirming participation of a device in this round of transmission and providing reference signals for the AP to construct beamforming filters. (iii) Using the constructed beamforming filters, the AP sends packets to all Wi-Fi/IoT devices. (iv) After receiving the data packets, each device sends an ACK/NACK packet to the AP.

Beamforming at AP. In this protocol, we need to figure out how to construct the precoding filters for beamforming at the AP so that each Wi-Fi/IoT device can successfully decode its desired signal. We take advantage of wireless channel reciprocity by directly using the decoding filters in the uplink as the beamforming filters in the downlink. Given that the uplink and downlink channels are reciprocal, if a set of spatial filters can support interference-free data transmission in the uplink, they can also support interference-free data transmission in the downlink.

Guided by this idea, we construct the beamforming filters as follows: First, the AP constructs the decoding filters for the Wi-Fi devices using (3.1) and for the IoT devices using (3.2) by leveraging the reference signals in the uplink NDP packets as shown in Fig. 3.4. Then, it directly uses constructed decoding filters to precode the downlink signals for both Wi-Fi and IoT devices. Mathematically, the AP precodes its downlink signals as follows:

$$\mathbf{S}(l,k) = \sum_{i=1}^{W} \mathbf{G}_i(k)^* S_i(l,k) + \mathbf{G}(k)^* S(l,k), \tag{3.3}$$

where $S_i(l,k)$ is the data that the AP wants to send to Wi-Fi device i on subcarrier k in OFDM symbol l; $S(l,k)$ is the data that the AP wants to send to IoT device k in data symbol l; $\mathbf{S}(l,k)$ is the precoded baseband signal vector that the AP sends to its M antenna ports on subcarrier k in

OFDM symbol l; $\mathbf{G}_i(k)$ and $\mathbf{G}(k)$ are calculated at the AP using (3.1) and (3.2) by leveraging the uplink sounding signals in the protocol.

Channel Calibration. In real systems, although over-the-air channels are reciprocal, the Tx and Rx RF circuits are not. To maintain the reciprocity of compound uplink and downlink channels, we employ the relative calibration method in [38]. This relative calibration method is an internal and standalone calibration method that can be done at the AP without requiring involvement of user devices. In our experiments, we implement this calibration method to maintain the channel reciprocity.

3.2.4 Discussions

The heterogeneous MU-MIMO uplink and downlink protocols, along with the proposed signal detection and beamforming methods, constitute the coexistence scheme that enables the Wi-Fi and IoT devices to share the spectrum simultaneously. We have the following remarks on the proposed coexistence scheme.

Remark 1. For both the signal detection method in the uplink and the beamforming method in the downlink, it is hard to analytically quantify their performance. Therefore, we resort to experiments to show their performance in real-world wireless environments.

Remark 2. In the proposed coexistence scheme, neither the signal detection nor the beamforming method requires CSI. Instead, they use the reference signals in the frame to construct the decoding/beamforming filters directly. As the channel estimation in conventional wireless networks typically incurs a large amount of airtime overhead, the removal of channel estimation in the proposed coexistence scheme not only improves the spectral efficiency but also reduces the implementation complexity.

Remark 3. The two MAC protocols are lightweight. The signal detection and beamforming

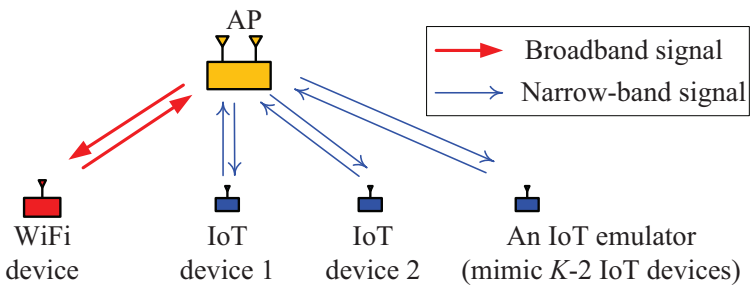


Figure 3.5: A prototyped WiFi-IoT system that comprises a two-antenna AP, a Wi-Fi device, two IoT devices, and an IoT emulator. The IoT emulator is used to mimic $(K-2)$ IoT devices for ease of implementation, $K \in \{12, 16, 24\}$.

methods have a low computational complexity. Therefore, the proposed coexistence scheme is amenable to practical implementation.

3.3 Experimental Evaluation

In this section, we conduct experiments in real-world wireless environments to evaluate the performance of WiFi-IoT in both uplink and downlink data transmissions.

3.3.1 System Implementation

We have built a prototype of WiFi-IoT in the network as shown in Fig. 3.5, which comprises an AP, a Wi-Fi device, two independent IoT devices, and an IoT emulator. The AP has two antennas, and the Wi-Fi/IoT devices have a single antenna. The IoT emulator is used to mimic $(K-2)$ IoT devices when $K \in \{12, 16, 24\}$. The purpose of this device is to reduce the experimental complexity. The system has been built on a software-defined radio (SDR) wireless testbed that consists of USRP2 [39] and GNURadio software package [40]. C++ language has been used to implement all the signal processing and protocol modules in GNURadio.

PHY Implementation. The communications between the AP and the Wi-Fi device are con-

ducted using the legacy IEEE 802.11 frame [32]. The baseband signal processing chains required for packet transmission and reception have been implemented on both the AP and the Wi-Fi device.

The communications between the AP and the IoT devices are conducted using the IoT frame in Fig. 2.4, with $M_p = 12$, $M_s = 50$, and $M_d = 50$. The total number of symbols in the IoT frame is set to 1044. The proposed baseband signal processing chains in Fig. 2.5 and Fig. 2.7 have been implemented on the AP and each IoT device for data packet transmission and reception.

At the AP, its sampling rate is set to 20 Msps, and its transmit power is set to 20 dBm. At the Wi-Fi device, its sampling rate is also set to 20 Msps, and its transmit power is set to 20 dBm as well. At the IoT devices, we use $4\times$ oversampling factor and thus set their sampling rate to 1 Msps. Since an IoT device uses a single subcarrier for packet transmission, we scale down its transmit power to $20 - 10\log_{10}(52/1) \approx 3$, where 20 (dBm) is Wi-Fi device's transmit power, 52 is the number of valid subcarriers used by Wi-Fi devices, and 3 (dBm) is an IoT device's transmit power. With this transmit power, an IoT device has a similar communication range as a Wi-Fi device.

MAC Implementation. We have implemented the proposed coexistence scheme on this WiFi-IoT system. Specifically, we have implemented the MAC protocol in Fig. 3.3 as well as the proposed heterogeneous MU-MIMO detection algorithms to enable uplink data transmissions. We have also implemented the MAC protocol in Fig. 3.4 and the proposed beamforming algorithm as well as the relative channel calibration method in [38] to enable downlink data transmissions.

3.3.2 Experimental Setup and Performance Metrics

Experimental Setup. We measure the performance of the Wi-Fi and IoT communications in an office building as shown in Fig. 3.6. The AP is placed at the spot marked by "AP". Around the AP, we randomly picked up 32 locations and divided them into 8 groups. Each group has 4 locations (marked by the same symbol in Fig. 3.6), at which we placed the Wi-Fi device and the three IoT

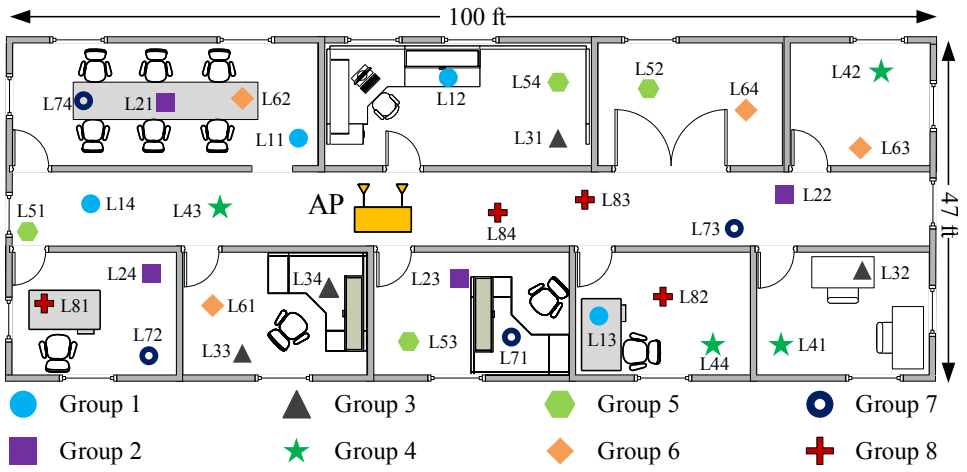


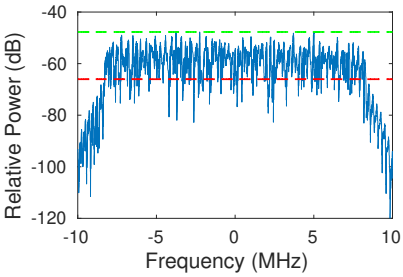
Figure 3.6: The floor plan for performance evaluation.

devices. Specifically, at location index i ($1 \leq i \leq 8$), the Wi-Fi device is placed at L_{i4}; the two IoT devices are placed at L_{i1} and L_{i2}; and the IoT emulator is placed at L_{i3}. In our experiments, we use the IoT device at L_{i1} as the representative when presenting the performance of IoT communications.

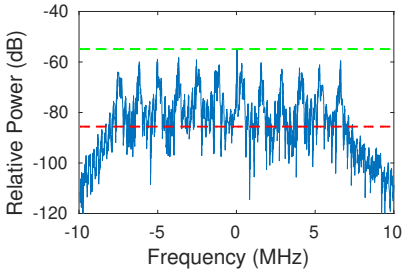
Performance Metrics. We use EVM and data rate, as defined in Section 2.9.1 (equations (2.18) and (2.19)), to quantify the performance of the proposed WiFi-IoT solution. Specifically, for the Wi-Fi device, its uplink/downlink data rate is calculated by: $r = \frac{48}{80} \times 20 \times \gamma(\text{EVM})$ Mbps, where 48 is the number of payload subcarriers, 80 is the points of one OFDM symbol (including CP), 20 is the Wi-Fi signal sampling rate (in Msps), and $\gamma(\text{EVM})$ is the average number of bits carried by one symbol and its possible values are given in Table 2.1. For the IoT device, its uplink/downlink data rate is calculated by: $r = 250 \times \gamma(\text{EVM})$ kbps, where 250 is the IoT signal baud rate (in ksps) and $\gamma(\text{EVM})$ is given in Table 2.1.

3.3.3 A Case Study

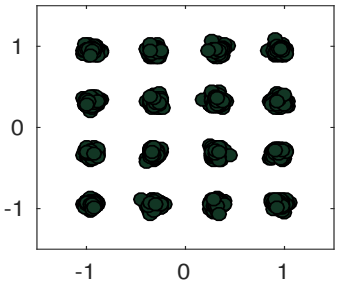
Before presenting the complete results, we first use a case study to examine the details of the proposed WiFi-IoT solution. In this case study, we placed the Wi-Fi and IoT devices at location



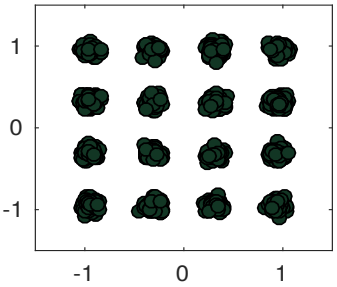
(a) Relative power spectral density of the received Wi-Fi signals at AP.



(b) Relative power spectral density of the received IoT signals at AP.



(c) Decoded Wi-Fi signals at AP.



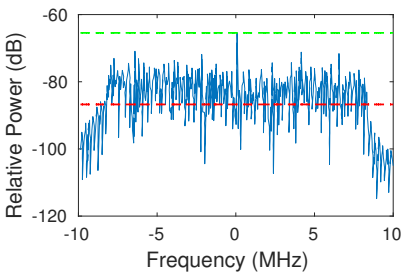
(d) Decoded IoT signals at AP.

Figure 3.7: Uplink performance: (a-b) shows relative power spectral density of the Wi-Fi and IoT signals received by the AP's first antenna; (c-d) shows the constellation of decoded Wi-Fi and IoT signals at the AP when $K = 12$.

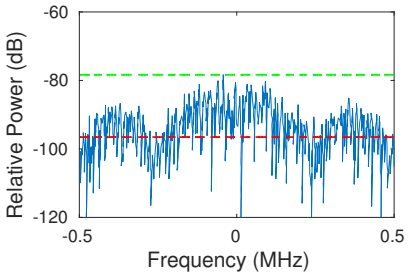
index 1 (i.e., L_{11}, L_{12}, L_{13}, and L_{14}), and set the number of IoT devices to 12 (i.e., $K = 12$).

Uplink Results. In the uplink, the Wi-Fi and IoT devices send their packets to the AP simultaneously. The AP needs to decode both Wi-Fi and IoT signals. It is interesting to see the received power spectral density of the received Wi-Fi and IoT signals at the AP. Fig. 3.7(a-b) shows our experimental results. We can see that the received Wi-Fi signals have relatively flat spectrum, whereas the received IoT signals have 12 spectral peaks. Each spectral peak corresponds to one IoT device's signal. The bandwidth of the signal from one IoT device is about 250 kHz.

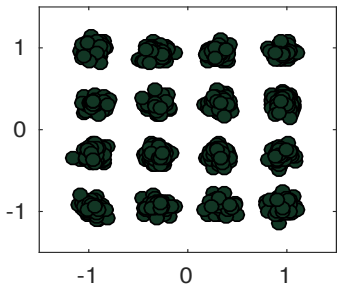
It is also interesting to see if the AP can successfully decode the concurrent Wi-Fi and IoT signals. Fig. 3.7(c) shows the constellation of the decoded Wi-Fi signals; Fig. 3.7(d) shows the constellation of the decoded IoT signals. It is evident that, using our proposed signal detection method, the AP can successfully decode both Wi-Fi and IoT signals. More specifically, our



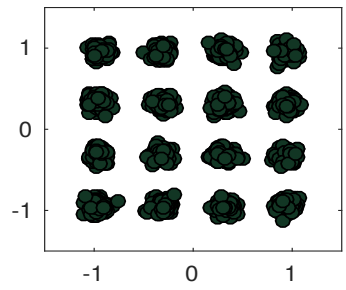
(a) Relative power spectral density of received signals at the Wi-Fi device.



(b) Relative power spectral density of received signals at one IoT device.



(c) Decoded signals at the Wi-Fi device.



(d) Decoded signals at one IoT device.

Figure 3.8: Downlink performance: (a-b) shows relative power spectral density of the received signals at the Wi-Fi device and one IoT device; (c-d) shows the constellation of decoded signals at the Wi-Fi device and one IoT device when $K = 12$.

experimental results show that the EVM of the decoded Wi-Fi and IoT signals are -25.6 dB and -23.9 dB, respectively. This indicates that the AP can successfully serve the heterogeneous devices (one broadband Wi-Fi device and 12 narrow-band IoT devices) simultaneously.

Downlink Results. In the downlink, the AP performs beamforming to send data packets to the Wi-Fi and IoT devices simultaneously. Similar to the uplink, we examine the power spectral density and the decoded signals on the receiver side (the Wi-Fi and IoT devices) to see if the AP can successfully serve both Wi-Fi and IoT devices simultaneously.

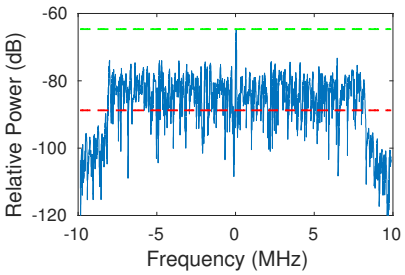
Fig. 3.8(a) shows the power spectral density of the received signals at the Wi-Fi device; Fig. 3.8(b) shows the power spectral density of the received signals at one IoT device. Fig. 3.8(c) shows the constellation of the decoded signals at the Wi-Fi device and Fig. 3.8(d) shows the constellation of the decoded signals at one IoT device. It is evident that both devices can successfully

decode their desired signals. The measured EVM is -21.5 dB at the Wi-Fi device and -22.0 dB at the IoT device.

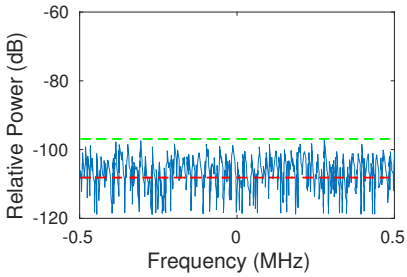
Scrutinizing Beamforming in Downlink. Since the proposed beamforming method plays a critical role in the downlink transmission, we would like to further scrutinize the experimental results to examine its performance. Specifically, we would like to see the effectiveness of the proposed beamforming filters in the mitigation of inter-user interference.

We first examine the effectiveness of Wi-Fi beamforming filter (3.1). To do so, we first let the AP only transmit Wi-Fi signal (by setting the IoT signal to zero), and then observe the received signal at the Wi-Fi and IoT devices when two different beamforming filters are used. Fig. 3.9 exhibits our experimental results. By comparing Fig. 3.9(a) with Fig. 3.9(c), we can see that the Wi-Fi device can receive Wi-Fi signals with similar strength when the AP uses those two beamforming filters. By comparing Fig. 3.9(b) with Fig. 3.9(d), we can see that the proposed Wi-Fi beamforming filter can effectively mitigate the inter-user interference for the IoT devices. Compared to the equal-power beamforming, our proposed beamforming filters have more than 20 dB cancellation capability for inter-user interference.

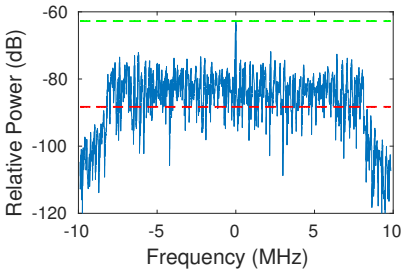
We now examine the effectiveness of IoT beamforming filter (3.2) in the downlink. Similarly, we first let the AP only transmit IoT signal (by setting the Wi-Fi signal to zero), and then observe the received signal at the Wi-Fi and IoT devices when two different beamforming filters are used. Fig. 3.10 exhibits our experimental results. By comparing Fig. 3.10(b) with Fig. 3.10(d), we can see that the proposed IoT beamforming filters can effectively mitigate the inter-user interference for the Wi-Fi device. It has more than 15 dB cancellation capability for inter-user interference.



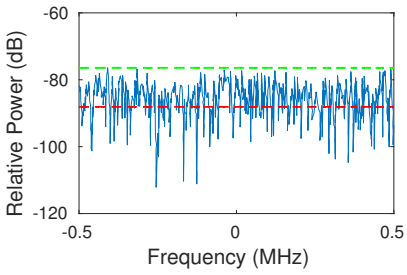
(a) Received signal at the Wi-Fi device when the AP uses beamforming filter (3.1).



(b) Received signal at the IoT device when the AP uses beamforming filter (3.1).



(c) Received signal at the Wi-Fi device when the AP uses beamforming filter $[\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}]^T$.



(d) Received signal at the IoT device when the AP uses beamforming filter $[\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}]^T$.

Figure 3.9: Relative power spectral density of the received signals at the Wi-Fi and IoT devices in the downlink when the AP sends Wi-Fi signals only.

3.3.4 Complete Experimental Results

We now present all the measured experimental results from the 8 different locations in Fig. 3.6. Again, we use one IoT device (the one placed at L_{i1}, $1 \leq i \leq 8$) as the representative when presenting the performance of IoT communication.

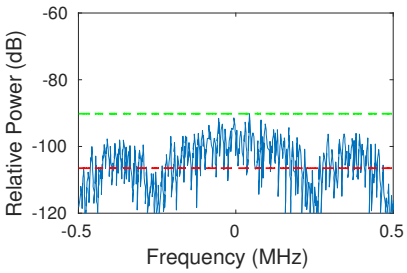
Uplink Results. We collect the experimental data at the AP during the uplink communications. Fig. 3.11(a-b) presents the measured EVM of the decoded Wi-Fi and IoT signals at the AP. We can see that the EVM of the decoded Wi-Fi signals is less than -25.7 dB when there is no IoT device in the network ($K = 0$), less than -24.4 dB when $K = 1$, less than -22.0 dB when $K = 12$, less than -21.6 dB when $K = 16$, and less than -19.8 dB when $K = 24$. The EVM of the decoded IoT signals is less than -21.3 dB when $K = 1$, less than -18.6 dB when $K = 12$, less than -18.3 dB when $K = 16$, and less than -15.0 dB when $K = 24$. Meanwhile, we can see that the EVM

of the decoded Wi-Fi and IoT signals slightly degrades as K increases. This is mainly because the inter-subcarrier interference becomes more significant as K increases.

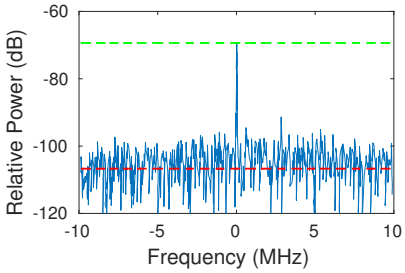
Fig. 3.11(c) presents the extrapolated data rate of the uplink Wi-Fi communication. The achievable uplink data rate at the Wi-Fi device is greater than 54 Mbps when $K = 0$, greater than 48 Mbps when $K = 1$, greater than 48 Mbps when $K = 12$, greater than 36 Mbps when $K = 16$, and greater than 36 Mbps when $K = 24$. The increase of IoT devices slightly degrades the data rate of Wi-Fi communications. This is what we expected, and the degradation is attributed to the interference from the IoT devices. Fig. 3.11(d) presents the extrapolated data rate of the uplink IoT communication. The achievable uplink data rate at one IoT device is greater than 750 kbps when $K = 1$, greater than 500 kbps when $K = 12$, greater than 500 kbps when $K = 16$, and greater than 375 kbps when $K = 24$. The extrapolated data rate is more than sufficient for most existing and future IoT applications.

Downlink Results. We present the experimental data collected at the Wi-Fi and IoT devices during the downlink communications. Fig. 3.12(a-b) presents the measured EVM of the decoded signals at the Wi-Fi and IoT devices. We can see that the EVM of the decoded Wi-Fi signals is less than -25.2 dB when $K = 0$, less than -22.9 dB when $K = 1$, less than -18.6 dB when $K = 12$, less than -17.3 dB when $K = 16$, and less than -15.4 dB when $K = 24$. The EVM of the decoded IoT signals is less than -22.4 dB when $K = 1$, less than -19.3 dB when $K = 12$, less than -18.2 dB when $K = 16$, and less than -14.6 dB when $K = 24$. Similar to the uplink, the EVM measured in the downlink slightly degrades as K increases. This is also because the interference leakage becomes more significant as K increases.

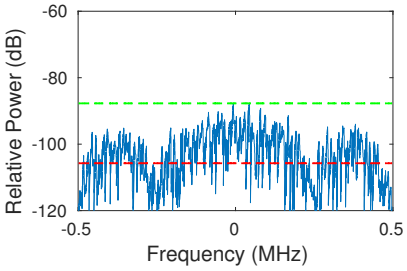
Fig. 3.12(c) presents the extrapolated data rate of the uplink Wi-Fi communication. The achievable downlink data rate at the Wi-Fi device is greater than 54 Mbps when $K = 0$, greater than 48 Mbps when $K = 1$, greater than 24 Mbps when $K = 12$, greater than 24 Mbps when



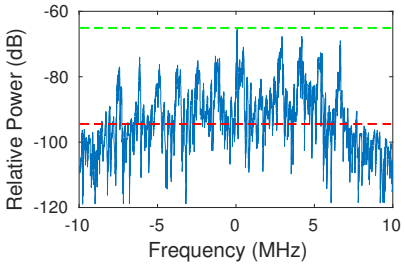
(a) Received signal at the IoT device when the AP uses beamforming filter (3.2).



(b) Received signal at the Wi-Fi device when the AP uses beamforming filter (3.2).



(c) Received signal at the IoT device when the AP uses beamforming filter $[\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}}]^T$.



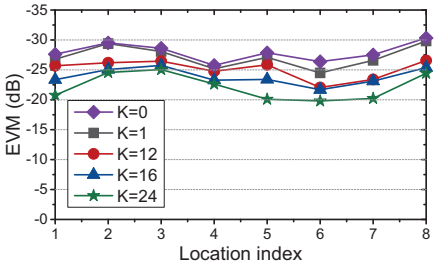
(d) Received signal at the Wi-Fi device when the AP uses beamforming filter $[\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}}]^T$.

Figure 3.10: Relative power spectral density of the received signals at the Wi-Fi and IoT devices in the downlink when the AP only sends IoT signals.

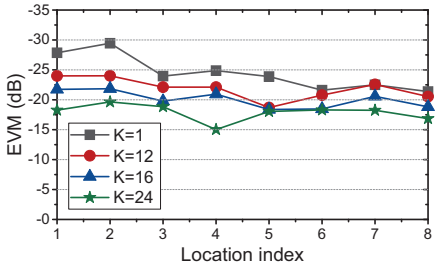
$K = 16$, and greater than 18 Mbps when $K = 24$. Fig. 3.12(d) presents the extrapolated data rate of the uplink IoT communication. The achievable downlink data rate at one IoT device is greater than 1000 kbps when $K = 1$, greater than 750 kbps when $K = 12$, greater than 500 kbps when $K = 16$, and greater than 375 kbps when $K = 24$. The achieved data rate for Wi-Fi and IoT communications meets the requirements of most Wi-Fi and IoT applications.

3.3.5 Observations

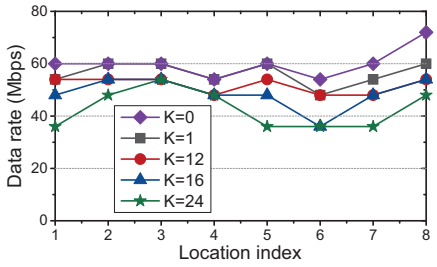
Based on the above experimental results, we have the following observations: First, a non-OFDM IoT device can communicate with an OFDM-based AP using a low sampling rate (250 ksps or 1 Msps). Second, in a typical office building, an AP with two antennas can serve one Wi-Fi device and 24 IoT devices simultaneously in both downlink and uplink. Third, the Wi-Fi device can



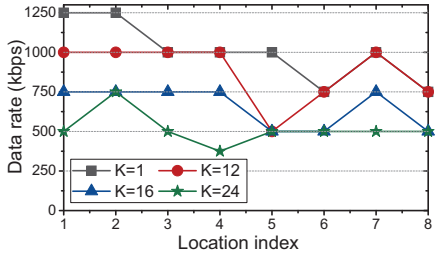
(a) Measured EVM of decoded Wi-Fi signal at the AP.



(b) Measured EVM of decoded IoT signal at the AP.



(c) Extrapolated data rate of uplink Wi-Fi communication.



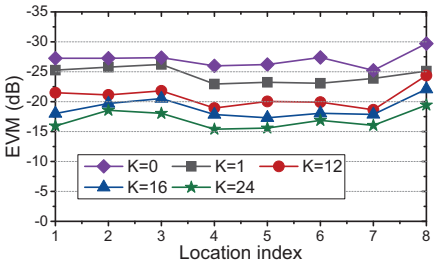
(d) Extrapolated data rate of uplink IoT communication.

Figure 3.11: Measured EVM and extrapolated data rate in the uplink communication.

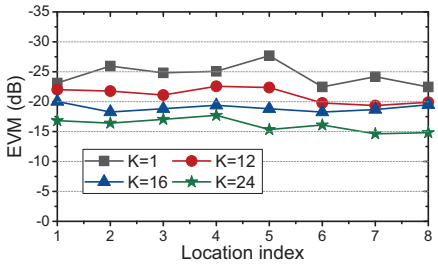
achieve more than 36 Mbps in the uplink and more than 24 Mbps in the downlink. Fourth, the IoT device can achieve more than 375 kbps in both uplink and downlink.

3.4 Chapter Summary

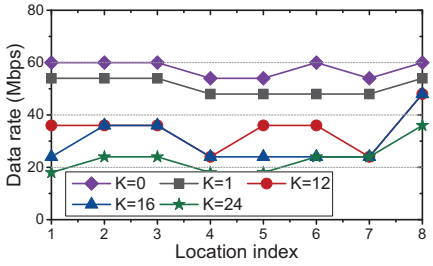
In this chapter, we proposed WiFi-IoT, an energy-efficient IoT communication solution for future Wi-Fi networks. WiFi-IoT features two innovative techniques. The first one is an asymmetric PHY design, which allows an OFDM-based AP to communicate with multiple QAM-based (non-OFDM) IoT devices simultaneously. Such an asymmetric PHY makes it possible for the IoT devices to transmit/receive their signals at a low sampling rate (250 ksps), thereby conserving their power consumption for radio communications. The second one is a transparent coexistence scheme, which enables an AP with multiple antennas to serve broadband Wi-Fi devices and narrow-band IoT devices simultaneously. We have built a prototype of WiFi-IoT on a wireless testbed and



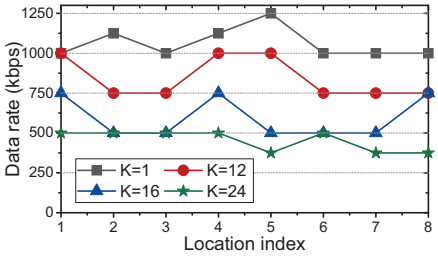
(a) Measured EVM of decoded Wi-Fi signal at the Wi-Fi device.



(b) Measured EVM of decoded IoT signal at the IoT device.



(c) Extrapolated data rate of downlink Wi-Fi communication.



(d) Extrapolated data rate of downlink IoT communication.

Figure 3.12: Measured EVM and extrapolated data rate in the downlink communication.

evaluated its performance in real-world wireless environments. Experimental results show that, using the proposed WiFi-IoT solution, a two-antenna AP can communicate with one legacy Wi-Fi device and 24 narrow-band IoT devices simultaneously in both uplink and downlink.

Chapter 4

Uplink Distributed MIMO

WLANs are a key component of the telecommunications infrastructure in our society. While many solutions have been produced to improve their downlink throughput, the techniques for enhancing their uplink throughput remain limited. The stagnation can be attributed to the lack of fine-grained inter-node synchronization due to the hardware limitation of most devices. In this chapter, we present an uplink distributed MIMO scheme (termed UD-MIMO) for WLANs to enable concurrent uplink transmissions in the absence of fine-grained inter-node synchronization. The enabling technique behind UD-MIMO is a practical solution to decoding uplink packets from asynchronous users. UD-MIMO makes it possible for WLANs to significantly improve their uplink throughput while not requiring tight inter-node synchronization. We have built a prototype of UD-MIMO on a wireless testbed and demonstrate its compatibility with commercial off-the-shelf Atheros 802.11 client devices (with modified Linux driver). Our experimental results show that, for a WLAN with 8 APs in a conference room, UD-MIMO offers $3.4\times$ throughput compared to interference-avoidance approach.

4.1 Introduction

The proliferation of wireless devices under the driving forces from emerging concepts such as smart cities, intelligent transportation systems, and the Internet of Things has led to unprecedented demands for wireless services. Cisco predicts that the wireless demands would double in the

next two years and reach 120 exabytes per month by 2021 [41]. As a key component of the telecommunications infrastructure in our society, WLANs carry even more data traffic for mobile devices than cellular networks. The predicament facing WLANs is that the increase of their capacity cannot catch up the growth of wireless demands. Such a predicament becomes particularly daunting in dense wireless environments such as conference rooms, football stadiums, cinemas, and airports.

A straightforward idea to increase the capacity of WLANs is to deploy more APs to enrich the service resources for users. This approach, however, does not work in dense wireless environments. The capacity of existing WLANs does not scale with the number of APs. This is because the existing WLANs use CSMA protocol to manage the interference. Such an interference-avoidance protocol only allows one AP to access the spectrum in a collision domain, no matter how many APs are deployed in this area. Another idea to increase the capacity of WLANs is to enhance AP's capability [42]. Given the advancement of MIMO technology in the past decades [43], it is common nowadays that a commercial AP (Wi-Fi router) is equipped with multiple antennas. However, the advancement of individual AP cannot fundamentally solve the network capacity problem because the number of antennas on an AP is limited by its physical size.

Distributed MIMO has been widely regarded as a promising technique to improve the capacity of WLANs. Given the fact that APs are connected via high-speed Ethernet cables in some scenarios, all the APs can jointly process the signals from/to multiple users. With a proper design, the APs can serve many users simultaneously instead of being limited by their co-channel interference. Consider the WLAN in Fig. 4.1 for example. If the network uses CSMA-based interference-avoidance technique, only one AP can access the spectrum at a time. In contrast, if the APs are jointly processing the signals, then the WLAN resembles a 4×4 MU-MIMO system, making it possible for the APs to serve the four users simultaneously.

While the throughput gain of distributed MIMO is attractive, the realization of distributed

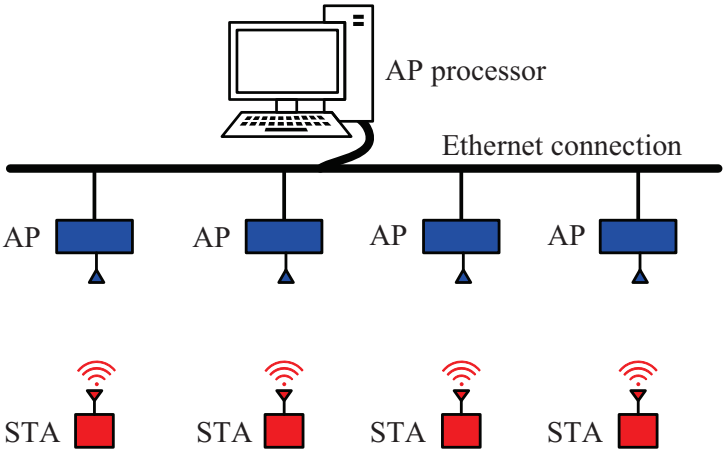


Figure 4.1: Illustrating distributed MIMO in a WLAN.

MIMO in practical WLANs is challenging. The challenge lies in the clock/time synchronization among the network devices (AP and STAs). Despite the APs being connected via Ethernet cables, those connections are suitable for data packet transmission, but not suitable for clock/time synchronization. As such, synchronizing the network devices for distributed MIMO is not a trivial problem. Although some system papers have studied the synchronization issues to enable distributed MIMO for WLANs, most of them are focused on downlink transmission (see, e.g., [44–47]). Very limited progress has been made so far in the design of a practical distributed MIMO scheme for concurrent uplink transmission. One may argue that most traffic in WLANs is carried by downlink, and thus the uplink capacity is not demanding. This may not be true in the next decades, given the increasing popularity of cloud-based applications that require frequent data transmission from user devices to cloud [48].

In this chapter, we present UD-MIMO, an uplink distributed MIMO scheme for WLANs. We consider a WLAN as shown in Fig. 4.1 and focus on the scenario of busy wireless environments such as conference rooms, enterprises, hotels, shopping malls, and airports. We assume that multiple users send their packets to the APs simultaneously. Upon reception of the mixed signals, the APs send the received signals to an AP processor via Ethernet connection, which typically has high

speed and low latency. At the AP processor, a signal detection technique is employed to decode the data packets. In such a network, if the devices are perfectly synchronized, then UD-MIMO would be identical to MU-MIMO, and conventional multi-user detection (MUD) methods such as ZF and MMSE would be able to decode signals at the AP processor. But in reality, the network devices are driven by independent oscillators. Consequently, they are neither time-aligned nor frequency-synchronized, making the signal detection problem particularly challenging.

One natural approach to solving the signal detection problem is by designing a sophisticated protocol to synchronize the network devices. This approach, however, has two issues. First, the time synchronization among stations (STAs, a.k.a. user devices) is not easy to achieve. In order for the APs to decode the packets, the time misalignment of STAs' transmissions should be less than the CP of an OFDM symbol, which is 800 ns in 802.11 networks. Given STAs' mobility, achieving such a fine-grained time synchronization among all the STAs will incur a large amount of airtime overhead. This issue was reflected by IEEE 802.11ac standard [32], which supports downlink MU-MIMO but does not supports uplink MU-MIMO. Second, the time and frequency synchronizations of STA-side transmissions require hardware modification of the user devices. Doing so will make UD-MIMO not compatible with already-existing 802.11 devices. For these two reasons, synchronizing the STAs for uplink transmissions is not a good approach to pursue.

We, therefore, explore an alternative approach: Instead of synchronizing the STAs, we live with their asynchrony and tackle the issue on the AP sides. Specifically, we develop a new MUD method that can decode the asynchronous data packets from multiple STAs. Through sophisticated signal processing functions, the new MUD method can decode the data packet from each STA by treating the packets from other STAs as interference. As such, it does not require synchronization among the STAs. This new MUD method not only removes the need for hardware modification of user devices, it also eliminates the huge airtime overhead induced by synchronization protocols.

We have built a prototype of UD-MIMO and evaluated its performance on two wireless testbeds: (i) The APs are custom-built using USRP devices, and the STAs are commercial Atheros 802.11 dongles with modified drivers. (ii) Both APs and STAs are custom-built using USRP devices. Based on our experimental results, we have the following observations: (i) UD-MIMO is compatible with commercial off-the-shelf Atheros 802.11 devices (with modified Linux driver). (ii) For a WLAN with 8 APs deployed in a conference room, UD-MIMO offers 3.4$\times$ uplink throughput compared to CSMA-based interference-avoidance approach. Meanwhile, UD-MIMO achieves more than 82% throughput of MU-MIMO, where all the APs and STAs are perfectly synchronized via external clocks.

4.2 Related Work

Synchronization in Distributed MIMO. [44–46] are the most relevant papers to this work. In [44], a scheme called JMB (or MegaMIMO) was proposed to enable downlink distributed MIMO in WLANs. Its main efforts focus on realizing phase and time synchronizations among independent APs so that a joint beamforming technique can be used to enable downlink MU-MIMO transmission. A similar idea called Airsync was proposed in [45] to address timing and carrier phase synchronizations for distributed downlink MU-MIMO transmission. One may wonder if the schemes proposed in [44] and [45] can be used to enable UD-MIMO as well. Actually, it cannot. Because doing so will require hardware modification of 802.11 client devices. In contrast, UD-MIMO not only maintains compatibility with 802.11 devices but also reduces the overhead (see Fig. 4.2 in this chapter and Fig. 3 in [44]).

In [46], a layering protocol called Chorus was proposed to achieve network-wide clock and time synchronization for LTE systems. However, Chorus relies on extra radio resource blocks and

new hardware to update frequency shift and phase errors. It is therefore considered an expensive solution. Apparently, UD-MIMO takes a completely different approach.

Synchronization in Wireless Networks. A large body of work (see, e.g., [49–54]) studied time and frequency synchronizations in wireless networks. For example, [49] proposed a distributed architecture called SourceSync to exploit the diversity of transmitters. Particularly, a specific protocol was proposed to meet the requirements of time synchronization on the transmitter side. Since SourceSync was dedicatedly devised for exploiting diversity, it cannot apply to distributed MIMO for spatial multiplexing. [50] analyzed the time and frequency synchronizations in large-sized dense wireless networks. However, these results cannot directly be applied to distributed MIMO systems, either because they are limited to theoretical analysis or because they entail an overwhelmingly large amount of overhead.

Performance of Distributed MIMO. [55] presented Signpost, a scalable MU-MIMO scheme without CSI feedback. [47] presented NEMOx, a hierarchical network architecture to achieve the scalability of distributed MIMO. [56] studied the performance of different precoding techniques in downlink distributed MIMO systems. However, these efforts focused on the practical realization of distributed MIMO but did not take into account the synchronization issues. Our work is orthogonal to this research line and complements these efforts.

4.3 UD-MIMO: An Uplink Distributed MIMO Scheme

We consider a dense WLAN as shown in Fig. 4.1, which comprises M single-antenna APs, N single-antenna STAs, and an AP processor. Such a network could be a Wi-Fi network deployed in conference rooms, shopping malls, or airports. For this network, we have the following assumptions: (i) The APs are connected via a high-speed wired connection, which is only good for exchange

data packets but not suitable for clock synchronization. This is true in reality. (ii) The STAs in the network could be incumbent 802.11a/g/n/ac user devices. While their software (firmware and driver) can be upgraded, their hardware (e.g., PLL circuit and baseband signal processing at the PHY layer) cannot be upgraded.

The objective of our design is to enable concurrent uplink transmissions in such a WLAN while preserving its compatibility with incumbent 802.11 client devices (STAs). As the performance of conventional WLANs is limited by co-channel interference, the success of UD-MIMO will significantly improve the network uplink throughput. For ease of exposition, we consider the network where each device has a single antenna. In the end, we shall see that UD-MIMO can also apply to the networks where devices have multiple antennas.

4.3.1 Our Approach

Given that the difference between UD-MIMO and point-to-point MIMO lies in the synchronizations, an intuitive approach to enable UD-MIMO is by designing a sophisticated mechanism to achieve the necessary synchronizations on both AP and STA sides. This approach, however, cannot maintain backward compatibility with existing 802.11 devices (STAs). This is because synchronizing the STAs requires the modification of their hardware. The estimation and compensation of carrier frequency offsets can only be done through baseband signal processing modules, which are hard-coded in ASIC chips and cannot be modified by upgrading the firmware or driver. Hence, synchronization operations cannot be conducted by existing 802.11 devices, and such an approach cannot maintain the backward compatibility of the network infrastructure. We propose a new approach for UD-MIMO. In our approach, the APs take full responsibility for addressing the synchronization issues, and the STAs do not need to perform any synchronization operations.

4.3.2 UD-MIMO Protocol

Fig. 4.2 shows the protocol for UD-MIMO transmission. It comprises the following three steps:

- **Step 1: Trigger frame broadcast.** The lead AP, which is designated by the AP processor, broadcasts a trigger frame (packet). This packet serves the following two purposes. i) *Announcing UD-MIMO transmission:* The trigger packet includes the addresses of the slave APs and STAs that are involved in this UD-MIMO. Upon reception of this packet, the slave APs and the STAs are notified of their participation in the UD-MIMO transmission. ii) *Providing reference packet for the slave APs to estimate their carrier frequency offsets:* Based on the received trigger frame from the lead AP, each slave AP estimates the carrier frequency offset between itself and the lead AP. The estimated carrier frequency offset is recorded at the slave AP and will be used in Step 2.

- **Step 2: Uplink data transmission.** Upon reception of the trigger frame, the STAs prepare their data packets and send radio signals into the air simultaneously. More specifically, each STA uses an aggregate frame format in Fig. 4.3 for the uplink data transmission. Note that, since the STAs operate independently, their transmissions will not exactly start at the same time. A time misalignment may exist, as illustrated in Fig. 4.2. On the AP side, each AP receives mixed radio signals from the STAs. Each slave AP compensates the carrier frequency offset between itself and the lead AP using the frequency offset value estimated in Step 1. Then, all the APs send their signal streams to the AP processor.

- **Step 3: Acknowledgment.** Upon the decoding results, the lead AP broadcasts an ACK/NACK packet to the STAs. The ACK/NACK packet has the information of which packets from which STAs were not successfully decoded. Based on this information, each STA prepares a re-transmission, if necessary, in the next round.

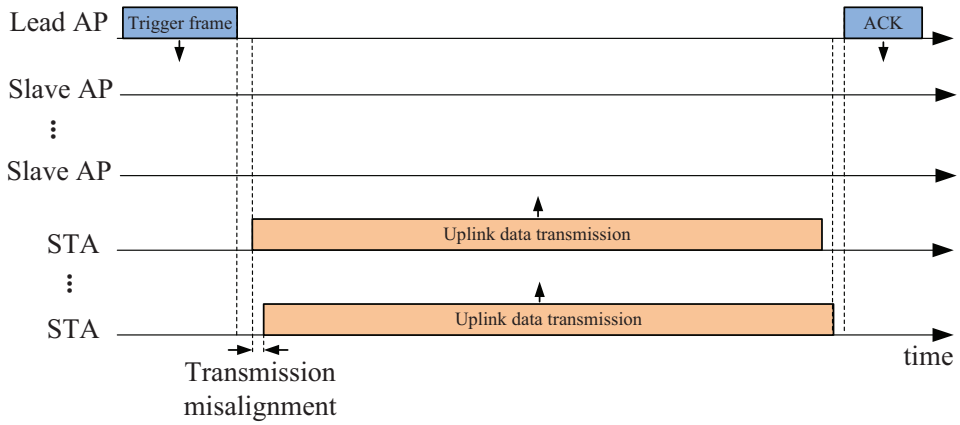


Figure 4.2: Proposed protocol for UD-MIMO.

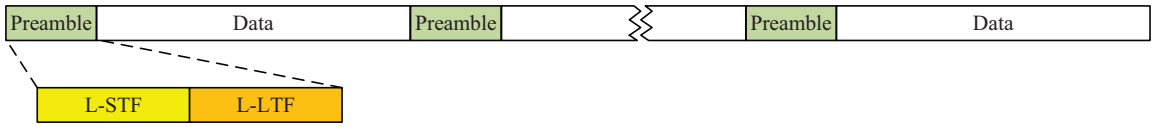
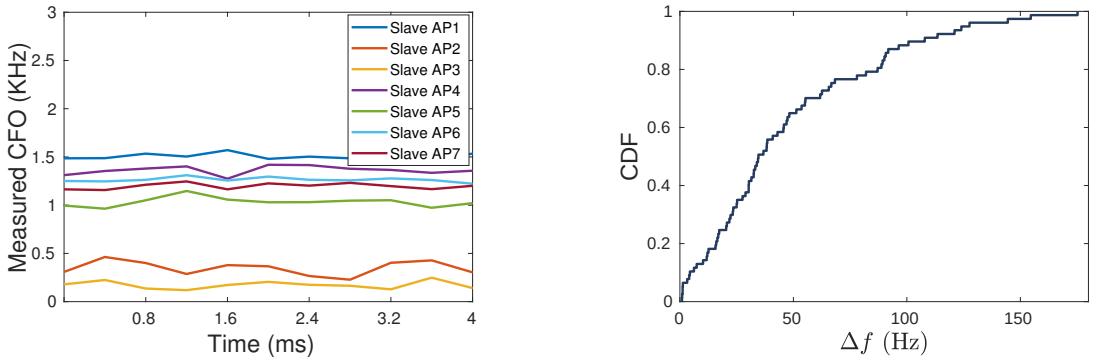


Figure 4.3: An aggregate frame for uplink transmission.

A practical consideration is whether the trigger frame from the lead AP is sufficient for the slave APs to synchronize their carrier frequency in the time period of uplink data transmission. To address this concern, we study the stability of frequency synchronization among the lead and slave APs. Fig. 4.4 shows the measured carrier frequency offsets at seven slave APs and the residual carrier frequency offsets after the compensation of frequency offsets. It is evident that the residual carrier frequency offsets are less than 180 Hz in 4 ms. This accuracy is sufficient for concurrent data transmission.

It is evident that the proposed protocol is simple and has low airtime overhead. But a big question is yet to be answered: how can the AP processor decode the data packets from the STAs? We focus on this question in the next section.



(a) (b)

Figure 4.4: Measured carrier frequency offsets at 7 slave APs with respect to a lead AP. (a) carrier frequency offsets over time; (b) distribution of residual carrier frequency offsets after frequency offset compensation.

4.4 Packet Detection

At the AP processor, decoding the data packets faces the following two challenges. First, since the STAs are driven by independent clocks, their carrier frequencies are not exactly the same. Consequently, from the APs' perspective, the received signals from different STAs have different carrier frequency offsets, which must be compensated for signal detection. However, it remains unknown how to handle such heterogeneous carrier frequency offsets at a MIMO receiver. Second, since the STAs operate independently, their uplink data packets are unlikely to be aligned in the time domain. Moreover, for 802.11 devices, the misalignment of data packets is hardly confined within the duration of OFDM's CP (800 ns). Such a time misalignment makes it hard for the AP processor to decode the data packets.

4.4.1 Overview

To address the STA-side asynchrony issue, we propose a new packet detection method for the AP processor. This new detection method lives with the STA-side asynchrony and tackles the asynchrony issue through baseband signal processing on the AP side (i.e., at the AP processor).

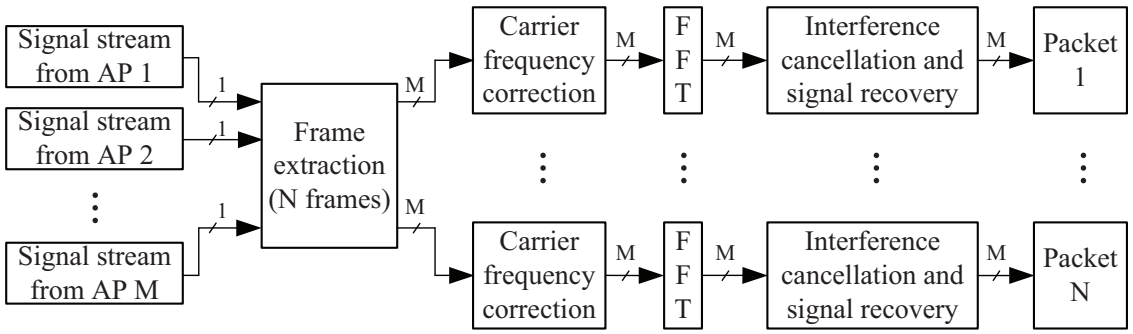


Figure 4.5: The schematic diagram of our packet detection method.

Fig. 4.5 shows the schematic diagram. The AP processor continuously receives the signal streams from the APs. From the signal streams, it extracts N signal frames, each of which corresponds to a packet from one STA. Then, it decodes each of the N signal frames separately, as illustrated in Fig. 4.5.

Consider one of the N signal frames, for example. Suppose that it corresponds to the data packet from STA i. We note that this signal frame includes not only the desired signal from STA i but also the undesired signals (interference) from other STAs. To decode this signal frame, the AP processor first performs carrier frequency correction. This module will estimate and compensate the carrier frequency offset between STA i and the APs (assuming the APs have been perfectly synchronized in Step 1 of our protocol). Then, the AP processor converts the signal to the frequency domain for signal detection. When performing signal detection, the AP processor treats the signals from other STAs (all the STAs except STA i) as unknown interference and constructs spatial filters to cancel the interference and equalize the channel distortion.

4.4.2 Frame Extraction

To extract signal frames from the signal streams, the AP processor employs cross-correlation. Specifically, the AP processor correlates each signal stream with a local copy of L-LTF. If the

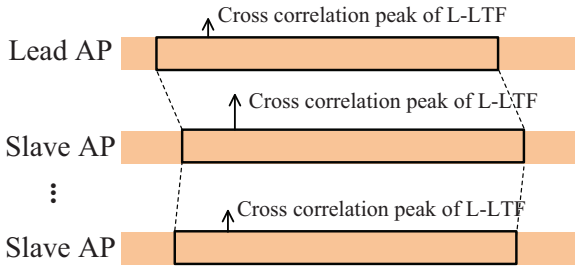


Figure 4.6: Extracting the signal frame for one STA at the AP processor via correlating signal streams with L-LTF.

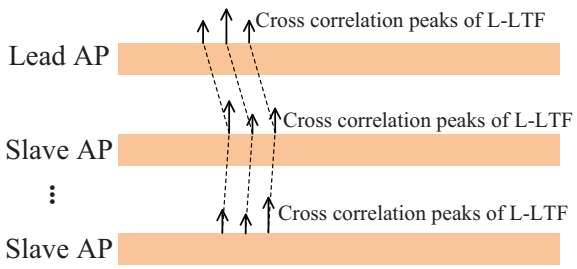


Figure 4.7: Extracting the signal frames for three STAs at the AP processor via correlating signal streams with L-LTF.

normalized correlation value is greater than a predefined correlation threshold (e.g., 0.4), then it is regarded as the start of a signal frame. Fig. 4.6 illustrates the extraction procedure when the network has one STA, and Fig. 4.7 illustrates the correlation peaks when the network has three STAs. When there are multiple correlation peaks, their corresponding signal frames are matched in order at each AP (see Fig. 4.7 as an example).

It is worth pointing out that the cross-correlation is conducted in the presence of interference and noise. The correlation value is therefore dependent on the strength of interference and the noise power. Hence, the correlation threshold should be meticulously chosen. A small threshold may lead to a false positive, and a large threshold may lead to a false negative. In our experiments, we set the threshold to 0.4 and find that it works well for the following two reasons: (i) Cross-correlation itself is resilient to interference. L-LTF has 160 samples and appears to be robust against interference. (ii) False negatives are acceptable for packet detection. A small correlation value (below the threshold) indicates that the desired signal is weak and the interference is strong. The exclusion of this stream will actually improve the performance of packet detection, provided that the spatial DoF is sufficient for packet detection.

4.4.3 Carrier Frequency Correction

Referring to Fig. 4.5, let us consider one of the N signal frames. Suppose it corresponds to the data packet from STA i. To decode this signal frame, the AP processor first performs carrier frequency correction. In the presence of inter-user (inter-STA) interference, conventional methods do not work because they are susceptible to interference. To tackle this issue, we propose a new method, which comprises two steps: (i) signal projection, and (ii) CP-based correlation. Denote $\mathbf{y}(n) \in \mathbb{C}^{M\times 1}$, $1 \leq n \leq N_s$, as the time-domain signal frame, where N_s is the number of samples in a frame. We first compute the eigenvectors as follows:

$$[\mathbf{u} \;\; \mathbf{d}] = \text{eig}\big(\sum_{n=1}^{N_s} \mathbf{y}(n)\mathbf{y}(n)^{\mathsf{H}}\big), \tag{4.1}$$

where $\text{eig}(\cdot)$ is eigendecomposition operator, $(\cdot)^{\mathsf{H}}$ is the Hermitian transpose operator, $\mathbf{u} \in \mathbb{C}^{M\times M}$ is the eigenvectors, and $\mathbf{d} \in \mathbb{C}^{M\times M}$ is diagonal matrix of eigenvalues. Then, we project the signals into the eigenvector space by letting $\tilde{\mathbf{y}}(n) = \mathbf{u}^{\mathsf{H}}\mathbf{y}(n)$, where $\tilde{\mathbf{y}}(n) \in \mathbb{C}^{M\times 1}$ and $1 \leq n \leq N_s$. Each row of $\tilde{\mathbf{y}}(n)$ can be used to estimate the carrier frequency offset, and we should pick up the best one (the one with highest signal-to-interference ratio). To do so, we perform the cross correlation again on each row of $\tilde{\mathbf{y}}(n)$ and choose the one with the maximum cross correlation value for carrier frequency offset estimation. Denote $\tilde{y}_m(n)$ as the row of $\tilde{\mathbf{y}}(n)$ that has the maximum cross correlation value. Denote $\hat{\theta}_i$ as the estimated phase offset per sample between the APs and STA i. Then we have $\hat{\theta}_i = (1/64)\arg\big(\sum_{n\in\mathcal{C}} \tilde{y}_m(n)\tilde{y}_m(n+64)^*\big)$, where $\arg(\cdot)$ is the angle of a complex number, $(\cdot)^*$ is complex conjugate operator, $\mathcal{C}$ is the set of samples in the CP of all OFDM symbols, and 64 is the distance between CP and its original copy in OFDM symbol.

After obtaining $\hat{\theta}_i$, we then compensate the carrier frequency offset by letting $\bar{\mathbf{y}}(n) = \mathbf{y}(n) \cdot e^{jn\hat{\theta}_i}$, $1 \leq n \leq N_s$. The resultant signal frame $\bar{\mathbf{y}}(n)$ is then sent to the FFT module, as shown in

Fig. 4.5.

4.4.4 Interference Cancellation and Signal Recovery

Problem Formulation. After correcting the carrier frequency offset, the FFT module in Fig. 4.5 converts the signal frame from the time domain to the frequency domain. We let $Y_j(l,k)$ denote the output signals from the FFT module, where $j \in \{1,2,\cdots,M\}$ is the index of received signal streams (APs), $l \in \{1,2,\cdots,L\}$ is the index of OFDM symbols, and $k \in \{1,2,\cdots,K\}$ is the index of OFDM subcarriers. Assume that the signals in a frame experience block channel fading. Then, the signal transfer function can be written as:

$$Y_j(l,k) = \underbrace{H_{ji}(k)X_i(l,k)}_{\text{desired signal}} + \underbrace{\sum_{i' \in \mathcal{N}, i' \neq i} H_{ji'}(k)\bar{X}_{i'}(l,k)}_{\text{unknown interference}} + \underbrace{W_j(l,k)}_{\text{noise}}, \tag{4.2}$$

where $H_{ji}(k)$ is the channel between AP j and STA i, $X_i(l,k)$ is the original signal from STA i, $\bar{X}_{i'}(l,k)$ is the interfering signal from STA i', and $\mathcal{N}$ is the set of STAs.

For the transfer function in (4.2), we have the following three remarks. First, this transfer function requires carrier frequency synchronization between STA i and the APs. But it does not require phase synchronization between STAs and APs. Actually, the phase offset between STA i and AP j is considered as a part of $H_{ji}(k)$. Second, $X_i(l,k)$ in (4.2) is the original signal transmitted by STA i. But $\bar{X}_{i'}(l,k)$ is not the original signal transmitted by STA $i' \in \mathcal{N}/\{i\}$. This is because $\bar{X}_{i'}(l,k)$ is completely distorted by the carrier frequency offset and time misalignment between STA i' and the APs. It is considered unknown interference in this transfer function. Third, since $\bar{X}_{i'}(l,k)$ is an unknown interfering signal, it is hard to estimate the channel $H_{ji'}(k)$. This is the core challenge in signal recovery.

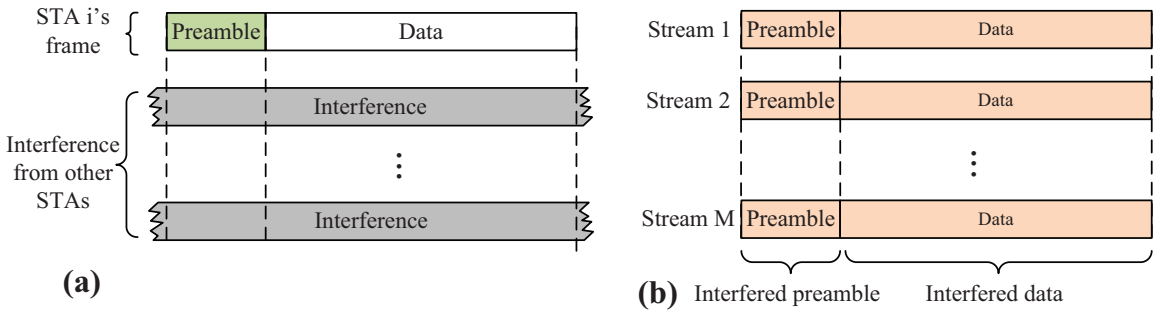


Figure 4.8: (a) Transmitted signal and interference at STAs. (b) Received signal and interference at the APs.

Our Detection Method. Based on (4.2), if we know all the channels, then we can construct a spatial filter $\mathbf{G}(k) = [G_1(k), G_2(k), \cdots, G_M(k)]$ so that $\sum_{j=1}^{M} G_j(k)H_{ji}(k) = 1$ and $\sum_{j=1}^{M} G_j(k)H_{ji'}(k) = 0$, $i' \in \mathcal{N}/\{i\}$. Such a spatial filter can cancel the interference and recover the desired signal. This method is actually the well-known zero-forcing MIMO detector. By taking into account the effect of noise, the zero-forcing detector can be elevated to an MMSE detector.

Now the question is how to construct the spatial filter $\mathbf{G}(k)$ in the absence of channel knowledge. To address this question, we propose a training-based method. Consider the signal frame transmission from STA i to the APs. As illustrated in Fig. 4.8(a), we assume that (i) STA i's preamble is interfered by unknown signals from other STAs; and (ii) STA i's preamble is independent of its interference. Then, we focus on the received signal frame at the AP processor, which is illustrated in Fig. 4.8(b). The signal frame is composed of two parts: *interfered preamble* and *interfered data*. Since the preamble is known at the AP processor *a priori*, we use the interfered preamble in Fig. 4.8(b) as the training sequence to construct the spatial filter for data detection. Specifically, we construct the spatial filter as follows:

$$\mathbf{G}(k) = \Big[\sum_{(l,k') \in \mathcal{Q}(k)} \mathbf{Y}(l,k')\mathbf{Y}(l,k')^{\mathsf{H}} \Big]^{+} \Big[\sum_{(l,k') \in \mathcal{Q}(k)} \mathbf{Y}(l,k')X_i(l,k')^{\mathsf{H}} \Big], \tag{4.3}$$

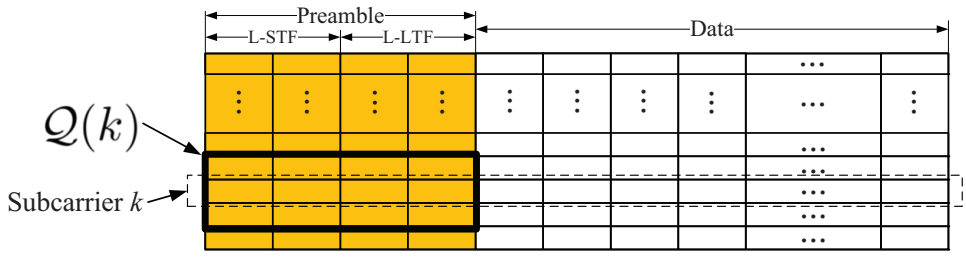


Figure 4.9: Illustrating set $\mathcal{Q}(k)$ for filter construction.

where $[\cdot]^+$ is Moore–Penrose inverse (pseudo-inverse). $\mathbf{Y}(l,k) \in \mathbb{C}^{M\times 1}$ is the frequency-domain signal vector, i.e., $\mathbf{Y}(l,k) = [Y_1(l,k), Y_2(l,k), \cdots, Y_M(l,k)]^{\mathsf{T}}$. $\mathcal{Q}(k)$ is a set of reference symbols in the preamble. $\mathcal{Q}(k)$ can be empirically set. In our experiments, we let $\mathcal{Q}(k) = \{1 \leq l \leq 4, k-1 \leq k' \leq k+1\}$, as shown in Fig. 4.9.

After constructing the spatial filter, we then use it to estimate the original signal by:

$$\hat{X}_i(l,k) = \sum_{j=1}^{M} G_j(k)^* Y_j(l,k), \tag{4.4}$$

where $\hat{X}_i(l,k)$ is the estimated signal from STA i and $G_j(k)$ is the jth entry in vector (filter) $\mathbf{G}(k)$.

Discussions. We have the following two remarks on the proposed packet detection method. First, the proposed method does not require channel knowledge to decode the packet, as evidenced by (4.3) and (4.4). Instead, it uses the interfered preamble as the training sequence to construct a filter, which is then used to cancel the interference and equalize the channel distortion for signal recovery. Second, the proposed detection method is a heuristic. We will resort to experiments to evaluate its performance. As we will see in Section 4.6.1, this detection method yields surprisingly superior performance. With this detection method, the performance of UD-MIMO is close to that of MU-MIMO in all tested scenarios.

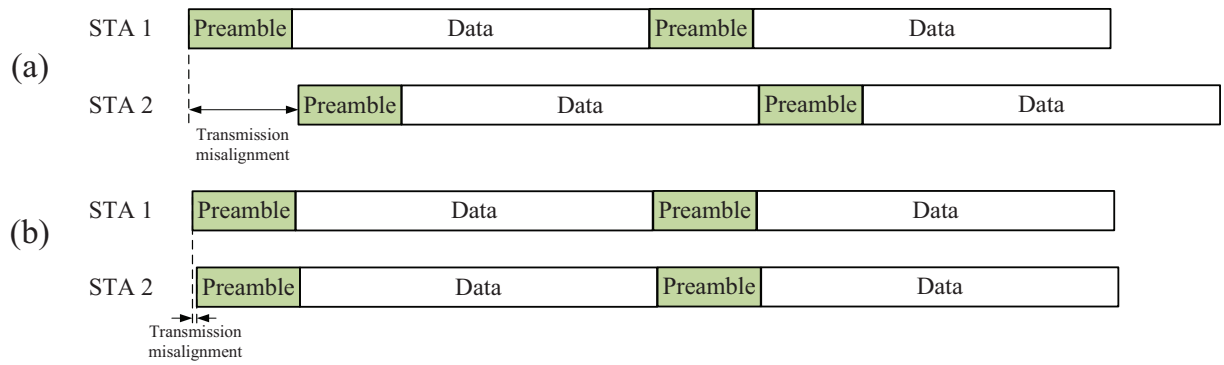


Figure 4.10: (a) Transmission misalignment is greater than the time duration of preamble. (b) Transmission misalignment is less than the time duration of CP (guard interval).

4.5 Compatibility with 802.11 Client Devices

In this section, we first point out the practical issues when UD-MIMO works with incumbent off-the-shelf Wi-Fi client devices and then propose a solution to these issues.

Practical Issues. UD-MIMO heavily relies on the new packet detection method to tame the asynchrony among the STAs. However, the packet detection method was proposed under the following two assumptions: (i) Referring to Fig. 4.8(a), STA i's preamble is interfered by the signals from other STAs. (ii) Referring to Fig. 4.8(a) again, STA i's preamble is linearly independent of the interfering signals from other STAs. To see why these two assumptions are mandatory, let us consider the examples in Fig. 4.10.

In Fig. 4.10(a), the transmission misalignment of the two STAs is greater than the time duration of the preamble. In this case, STA 1's first frame cannot be decoded at the AP processor. This is because its preamble is not interfered by the signal from STA 2. As a result, the spatial filter constructed based on this preamble cannot cancel the interference from STA 2. For STA 1's second frame, it can be decoded at the AP processor because its preamble is interfered by the signal from STA 2. In contrast, for STA 2's two frames, both of them can be decoded at the AP processor because their preambles are interfered by the signal from STA 1.

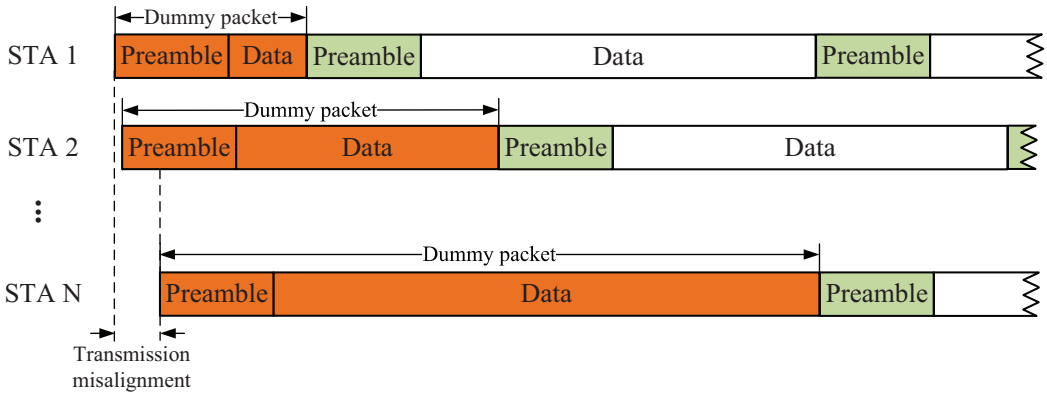


Figure 4.11: Insert a different-length dummy packet at each STA to enable UD-MIMO transmission.

In Fig. 4.10(b), the transmission misalignment of the two STAs is less than the time duration of CP (guard interval). In this case, all of the four frames (two for STA 1 and two for STA 2) cannot be decoded at the AP processor. This is because their preambles are interfered by the same interference. From the AP processor's perspective, it has no way to differentiate the signal from STA 1 and that from STA 2. The spatial filter constructed based on the interfered preamble can neither cancel interference nor equalize the channel distortion. Therefore, all four frames cannot be decoded.

Such cases are likely to occur in practice. In Wi-Fi networks, the preamble is 16 microseconds, and the CP is 0.8 microseconds for long guard interval option and 0.4 microseconds for short guard interval option. Suppose that the synchronization error achieved by the timing synchronization function (TSF) specified in IEEE 802.11 is uniformly distributed in [0, 20] microseconds [57]. Then, the probability of case 1 is about 20%, and the probability of case 2 is 4% (or 2% for short guard interval option). Collectively, the probability of these two cases is 24%. Therefore, properly handling these two cases is imperative towards the real application of UD-MIMO.

Our Solution. To fulfill those two assumptions in the presence of STAs' transmission misalignment, our solution is simple. We insert a dummy packet at the beginning of uplink transmission at each STA, as illustrated in Fig. 4.11. By properly setting the length of the dummy packet, the

preamble of each data packet will meet those two requirements. To determine the length of the dummy packet at each STA, let us again assume that the transmission misalignment is within 20 microseconds [57]. Then, we set the length of the dummy packet at STA i to $7i$ OFDM symbols ($1 \leq i \leq N$), including both preamble and data.

For the proposed solution, three remarks are in order.

Remark 1. To fulfill those two assumptions, the preambles from different STAs can partially overlap with each other. For example, the L-STF from one STA can overlap with the L-STF from another STA. In such a case, the AP processor can still decode the packets. Taking this fact into consideration may help reduce the length of dummy packets at the STAs.

Remark 2. Apparently, the proposed solution entails additional airtime overhead to enable UD-MIMO transmission. Further, the overhead slightly increases with the number of STAs. This issue can be alleviated by aggregating more packets (signal frames) in the uplink transmission. Since the channel coherence time is long enough in WLANs, an aggregate frame can accommodate hundreds of OFDM symbols. Then, the amortized overhead is acceptable in practice.

Remark 3. Since the dummy packet is a normal packet, no hardware modification is needed to insert the dummy packet for a commercial Wi-Fi client device. Rather, it can be implemented through modifying a Wi-Fi device's driver. The length of the dummy packet can be specified in the trigger frame by the lead AP. On the AP side, the AP processor will automatically drop the dummy packet, either because it cannot be decoded or it does not have necessary MAC information. In either case, the dummy packet will not affect the upper-layer applications.

4.6 Experimental Evaluation

In this section, we conduct experiments to evaluate the performance of UD-MIMO on the two wireless testbeds.

4.6.1 Implementation and Experimental Setup

We have built two testbeds to evaluate the performance of UD-MIMO in real wireless environments.

802.11 Wi-Fi Dongle Testbed. The purpose of this testbed is to validate the practicality of UD-MIMO as well as its compatibility with commercial off-the-shelf Wi-Fi devices. For the STAs, we use Wi-Fi dongles (Alfa AWUS036NHA Wireless USB Adapters), which are built on Qualcomm Atheros AR9271 chipset [58] and support IEEE 802.11b/g/n. We modify its firmware (`modwifi-ath9k-htc` in [59]) to disable carrier sense, RTS/CTS, ACK, set SIFS/AIFS to zero, and insert a dummy packet for UD-MIMO. For simplicity, we fix the MCS index to 2, which corresponds to QPSK modulation, 3/4 coding rate, and 18 Mbps data rate. While we use this specific MCS, UD-MIMO works with other MCS as well. We set channel bandwidth to 20 MHz and guard interval (OFDM CP) to 800 ns. The transmit power is fixed to 17 dBm, and the carrier frequency is set to 2.427 GHz (channel 4).

For the APs, we implement them using a set of USRP N210 devices [39]. Each USRP N210 device is connected to a D-Link SmartPro Switch via 1Gbps Ethernet RJ45 Cord, and the switch is connected to a computer via 10Gbps SFP+ DAC Cable. A software suite is developed using C++ and deployed at the computer to implement the protocol (see Fig. 4.2) and process the baseband signals. The output of our software suite is estimated signals from the STAs. Post-processing modules (e.g., deinterleaving, channel decoding, descrambling, and decryption) are not implemented.

USRP Testbed. The purpose of this testbed is to quantify the performance gap between UD-

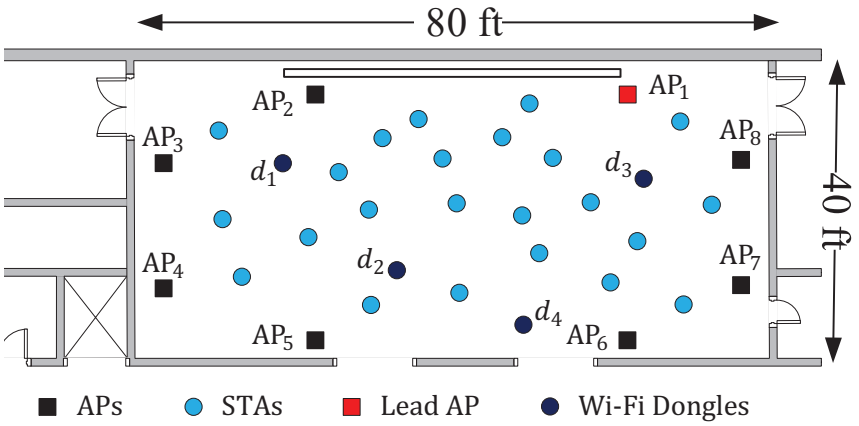


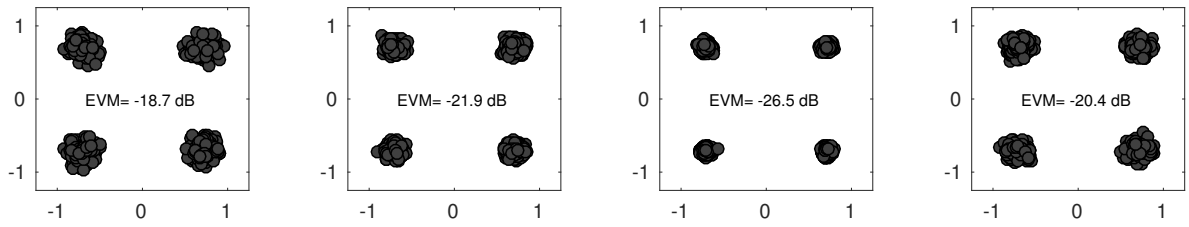
Figure 4.12: A conference room for UD-MIMO evaluation.

MIMO and MU-MIMO in the same scenarios. In this testbed, both APs and STAs are custom-built using USRP N210 devices. As such, we have full control for both APs and STAs. To measure the performance of MU-MIMO, we synchronize both APs and STAs using external clocks (Ettus' Octoclock CDA-2990G [60]). For ease of experimentation, we set the sampling rate to 5 Msps. Other parameters are the same as the 802.11 testbed.

Experimental Setup. Fig. 4.12 shows our experimental setup in a large conference room, where 8 APs and N STAs are deployed. The 8 APs are placed at the locations marked by solid boxes, and one of them is selected as lead AP. The 8 APs are connected to a computer via a 1/10 Gbps Ethernet switch. A set of spots (small circles in the figure) are marked out for the possible locations for the N STAs.

4.6.2 Comparison Baseline and Performance Metrics

Comparison Baseline. We use MU-MIMO as the comparison baseline to evaluate the performance of UD-MIMO. MU-MIMO serves as an upper bound for UD-MIMO. We quantify the performance gap between UD-MIMO and MU-MIMO. In MU-MIMO, all APs are synchronized via external clocks, and all STAs are synchronized via external clocks.



(a) Dongle 1 (b) Dongle 2 (c) Dongle 3 (d) Dongle 4

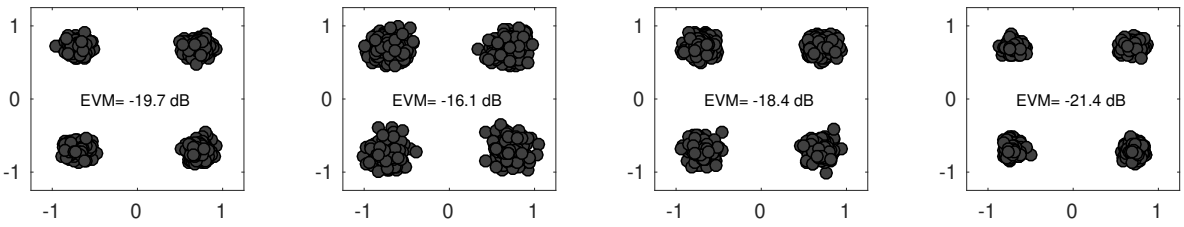
Figure 4.13: The demodulated signals from the four dongles when CSMA is used (four dongles in four different time slots).

Performance Metrics. We use EVM and data rate, as defined in Section 2.9.1 (equations (2.18) and (2.19)), to evaluate the performance of UD-MIMO in practice. Specifically, the data rate is calculated by: $r = \frac{48}{80} \times \mathrm{BW} \times \gamma(\mathrm{EVM})$ Mbps, where 48 is the number of payload subcarriers, 80 is the points of one OFDM symbol (including CP), 20 is the signal sampling rate (in Msps), and $\gamma(\mathrm{EVM})$ is the average number of bits carried by one symbol and its possible values are given in Table 2.1. In our experiments, we use $\mathrm{BW} = 20$ for the 802.11 testbed and $\mathrm{BW} = 5$ for the USRP testbed.

4.6.3 802.11 Wi-Fi Dongle Testbed

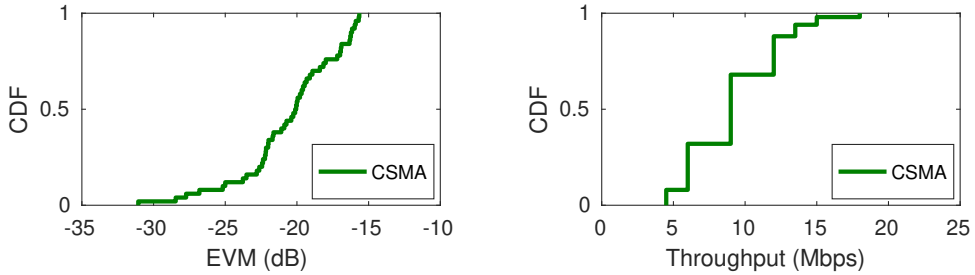
On this testbed, we measure the uplink data rate per STA in two schemes: CSMA (interference avoidance) and UD-MIMO.

CSMA. In conventional Wi-Fi networks, only one Wi-Fi dongle can be allowed to communicate with one AP in a time slot. As such, we consider four dongles in four different time slots. In the ith time slot, dongle i placed at location d_i sends data packet to the lead AP, $1 \leq i \leq 4$. Since each time slot has only one active dongle, there is no interference in this case. Fig. 4.13 plots the demodulated signal at the lead AP in the four time slots. Specifically, the measured EVMs for the four dongles are -18.7 dB, -21.9 dB, -26.5 dB, -20.4 dB, respectively. We then extrapolate the



(a) Dongle 1 (b) Dongle 2 (c) Dongle 3 (d) Dongle 4

Figure 4.14: The demodulated signals from the four dongles when UD-MIMO is used (four dongles in the same time slot).



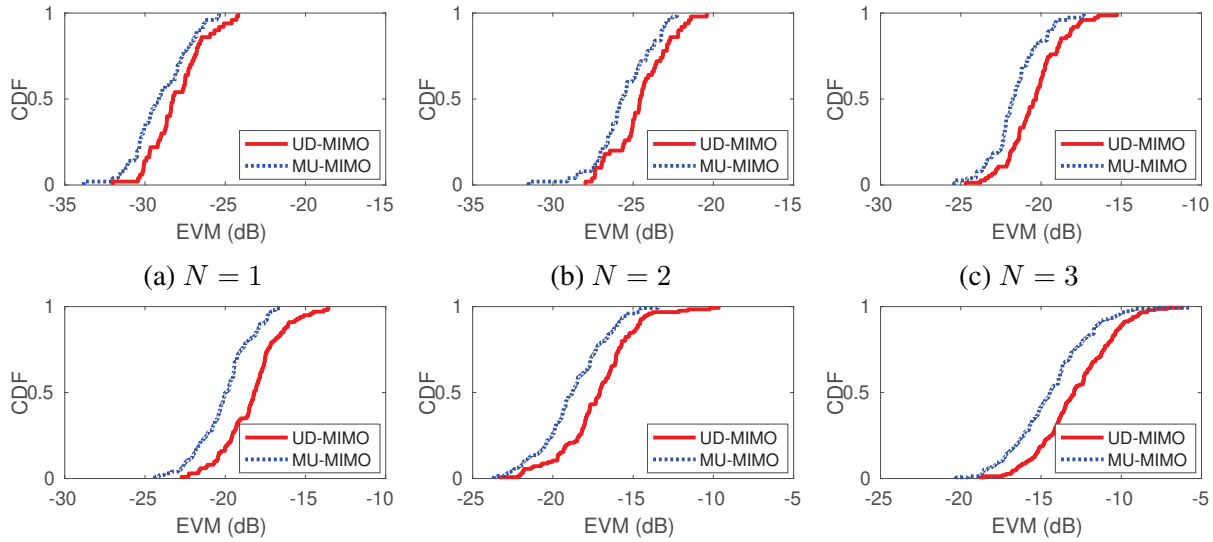
(a) Measured EVM. (b) Throughput.

Figure 4.15: Performance of the CSMA (interference-free) scheme.

data rate based on the measured EVM values. Since each dongle uses one-fourth of time resources for data transmission, the data rate should be divided by four in the calculation. Therefore, the calculated data rate is 6.0 Mbps for dongle 1, 9.0 Mbps for dongle 2, 13.5 Mbps for dongle 3, 9.0 Mbps for dongle 4. Collectively, the total throughput achieved by CSMA is 37.5 Mbps.

UD-MIMO. Using UD-MIMO, we let the 8 APs serve 4 Wi-Fi dongles simultaneously in the uplink. The 4 dongles are placed at d_1, d_2, d_3, and d_4 in Fig. 4.12. The received signals at the 8 APs are jointly decoded at the computer. Fig. 4.14 shows the constellation of the decoded signals. As shown in the figure, the measured EVMs are -19.7 dB, -16.1 dB, -18.4 dB, and-21.4 dB, respectively. The calculated data rates are 36 Mbps, 24 Mbps, 24 Mbps, and 36 Mbps, respectively. Collectively, the total uplink throughput achieved by UD-MIMO is 120 Mbps.

Observations. Based on the above experimental results, we have the following observations. First, UD-MIMO can serve multiple commercial off-the-shelf Wi-Fi client devices simultaneously



(a) $N = 1$ (b) $N = 2$ (c) $N = 3$

(d) $N = 4$ (e) $N = 5$ (f) $N = 6$

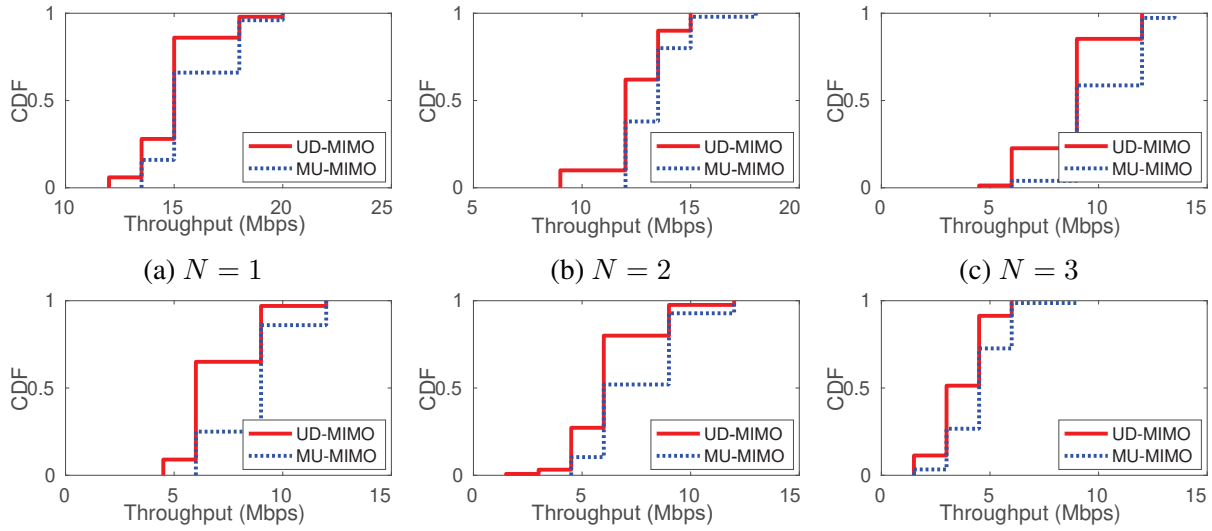
Figure 4.16: Measured EVM of the demodulated uplink signal from each of the N STAs.

in real wireless environments. This indicates that UD-MIMO is compatible with incumbent Wi-Fi client devices. Second, compared to conventional CSMA-based Wi-Fi networks, UD-MIMO can improve uplink throughput significantly. For the above case (8 APs and 4 dongles), UD-MIMO offers $3.2\times$ throughput gain compared to CSMA.

4.6.4 USRP Testbed

On the USRP testbed, we measure the performance of UD-MIMO and MU-MIMO in the same scenarios and quantify their performance gap.

CSMA. This is an interference-free case. The interference is avoided in the time domain by assigning different STAs into different time slots. The STAs send their packets to the lead AP using a round-robin scheduler. We measure the EVM of the decoded signal for each STA at the lead AP. Fig. 4.15(a) plots the distribution of our measured EVM when the STA is placed throughout all the locations (small circles) in Fig. 4.12. Fig. 4.15(b) plots the distribution of our calculated data rate. Since the AP serves a single STA in each time slot, the average uplink throughput achieved



(a) $N = 1$ (b) $N = 2$ (c) $N = 3$

(d) $N = 4$ (e) $N = 5$ (f) $N = 6$

Figure 4.17: The data rate achieved by each of the N STAs.

by CSMA is 9.3 Mbps.

UD-MIMO versus MU-MIMO. In UD-MIMO, we let the 8 APs serve N STAs simultaneously in the uplink, where $1 \leq N \leq 6$. In each instance, we place the N STAs at N different locations (the small circles) in Fig. 4.12. At the computer, we measure the EVM of demodulated uplink signal from each of the N STAs. We then repeat the same measurements for MU-MIMO, for which we synchronize the 8 APs using an external clock (10 MHz reference signal and 1 PPS) and synchronize the N STAs using another external clock.

Fig. 4.16 plots the distribution of our measured EVM when UD-MIMO and MU-MIMO are used. In UD-MIMO, the average EVM of the demodulated uplink signals is -28.1 dB when the APs serve one STA, -24.5 dB when the APs serve two STAs, -20.4 dB when the APs serve three STAs, -18.3 dB when the APs serve four STAs, -17.2 dB when the APs serve five STAs, and -14.3 dB when the APs serve six STAs. Moreover, as shown in the figure, the EVM gap of UD-MIMO and MU-MIMO is only about 2.0 dB. This means that UD-MIMO successfully resolves the synchronization issues in distributed WLANs.

We extrapolate the measured EVM to each STA's data rate (with $b = 5$ Msps). Fig. 4.17 presents the calculated data rate. The staircase shape of the curves is caused by the MCS, which yields a discrete data rate in nature. When UD-MIMO is used, the average data rate per STA is 15.0 Mbps when the APs serve one STA, 12.4 Mbps when the APs serve two STAs, 8.7 Mbps when the APs serve three STAs, 7.0 Mbps when the APs serve four STAs, 6.2 Mbps when the APs serve five STAs, and 3.7 Mbps when the APs serve six STAs. Accordingly, the total uplink throughput achieved by UD-MIMO is 15.0 Mbps when $N = 1$, 24.8 Mbps when $N = 2$, 26.2 Mbps when $N = 3$, 28.0 Mbps when $N = 4$, 31.0 Mbps when $N = 5$, and 22.1 Mbps when $N = 6$.

Throughput Comparison. Finally, we compare the total uplink throughput achieved by CSMA, UD-MIMO, and MU-MIMO. For CSMA, since it serves one STA at a time, the total uplink throughput is the average of all STAs' data rates. For UD-MIMO and MU-MIMO, since it serves N STAs simultaneously, the total uplink throughput is the multiplication of N and the average of per-STA data rate. Fig. 4.18 presents the comparison of the total uplink throughput achieved by the three techniques. It is evident that, in each case, the throughput of UD-MIMO is much higher than that of CSMA and close to that of MU-MIMO. On average of the six cases, UD-MIMO achieves $3.4\times$ throughput compared to CSMA and achieves 82% throughput of MU-MIMO. One may notice that the throughput of all three techniques decreases when the number of STAs increases from 5 to 6. This is because the sixth STA brings significant co-channel interference to the existing 5 STAs. The significant increase of co-channel interference can be attributed to the ill-conditioned MIMO channel between the 6 STAs and the 8 APs.

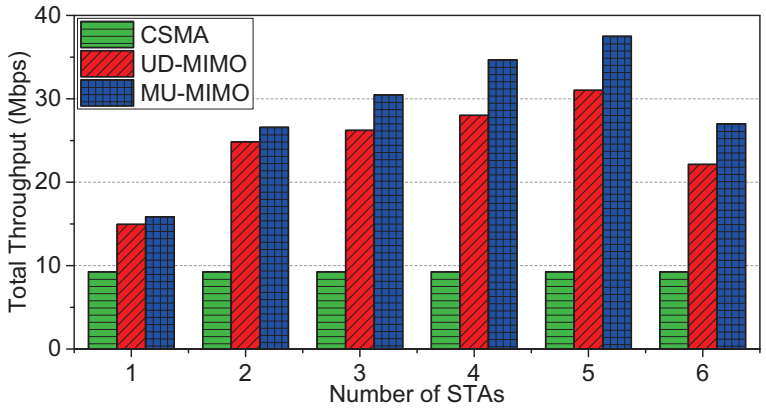


Figure 4.18: Throughput comparison of CSMA, UD-MIMO, and MU-MIMO.

4.7 Chapter Summary

In this chapter, we presented UD-MIMO, a practical uplink distributed MIMO scheme for WLANs. UD-MIMO enables concurrent data transmissions from multiple STAs to multiple APs. UD-MIMO is compatible with commercial off-the-shelf 802.11 devices (with modified driver). The enabler behind UD-MIMO is a new signal detection method, which can decode concurrent data packets from asynchronous STAs. We have built a prototype of UD-MIMO on two wireless testbeds and demonstrated its compatibility with Qualcomm Atheros 802.11 devices. Our experimental results show that UD-MIMO offers $3.4\times$ throughput compared to the CSMA-based interference-avoidance approach. Our experimental results also show that UD-MIMO achieves 82% throughput of MU-MIMO.

Chapter 5

MU-MIMO in LoRaWANs

LoRa has emerged as a key wireless communication technology for a gateway to provide geographically-distributed IoT devices with low-rate, long-range connections. In this chapter, we present MaLoRaGW, the first-of-its-kind Multi-antenna LoRa GateWay that enables MU-MIMO LoRa communications in both uplink and downlink. MaLoRaGW was inspired by the success of MU-MIMO in cellular and Wi-Fi networks. The key component of MaLoRaGW is a joint baseband PHY design for uplink packet detection and downlink beamforming. Its innovation lies in three modules: spatial signal projection, accurate channel estimation, and implicit beamforming, all of which reside only in a LoRa gateway and require no modification on LoRa client devices. We have built a prototype of two-antenna MaLoRaGW on a USRP device and extensively evaluated its performance with commercial LoRa dongles in three scenarios: lab, office building, and university campus. Our experimental results show that, compared to the state-of-the-art, the two-antenna MaLoRaGW increases the throughput by 10% and reduces the PER by 40% in uplink. In downlink, it improves the throughput by 95% while maintaining a similar PER.

5.1 Introduction

Recent years have witnessed a continuous and rapid increase in the number of low-cost IoT devices, most of which have a low data rate requirement but desire to prolong their battery lifetime [61–64]. LoRaWAN has emerged as a key wireless communication technology to connect a large number of

geographically-distributed IoT devices with a single gateway [65–67]. LoRa devices employ chirp spread spectrum (CSS) modulation at the physical layer, which allows a LoRa receiver to decode packets at very low signal-to-noise (SNR) scenarios (e.g., -20dB [68]) and therefore permits a long communication range (e.g., 8km), as indicated by its name.

A central problem with LoRaWAN is packet collision, which fundamentally limits its packet delivery rate and network throughput in user-dense scenarios [61, 69–71]. Different collision recovery approaches have been proposed to address this problem [72–80]. Most of these approaches aim to decode collided LoRa packets by exploiting the unique signal features in the frequency (see, e.g., [75, 81]) and time/power (see, e.g., [76, 77, 82]) domains. While these approaches have demonstrated a significant enhancement for LoRa collision resilience, the state-of-the-art is far from satisfaction in practice. For example, mLoRa [83] and FTrack [75] can rarely decode three collided LoRa packets while maintaining its symbol error rate less than 10%. Since the potential in time and frequency domains has been well exploited, it is natural to explore the spatial domain of a LoRa gateway so that its collision resilience can be further improved by leveraging multi-antenna techniques.

In addition, most existing LoRa work focuses on uplink packet detection. The downlink transmission efficiency of LoRaWAN has not been well explored. One may think that, in LoRaWAN, uplink is important while downlink is not. This was true. However, with the continuous expansion of LoRa application landscape, downlink communications have become increasingly important for some LoRa applications such as remote sensor control, massive machine operation, and smart city/building management. When LoRa devices require ACK to confirm the successful delivery of their packets (e.g., TCP connection), the downlink transmission becomes particularly useful. Multiple antennas on a LoRa gateway will make it possible to enable concurrent downlink transmission, which will significantly improve the downlink throughput and find many applications in

emerging LoRaWANs.

In this chapter, we present MaLoRaGW, the first-of-its-kind multi-antenna LoRa gateway that enables MU-MIMO LoRa communications in both uplink and downlink. MaLoRaGW was inspired by the success of MU-MIMO in cellular and Wi-Fi networks [38, 55, 84–86]. It exploits the SDoF provided by its multiple antennas for two purposes: i) enhance packet detection in uplink and ii) enable concurrent packet transmission in downlink. In uplink, MaLoRaGW projects the received multiple signal streams into different spatial subspaces, making it possible to decode a weak collided LoRa packet that cannot be decoded by a LoRa receiver with a single antenna. In downlink, MaLoRaGW performs beamforming to send multiple independent packets to different users, thereby improving downlink throughput and reducing the packet round-trip latency.

Realizing MaLoRaGW in practice is a nontrivial task. In uplink, reaping the multiplexing gain requires the MU-MIMO channel knowledge to mitigate strong inter-user interference so as to decode weak collided packet [43]. However, estimating channel knowledge requires fine-grained frequency/timing synchronization, which in turn requires channel knowledge for signal projection [45, 87]. The synchronization and channel estimation form a death loop, which must be broken in order to decode a weak collided packet. To address this issue, we perform principal component analysis (PCA) on the received signal streams and find that the principal components (i.e., eigenvectors of the time-domain signals' covariance matrix) perform very well in the separation of weak and strong packets. We thus use the principal components of received signal streams to project them into their subspace. Based on the projected signal streams, channel coefficients are estimated for each user and the corresponding packets are decoded.

In downlink, concurrent transmission relies on beamforming to pre-mitigate inter-user interference so that each LoRa device is capable of decoding its desired packets; and the construction of beamforming filters requires downlink channels between a gateway and its users. The key chal-

lenge here is channel acquisition. An approach that is widely used in cellular and Wi-Fi networks is explicit channel feedback, i.e., each user estimates downlink channel and reports it back to base station (or access point). However, LoRa devices do not have the luxury to perform explicit channel feedback. To address this issue, MaLoRaGW adopts implicit channel estimation. That is, MaLoRaGW estimates uplink channels based on its received (collided or uncollided) packets and performs channel calibration to infer downlink channels based on the estimated uplink channels. This approach is transparent to LoRa users, making MaLoRaGW backward compatible with off-the-shelf LoRa devices.

We have built a prototype of MaLoRaGW and evaluated its performance in realistic scenarios of three different scales: lab, office building, and university campus. It has been validated that MaLoRaGW is backward compatible with COTS LoRa devices. Extensive experiments have been conducted to evaluate the PER and throughput performance of MaLoRaGW against the state-of-the-art LoRa gateways. Our experimental results show that, in uplink, the two-antenna MaLoRaGW increases the throughput by 10% and reduces the PER by 40%. In downlink, it improves the throughput by 95% while maintaining a similar PER. One may wonder why uplink throughput gain is small. This can be partially attributed to the fact that a single-antenna LoRa device is already capable of decoding collided packets thanks to its CSS modulation. Actually, in uplink, the benefit of MaLoRaGW manifests in the PER reduction (by 40%), which leads to a more reliable LoRa communication.

The contributions of this chapter can be summarized as follows.

- MaLoRaGW, to the best of our knowledge, is the first one of studying MU-MIMO for LoRaWANs. It presents a novel LoRa PHY design that enhances concurrent packet detection in uplink and enables concurrent packet transmission in downlink.

- MaLoRaGW introduces new signal processing techniques to enable MU-MIMO for LoRaWAN, including PCA-based synchronization, robust channel estimation, and implicit beamforming. Through a joint uplink and downlink design, MaLoRaGW is backward compatible with COTS LoRa devices.

- MaLoRaGW has been evaluated in three different-scale scenarios with off-the-shelf LoRa devices. Experimental results show that, compared to existing LoRa gateways, a two-antenna MaLoRaGW can slightly improve the throughput in uplink but nearly double the throughput in downlink.

5.2 Related Work

Recently, many schemes have been proposed to address the packet collision problem in LoRaWANs. Most existing schemes leverage the CSS modulation features in the frequency (e.g., [75, 81]) and time/power (e.g., [76, 77, 79, 80, 82, 83]) domains to decode collided packets. MaLoRaGW is designed based on existing work (e.g., Choir [81], FTrack [75] and PCube [88]) for its concurrent uplink transmission, and it complements existing work by enabling concurrent downlink transmission for the first time.

Frequency Feature Exploitation. In [81], Eletreby et al. proposed Choir to differentiate the collided packets from different users based on their unique frequency offsets. Choir was designed based on the observation that many hardware imperfections (e.g., time, frequency, or phase offsets) contribute to the frequency shifts of CSS chirps. In [75], Xia et al. proposed FTrack to decode collided chirps for LoRa. FTrack applies a sliding demodulation window to the received signals and traces the variations of the detected frequencies. By leveraging the timing misalignment of collided chirps, FTrack filters out the undesired chirps based on their discontinued frequencies.

MaLoRaGW borrows the ideas from these works but extends them to the spatial domain. Moreover, MaLoRaGW extends concurrent transmission from uplink to downlink.

Power Feature Exploitation. In [76], Tong et al. proposed a scheme called CoLoRa to enable multi-LoRa packet reception. CoLoRa uses the time offset between collided packets as well as the peak power ratio of the demodulated chirps to differentiate collided chirps. Similarly, NScale in [82] successfully demodulates collided chirps using both normal down-chirp and non-stationary scaled down-chirp symbols. It first pairs the resultant peaks and then calculates their peak scaling factors to differentiate collided packets from different users. In [77], Wang et al. introduced a collision recovery scheme (called OCT) by leveraging the time and power offsets of collided packets. OCT first ranks the power of the detected peaks and then classifies the collided packets based on their peak power. Similar ideas have also been studied in [79, 80].

MIMO Diversity for LoRa Uplink. Thus far, spatial domain has rarely been exploited for LoRa. Pioneering work [88] presents a MIMO-based gateway design (PCube) to decode the collided packets. PCube combines LoRa signal features in time and frequency domains with the measured phase difference over different antennas to decode the collided packets. MaLoRaGW differs PCube in two aspects: i) PCube considers uplink only while MaLoRaGW mainly focuses on downlink; ii) PCube exploits spatial diversity while MaLoRaGW exploits spatial multiplexing.

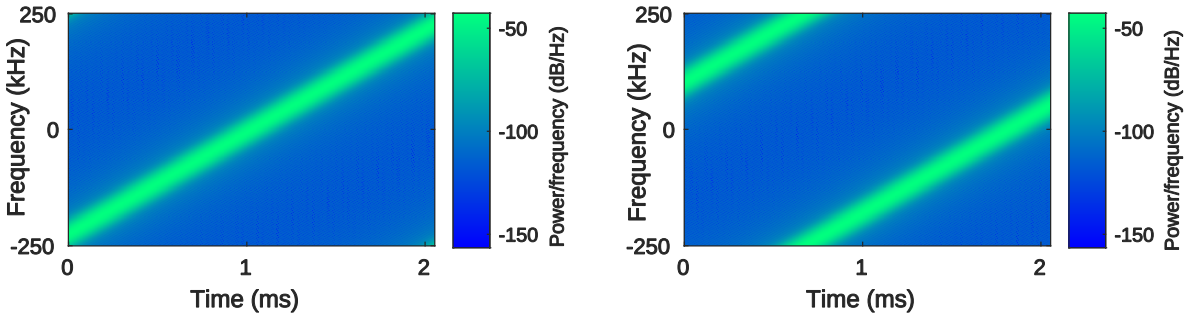
New Waveform and Detector for LoRa. In [89], Li et al. introduced CurvingLoRa to improve the capacity of LoRaWANs. The key idea is to use a non-linear base chirp for the modulation of LoRa signal. While it can significantly increase the capacity of a LoRa gateway, it is not backward compatible with incumbent LoRa devices and may become more sensitive to timing offset. In [90], Li et al. proposed a demodulation scheme (called NELoRa) for LoRa devices to improve their receivers' sensitivity by leveraging deep neural networks. MaLoRaGW is orthogonal and complementary to this research line.

LoRa for Sensing. LoRa has recently shown a great potential to enable long-range sensing applications [91–93]. In [66], Zhang et al. proposed a LoRa-based sensing model to enable through-the-wall human activity sensing over long (e.g., 25 m) distances. In [94], Xie et al. proposed Sen-fence, a LoRa-based multi-antenna and multi-gateway system, that jointly enhances the sensing range and weakens the impact of undesired movements in the sensing field of view using virtual beamforming techniques. In [95], Chen et al. proposed WideSee, a contactless sensing scheme. It leverages a LoRa transceiver mounted on a drone to enable target human detection and localization over long ranges. MaLoRaGW does not belong to this research area but may provide insights for future LoRa sensing by leveraging multiple antennas.

5.3 A Primer of LoRa

LoRa is a wireless system that offers long-range and low-power wireless connectivity for IoT devices. LoRaWANs are typically configured to a star network topology, where a centralized gateway serves LoRa devices in both uplink and downlink. In North America, LoRa operates in 915 MHz frequency bands, where 64 channels of 125 kHz and 8 channels of 500 kHz are specified for upstream, and 8 channels of 500 kHz are specified for downstream. LoRa devices support data rates ranging from 980 bps to 21.9 kbps, depending on the channel bandwidth and the modulation spreading factor [96].

Chirp Spread Spectrum Modulation. LoRa uses CSS technique for data modulation. CSS is a spreading technique that linearly sweeps the entire channel bandwidth within a symbol duration. Denote f_{sym} and f_s as the symbol frequency and the sampling frequency, respectively. Also, denote $N = 2^{\text{SF}}$ as the symbol duration, where SF is the spreading factor. LoRa supports spreading factors from 6 to 12, depending on the channel bandwidth and the required data rate. The transmitted



(a) (b)

Figure 5.1: The spectrogram of CSS modulation: (a) the base up-chirp symbol, (b) the modulated symbol '1010011101' (669 in decimal), where $f_{sym} = \frac{669}{2^{10}} \times 500$ kHz ≈ 326.7 kHz.

8 Symbols	4.25 Symbols	8 Symbols		L Bytes	2 Bytes
Preamble	Synchronization word	PHDR	PHDR_CRC	PHY payload	CRC (uplink only)

Figure 5.2: The frame structure for LoRa communications.

symbol using CSS modulation can be written as [81]:

$$x[n] = c[n]e^{j2\pi \frac{f_{sym}}{f_s} n}, \qquad 0 \leq n < N, \tag{5.1}$$

where $c[n] = e^{j2\pi(\frac{n^2}{2N} + \frac{f_0}{f_s} n)}$ is the base up-chirp signal and f_0 is the initial base up-chirp frequency. Fig. 5.1 shows the spectrogram of the based chirp signal and a modulated symbol (e.g., '1010011101' = 669) using CSS modulation, when SF $= 10$, $f_s = 500$ kHz, and $f_0 = 0$ kHz.

LoRa Packet Structure. Fig. 5.2 shows the frame structure used in LoRa communications. The frame consists of a preamble, a synchronization word, an optional PHY header and its CRC, and PHY payload. The preamble consists of 8 up-chirp symbols, followed by 2 more up-chirp symbols and 2.25 down-chirp symbols as the synchronization word. The preamble and synchronization word are used for frame extraction and synchronization purposes. The PHY header (PHDR) and PHY header CRC (PHDR_CRC) are optional. When used, they indicate the length of payload, the coding rate, and the presence of CRC for payload. The PHY payload carries MAC header

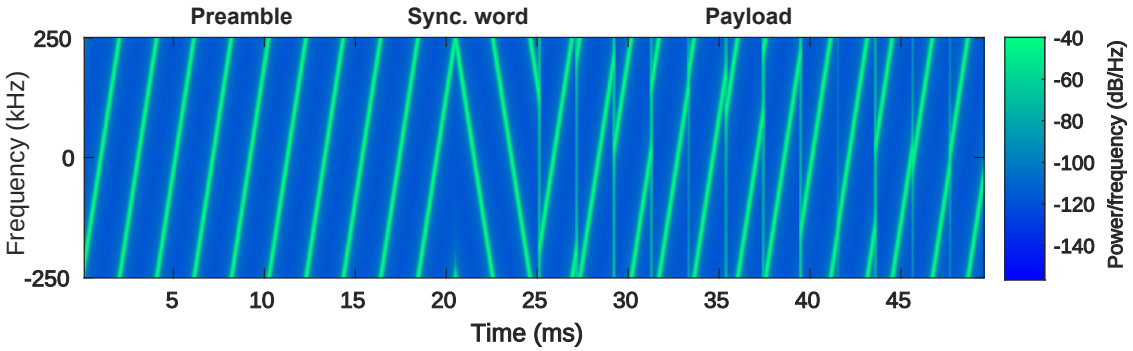


Figure 5.3: The transmitted LoRa frame.

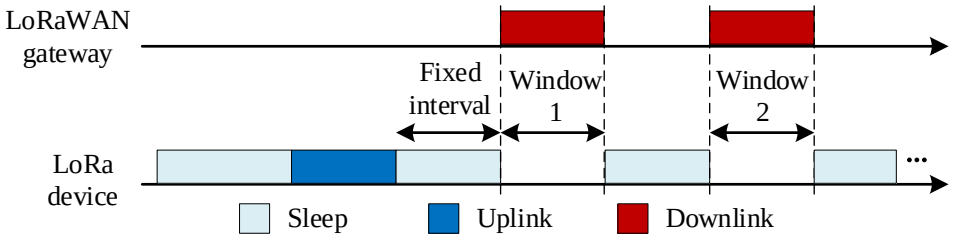


Figure 5.4: LoRa medium access protocol.

and MAC data. The maximum payload size depends on the data rate, which is determined by the spreading factor and channel bandwidth. The LoRa packet is then modulated by CSS modulation and transmitted over the air, as exemplified in Fig. 5.3.

LoRaWAN Medium Access. Fig. 5.4 shows the medium access protocol for LoRa devices in a LoRaWAN. A LoRa device uses ALOHA protocol to access the medium. It can initiate uplink transmissions once it wakes up and has data to send. Upon the completion of uplink transmissions, it turns to sleep for a fixed time interval ($\sim$1 s) and opens up two time windows for data reception. The LoRa device receives the downlink information (e.g., ACK packets and gateway commands) within these two time windows and turns to sleep for the rest of the time [97].

Concurrent Transmission. While LoRaWAN already supports concurrent packet transmission in uplink [75, 78, 81, 82], it cannot support concurrent transmission in downlink. This is because the packets from a gateway will have the same features in time and frequency domains. If a gateway

adds two packets together and sends them to two users, then the two users will decode the same packet (the one with a stronger power) rather than their own packets.

5.4 Understanding MU-MIMO

MU-MIMO is a key technology for OFDM-based wireless networks such as cellular [38, 98] and Wi-Fi [99, 100] networks. It has been widely deployed in real-world wireless communication systems and demonstrated a significant gain of spectrum efficiency. MU-MIMO exploits the SDoF provided by multiple antennas to separate signal streams, making it possible for a gateway (a.k.a., access point or base station) to support multiple concurrent data packet transmissions in both uplink and downlink.

Uplink Transmission. Fig. 5.5(a) shows uplink MU-MIMO transmission. Denote K as the number of user devices and M as the number of antennas at the gateway. Denote $\mathbf{H}_{\text{ul}} \in \mathbb{C}^{M\times K}$ as the compound uplink channel matrix. Mathematically, the received signal at the gateway, $\mathbf{y}_{\text{gw}} \in \mathbb{C}^{M\times 1}$, can be written as:

$$\mathbf{y}_{\text{gw}} = \mathbf{H}_{\text{ul}}\mathbf{x}_{\text{ul}} + \mathbf{n}, \tag{5.2}$$

where $\mathbf{x}_{\text{ul}} \in \mathbb{C}^{K\times 1}$ is the vector of transmitted signals from user devices, and $\mathbf{n} \in \mathbb{C}^{M\times 1}$ is the additive noise vector. To decode $\mathbf{x}_{\text{ul}}$, the gateway first estimates channel matrix $\mathbf{H}_{\text{ul}}$ and then uses the estimated channel to design an equalizer for signal detection.

Downlink Transmissions. Fig. 5.5(b) shows downlink MU-MIMO transmission. Denote $\mathbf{H}_{\text{dl}} \in \mathbb{C}^{K\times M}$ as the compound downlink channel matrix. Denote $\mathbf{P} \in \mathbb{C}^{M\times K}$ as the beamforming matrix used by the gateway. The vector of signals received by all user device can be written as:

$$\mathbf{y}_{\text{ud}} = \mathbf{H}_{\text{dl}}\mathbf{P}\mathbf{x}_{\text{dl}} + \mathbf{n}, \tag{5.3}$$

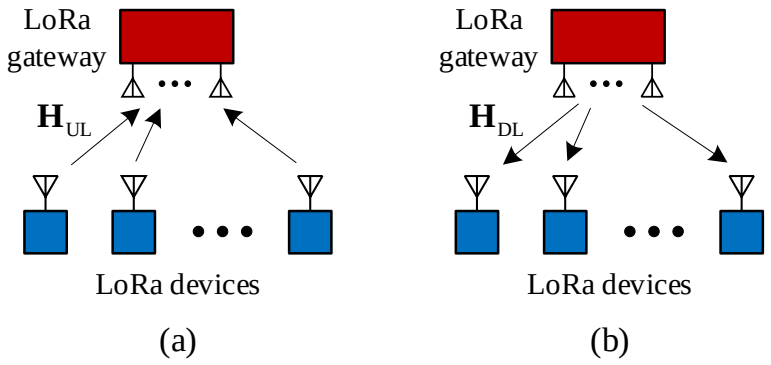


Figure 5.5: MU-MIMO transmission: (a) uplink; (b) downlink.

where $\mathbf{x}_{\text{dl}} \in \mathbb{C}^{K\times 1}$ is the transmitted symbol vector at the gateway for K user devices, and $\mathbf{n}$ is noise vector at user devices. In cellular and Wi-Fi networks, protocols have been specified for a gateway to obtain $\mathbf{H}_{\text{dl}}$. With the downlink channel knowledge, precoders such as ZF and MMSE can be used to separate data streams for different users in the spatial domain. Particularly, the ZF precoder is constructed by letting $\mathbf{H}_{\text{dl}}\mathbf{P} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Through the transmitter-side precoding, inter-user inference is pre-cancelled and each user device can decode its desired signal.

5.4.1 Challenges and Our Approach

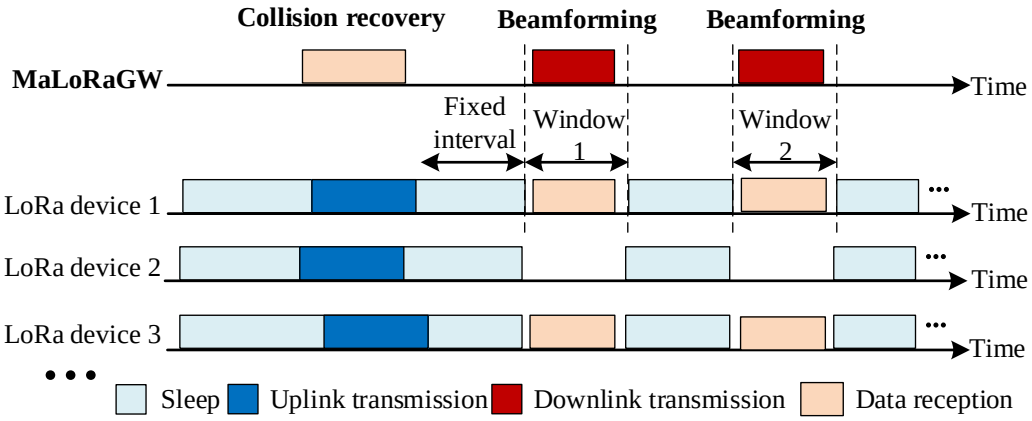
MU-MIMO has received a great success in cellular and Wi-Fi networks, and it is one of key technologies for 5G/6G networks. However, while there are some prior works on studying MIMO for LoRaWAN, little research work has been done for the design of MU-MIMO schemes in LoRaWAN. This stagnation underscores the grand challenges in the realization of MU-MIMO for LoRaWAN, which we describe as follows.

Uncoordinated Transmission in Uplink. In cellular and Wi-Fi (e.g., 802.11ax) systems, dedicated MU-MIMO protocols have been specified in the standards to coordinate user devices for concurrent uplink transmission. User devices' packets are aligned in both time and frequency when

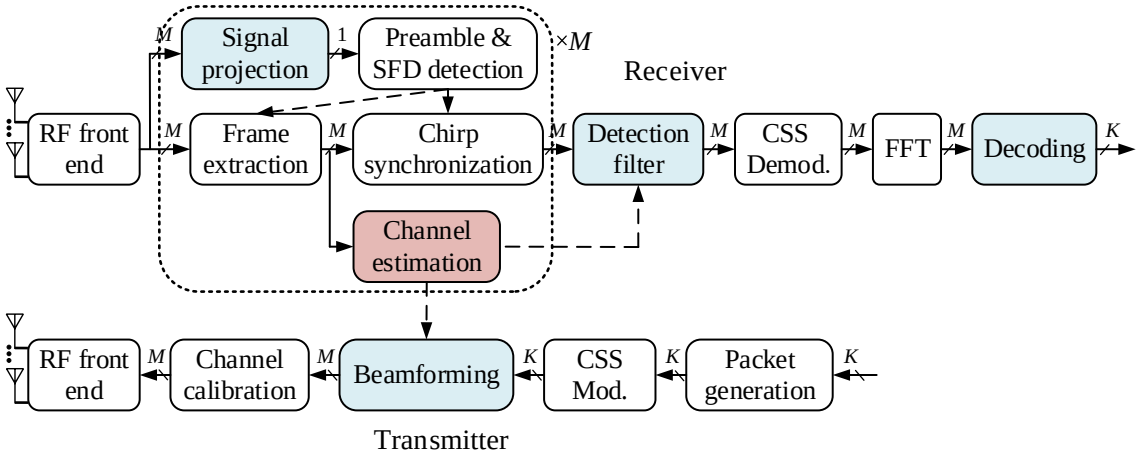
arriving at the gateway. The time and frequency alignments make it easy for a gateway to detect the collided packets. In addition, user devices in these systems carry orthogonal pilots (e.g., VHT in 802.11 [101] and demodulation reference signals in LTE [102]) in their packets. The orthogonal pilots allow a gateway to estimate the channels between itself and user devices, which play a critical role in the detection of concurrent packets. Unfortunately, LoRaWAN does not have such luxuries for MU-MIMO transmissions. It uses ALOHA protocol for medium access control, which does not support coordination among user devices [103]. The collided packets at a gateway could be fully or partially overlapping, and they may bear different carrier frequency offsets and chirp timing offsets.

Channel Acquisition in Downlink. In cellular and Wi-Fi systems, protocols have been specified for channel feedback. For example, each user involved in downlink MU-MIMO transmission will estimate its downlink channel and report the estimated channel (after quantization and compression) to an access point, which then constructs precoding vectors for downlink beamforming [104]. Such a channel sounding protocol requires cooperation from user devices. However, LoRa devices are typically of low cost, low power, and low computation. It is impractical to require LoRa devices to estimate and report their channels. In addition, one of our design objectives is to maintain backward compatibility with incumbent LoRa devices. Requiring channel feedback clearly contradicts our design objective.

Our Approach. To address the above challenges, MaLoRaGW employs a joint design for uplink packet detection and downlink beamforming. The joint design features PCA-based signal projection, robust channel estimation, and implicit beamforming. Different from OFDM systems, LoRa has collision recovery capability even if a gateway has a single antenna [75, 78–80, 82]. Therefore, MaLoRaGW aims to enhance the collision recovery capability of a LoRa gateway by jointly exploiting signal features in spatial, frequency, and time domains. We note that, since



(a) Proposed protocol for MaLoRaGW's MU-MIMO transmission.



(b) Proposed PHY design for MaLoRaGW's MU-MIMO transmission.

Figure 5.6: Joint protocol and PHY design for MaLoRaGW to support both uplink and downlink MU-MIMO transmissions.

LoRa is not a single-tone frequency channel as OFDM systems, completely mitigating inter-user interference may not be possible. However, it is still possible to support multiple packet transmission in practice, thanks to the interference resilience of CSS modulation.

5.5 Design

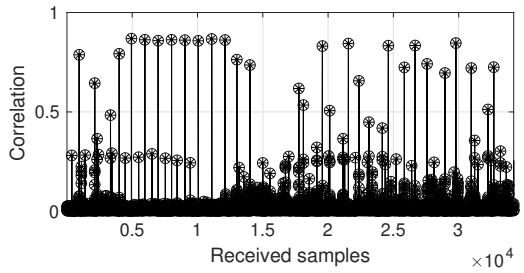
MaLoRaGW is an M-antenna LoRa gateway to enhance its packet detection in uplink and enable its concurrent transmission in downlink. Fig. 5.6(a) shows its MAC protocol, and Fig. 5.6(b) shows

its PHY-layer signal processing diagram. In Fig. 5.6(b), the colored modules are our new design, and the rest are LoRa's legacy modules. The Rx chain is to decode collided packets from multiple user devices, while the Tx chain is to deliver data packets to the same set or a subset of those user devices. MaLoRaGW can decode more than M collided packets in uplink by jointly leveraging the signal features in spatial, frequency, and time domains. However, MaLoRaGW can deliver at most M concurrent packets in downlink, as the downlink transmission relies solely on the spatial degrees of freedom to separate signals.

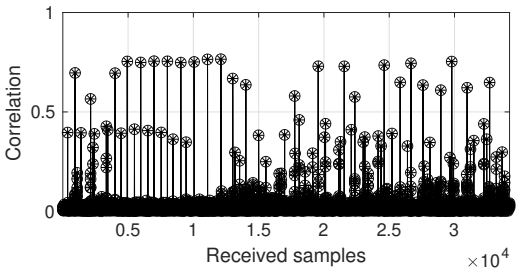
5.5.1 Overview

A key component of our design is channel estimation. Prior works on LoRa collision recovery [75–77, 79–81] exploit CSS signal features in frequency, time, and/or power domains to decode packets, and thus do not require to have CSI. In contrast, CSI plays a key role in MU-MIMO transmissions. In uplink, CSI is needed to separate strong and weak signals from their collided packets, so that weak signal can be decoded by leveraging its spatial features. In downlink, CSI is needed to construct precoding vectors for beamforming, which is the enabler of downlink concurrent packet transmission.

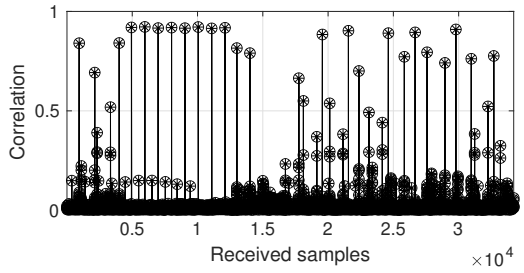
To estimate the CSI, some system imperfections must be corrected first. Like other wireless communication systems, LoRa receivers suffer from CFO and chirp timing offset (CTO) in their channel estimation. The frequency and timing offsets must be corrected prior to channel estimation. To do so, a PCA-based signal projection method is proposed for frequency and timing offset estimation. The projection is made in the spatial domain. It is effective to separate weak and strong signals from collided packets, making it possible to correct CFO/CTO for both weak and strong packets and therefore accurately estimate their channels. The estimated channels are then used for two purposes: i) enhance uplink packet detection and ii) construct precoding vectors for downlink



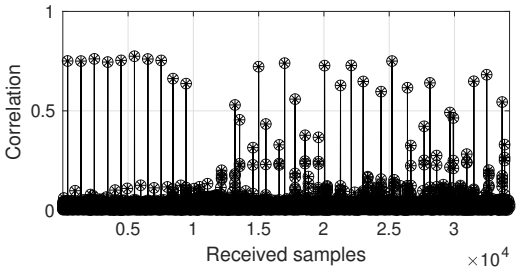
(a) Correlation result on ant 1.



(b) Correlation result on ant 2.



(c) Correlation result after signal projection for strong user.



(d) Correlation result after signal projection for weak user.

Figure 5.7: Correlation results before and after projection.

beamforming.

5.5.2 Preamble Detection

LoRa receivers use the cross correlation between a base up-chirp and the received signal to detect the preamble of a packet. Although cross correlation is resilient to interference, it does not work well when the interference is much stronger than the signal of interest. Consider a LoRa gateway equipped with two antennas. It receives two collided packets, which have 20 dB difference in their signal strengths. Fig. 5.7(a-b) shows the cross-correlation results of the received signals at the gateway's antennas. It can be seen that the cross correlation has a poor performance for the weak user as it is significantly corrupted by the strong interference. The poor performance of preamble detection always leads to a failure of packet detection. In what follows, we present our treatment to this problem.

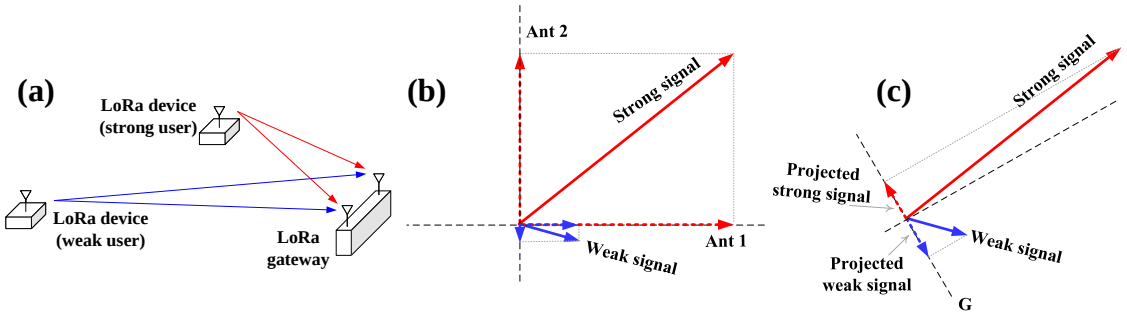


Figure 5.8: Illustration of spatial projection for collided packets. (a) A case of collided packet; (b) signal strengths on two antennas; (c) signal projection.

Basic Idea. When a gateway has multiple antennas, the spatial degrees of freedom can be leveraged to improve the performance of preamble detection. Consider a gateway equipped with two antennas as shown in Fig. 5.8(a). It receives packets from two collided packets from two users. One user is close to gateway, while the other is far from gateway. To detect the weak signal from the distant user, a natural approach is to check the signal stream from each antenna and find the best one to perform cross correlation. However, this approach does not work well. This is because the antennas are close to each other and tend to have similar SNR, as illustrated in Fig. 5.8(b). To address this issue, we propose a spatial projection approach by leveraging the multiple antennas on a gateway. Fig. 5.8(c) illustrated the basic idea of our approach. If we can find a good spatial vector ($\mathbf{G}$ in Fig. 5.8(c)), the projected signal on $\mathbf{G}$ will have similar strength as the projected interference. The question to ask is how to find a good projection vector $\mathbf{G}$. The challenge here is that the gateway does not have channel knowledge yet, and it must find a vector without channel knowledge.

PCA-based Spatial Projection. MaLoRaGW employs PCA to construct the projection vectors. While PCA is widely used to reduce the dimension of data, here it is used to separate strong and weak signals so that weak signal can be detected. It works as follows. MaLoRaGW first computes the covariance matrix of the received signal streams by $\mathrm{R}_{\mathrm{yy}} = \sum_i \mathbf{y}_{\mathrm{gw}}(i)\mathbf{y}_{\mathrm{gw}}(i)^{\mathsf{H}} \in \mathbb{C}^{M\times M}$,

where i is signal sample index. Then, it performs $[\mathbf{Q}, \Lambda, \mathbf{Q}^{-1}] = \mathrm{svd}(\mathbf{R}_{\mathrm{yy}})$, where $\mathrm{svd}(\cdot)$ is the SVD operator, and $\Lambda \in \mathbb{R}^{M \times M}$ is a diagonal matrix carrying the singular values of $\mathbf{R}_{\mathrm{yy}}$ in non-increasing order. The columns in $\mathbf{Q}$ represent the singular vector of R_{yy}. If the number of gateway's antennas is greater than that of collided packets, $\mathbf{Q}$ can be further divided to a signaling subspace $\mathbf{Q}_s \in \mathbb{C}^{M \times K}$ and a null subspace $\mathbf{Q}_n \in \mathbb{C}^{M \times (M-K)}$, i.e., $\mathbf{Q} = [\mathbf{Q}_s | \mathbf{Q}_n]$. The principal components of received signals can be calculated as: $\mathbf{y}_p = \mathbf{Q}_s^{\mathsf{H}} \mathbf{y}_{\mathrm{gw}}$, where $\mathbf{y}_p$ is projected signals.

The signal space includes the basis corresponding to non-zero eigenvalues, which is highly correlated with $\mathbf{H}_{\mathrm{UL}}$. We use the signal space to project the received signal into the signals' direction to strengthen the desired signal SNR and reduce the interference signal. Mathematically, the projected signal for each user can be expressed as: $\mathbf{y}_p = \mathbf{q}_s^{\mathsf{H}} \mathbf{y}_{\mathrm{gw}}$, where $\mathbf{q}_s$ is a column of $\mathbf{Q}_s$ and it is the signal basis corresponding to the user.

To see the effectiveness of the projection, let us reconsider the example presented in Fig. 5.7(a-b). We compute SVD on the received signals and then perform cross correlation on the primary components of the received signals. Fig. 5.7(c-d) shows the cross correlation results. It can be seen that the correlation peak is significant, indicating that the weak signal can be successfully detected.

Preamble Search. After signal projection, MaLoRaGW performs cross correlation between the base up-chirp symbol and the primary components of received signals (i.e., $\mathbf{y}_p$). It searches for the beginning of the preamble by solving the following problem:

$$\hat{\zeta} = \arg\max_{\zeta} \sum_{l=1}^{L} \left| \sum_{n=0}^{N-1} y_p[n + \zeta + lN] c^{-1}[n] \right|, \tag{5.4}$$

where $L = 8$ is the number of base up-chirps in LoRa preamble, and $c[n]$ is the base up-chirp signal.

5.5.3 Channel Estimation

Consider a point-to-point LoRa communication link, where both transmitter and receiver are equipped with a single antenna. Denote h as the channel between the transmitter and the receiver. Assume that LoRa signal experiences a flat fading channel when traversing from the transmitter to the receiver. This is true in most cases as the LoRa signal has a narrow bandwidth (i.e., ≤ 500 kHz) [81]. If the transmitter and receiver are perfectly synchronized in time and frequency and noise is negligible, then the received chirp can be written as:

$$y[n] = h \cdot c[n] e^{j2\pi \frac{f_{sym}}{f_s} n}, \qquad \text{for } 0 \leq n < N, \tag{5.5}$$

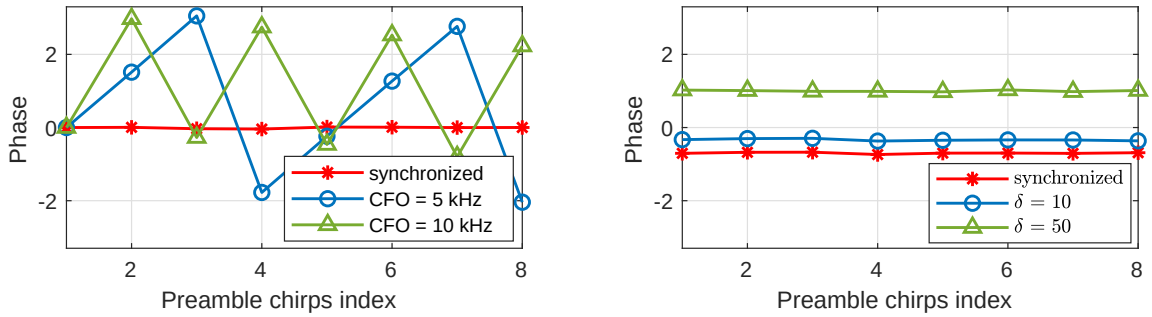
where f_{sym} and f_s are the symbol frequency and sampling frequency, respectively.

By demodulating the received chirp and applying FFT to the demodulated signals, the channel can be estimated as:

$$\widehat{h} = h[m = \frac{f_{sym}}{f_s}] = \frac{1}{N} \sum_{n=0}^{N-1} c^{-1}[n] y[n] e^{-j2\pi \frac{mn}{N}}. \tag{5.6}$$

However, estimating channel in LoRa is not that simple. Like other communication systems, LoRa receivers suffer from CFO and CTO, which must be estimated and corrected in order to estimate channel coefficients. In what follows, we focus on the estimation of CFO and CTO of chirps.

Carrier Frequency Offset (CFO). CFO is mainly caused by the frequency mismatch between transmitter and receiver. Denote Δf as the CFO between transmitter and receiver. Mathematically,



(a) The impact of CFO.

(b) The impact of CTO.

Figure 5.9: The measured phases of the decoded preamble chirps in practice.

the lth received chirp at the gateway in the presence of CFO can be expressed as:

$$y_l[n] = he^{j2\pi\frac{\Delta f}{f_s}lN}c[n]e^{j2\pi\frac{f_{sym}+\Delta f}{f_s}n}, \tag{5.7}$$

for $0 \leq n < N$ and $0 \leq l < L$, where L here denotes the total number of chirps in one frame. Then, the channel estimation in (5.6) can be rewritten as:

$$\widehat{h} = h[m = \frac{f_{sym} + \Delta f}{f_s}] = \frac{1}{N}e^{-j2\pi\frac{\Delta f}{f_s}lN}\sum_{n=0}^{N-1} c^{-1}[n]y_l[n]e^{-j2\pi\frac{mn}{N}}, \tag{5.8}$$

for $0 \leq l < L$. From (5.8), it can be seen that the CFO induces a phase shift to the received chirps. This phase shift changes linearly over the consecutive received chirps. Fig. 5.9(a) shows the measured phase of the decoded preamble chirps when the CFO between transmitter and receiver is 0kHz, 5kHz, and 10kHz. It can be seen that the unwrapped phase indeed increases linearly with the chirp index. Based on (5.8), we present a two-step approach to estimate the CFO.

Coarse CFO Correction. We take advantage of up-chirp and down-chirp in LoRa frame to estimate the CFO. We first measure the frequencies of preamble up-chirp symbols and synchronization down-chirp symbols in the received signal, and then use the measured symbol frequencies

to infer the CFO. Nevertheless, the inference is not straightforward because the measured symbol frequencies may be caused by CFO, CTO, or both. Fortunately, the CFO and CTO have different impacts on the measured symbol frequencies. Specifically, a positive CFO will result in a positive frequency shift on both up-chirp and down-chirp symbols, while a positive CTO will cause a positive frequency shift on up-chirp symbols and a negative frequency shift on down-chirp symbols. This difference makes it possible to infer the CFO based on the measured (up-chirp and down-chirp) symbol frequencies. Based on this observation, we estimate the CFO as follows:

$$\Delta\widehat{f}_{coarse} = \frac{1}{2}\Big(\frac{1}{8}\sum_{l=1}^{8} z_l^{\mathrm{up}} + \frac{1}{2}\sum_{l=1}^{2} z_l^{\mathrm{dw}}\Big), \tag{5.9}$$

where z_l^{up} is the measured frequency of the lth up-chirp symbol in preamble, and z_l^{dw} is the measured frequency of the lth down-chirp symbol in synchronization word. Once $\Delta\widehat{f}_{coarse}$ is calculated, the estimated channel is updated as follows: $\widehat{h}_l \leftarrow \widehat{h}_l \cdot e^{j2\pi\frac{\Delta\widehat{f}_{coarse}}{fs}lN}$ for $0 \le l < L$.

Fine CFO Correction. The CFO estimated in (5.9) is not accurate enough. Although the residual CFO is small, it leads to a phase shift accumulated over preamble symbols. This phase shift will degrade uplink packet detection and downlink beamforming. To estimate the residual CFO, we search a narrow frequency offset range (e.g., -10 Hz to 10 Hz) and find the frequency offset, $\Delta\widehat{f}_{fine}$, that minimizes the phase variance of estimated channels over preamble symbols. Denote $\phi_l = \angle\big(\widehat{h}_l \cdot e^{j2\pi\frac{\Delta f}{fs}lN}\big)$ as the phase of the estimated channel coefficient for the lth preamble up-chirp symbol. Then, the residual CFO is calculated by:

$$\Delta\widehat{f}_{fine} = \arg\min_{\Delta f} \sigma_\Phi^2 \ , \tag{5.10}$$

where σ_Φ^2 is the variance of the phase vector $\Phi = [\phi_1, \phi_2, \ldots, \phi_L]$.

Chirp Timing Offset (CTO). CTO refers to the timing misalignment between the received chirps and the applied demodulation (dechirp) window. Denote δ as CTO. Then, the demodulated chirp can be written as:

$$c^{-1}[n-\delta]y[n] = h \cdot e^{-j2\pi\phi} e^{j2\pi(\frac{f_{sym}}{f_s}+\frac{\delta}{N})n}, \tag{5.11}$$

for $\delta \leq n < N+\delta$, where $\phi = \frac{\delta^2}{2N} - \frac{f_0}{f_s}\delta$. Then, the estimated channel (5.6) can be re-written as:

$$\widehat{h} = h[m = \frac{f_{sym}}{f_s} + \frac{\delta}{N}] = \frac{1}{N} e^{j2\pi\phi} \sum_{n=0}^{N-1} c^{-1}[n-\delta]y[n] e^{-j2\pi\frac{mn}{N}}. \tag{5.12}$$

From (5.12), it can be seen that the CTO introduces an additional phase shift ϕ to the observed channel. This phase shift is constant for all chirps in preamble and can be inferred based on δ. Fig. 5.9 shows the measured phase of the demodulated preamble chirps, when $\delta = 0$, $\delta = 10$, and $\delta = 50$. Fortunately, as explained before, the CTO can be estimated by:

$$\widehat{\delta} = \frac{N}{2f_s} \cdot \Big(\frac{1}{8}\sum_{l=1}^{8} z_l^{\text{up}} - \frac{1}{2}\sum_{l=1}^{2} z_l^{\text{dw}}\Big). \tag{5.13}$$

Based on the estimated CTO, the channel phase shift caused by CTO can be calculated by: $\phi = \frac{\widehat{\delta}^2}{2N} - \frac{f_0}{f_s}\widehat{\delta}$, which is then used to correct the estimated channel.

Channel Estimation for Collided Packets. While the above channel estimation was presented for a LoRa receiver when decoding an interference-free (collision-free) packet, it can be extended to the case where packets collide. When the desired preamble symbols are interfered by the symbols of other packets, MaLoRaGW traces the frequency of decoded symbols within the first 10 demodulation windows and identifies the peaks associated with the desired preamble symbols. To do so, MaLoRaGW first uses a low pass filter to mitigate the frequency components of undesired

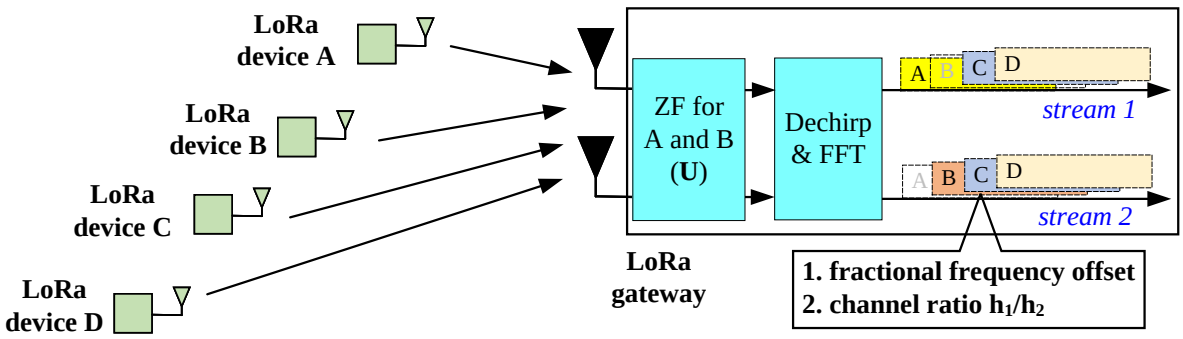


Figure 5.10: Illustrating the decoding algorithm for MaLoRaGW in uplink.

symbols. The rationale behind this operation is that, after frame synchronization, the frequencies of the desired preamble symbols are around zero. Yet, the output of low-pass filter may still include the low frequency components of interfering chirps. MaLoRaGW then uses a maximum likelihood method to search for the frequency peaks with the highest correlation value. This operation is based on the fact that the frequency shift caused by CFO and CTO is almost identical for the same user but different for different users. After identifying the frequencies of the desired preamble symbols, the above channel estimation method is applied to estimate the channel between the gateway's each antenna and the user device.

5.5.4 Decoding Algorithm

Our decoding algorithm comprises the following steps: spatial signal projection, chirp demodulation and FFT operation, and peak clustering. Fig. 5.10 shows an example of our decoding diagram. In what follows, we explain them in detail.

Spatial Signal Projection. The purpose of spatial signal projection is to alleviate the near-far effect so as to improve the signal detection accuracy. Here, we use ZF as the projection method. Denote $K' = \min(K, M)$, where K is the number of LoRa devices, and M is the number of gateway's antennas. We select K' out of K LoRa devices based on the strength of

their estimated channel coefficients. Denote $\mathbf{H}_{\text{ul}} \in \mathbb{C}^{M \times K'}$ as the estimated uplink channel matrix for those selected K' LoRa devices. Then, we construct the ZF projection matrix by letting $\mathbf{U} = (\mathbf{H}_{\text{ul}}^{\mathsf{H}}\mathbf{H}_{\text{ul}})^{-1}\mathbf{H}_{\text{ul}}^{\mathsf{H}}$, where $(\cdot)^{\mathsf{H}}$ is conjugate transpose operator, and apply it to the signal streams.

The ZF projection will reduce inter-user interference. Consider Fig. 5.10 for example. The gateway receives strong signals from A and B and weak signals from C and D. It constructs the ZF matrix using A's and B's channel coefficients. This ZF matrix will mitigate the signal from A for stream 1 and mitigate the signal from B for stream 2, making it easier to decode weak signals from C and D.

FFT Peak Clustering. After chirp demodulation and FFT operations, the FFT outputs will have multiple amplitude peaks in the demodulation window. These peaks may correspond to different LoRa devices and their different chirps (due to the lack of inter-user synchronization). It is critical to identify each FFT peak and find out its corresponding source device and corresponding chirp. To do so, we jointly exploit two features of each FFT peak for decoding collided packets: *Time and frequency offsets* and *channel ratio*.

i) Time and frequency offsets: All chirps from the same LoRa device have a unique fractional frequency offset [81], which can be used to identify an FFT peak's source device. For each of the output streams, MaLoRaGW exploits the frequency and time features of each FFT peak to disentangle collided packets. Specifically, MaLoRaGW first identifies the FFT peaks in consecutive demodulation windows and then removes those peaks with the same frequency in two consecutive demodulation windows. The rationale behind this operation is twofold. First, the chirps from different LoRa devices are very likely misaligned in time. Second, when an interfering chirp is misaligned with the demodulation window, its energy is proportionally split over two consecutive demodulation windows; however, its frequencies will be the same in the two demodulation windows.

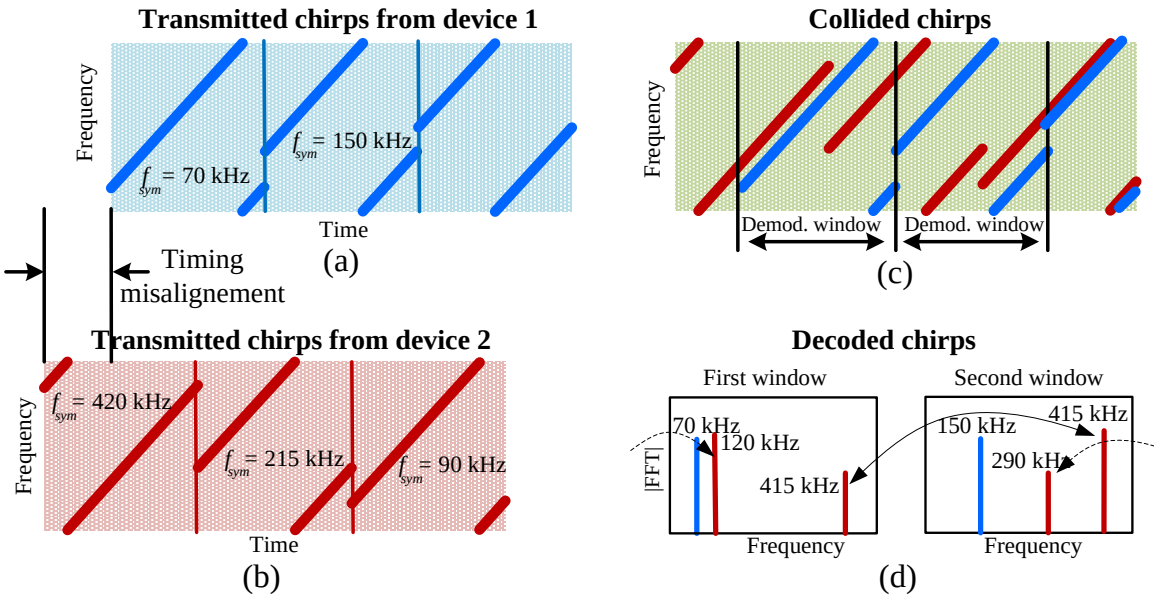


Figure 5.11: Illustration of demodulating collided chirps that are misaligned in time.

Fig. 5.11 illustrates the use of time and frequency features of CSS chirps to disentangle collided packets. In this example, the chirp signals from the two LoRa devices are misaligned in time. When the demodulation window is aligned with the chirps from the first LoRa device, it misaligns with the chirps from the second LoRa device. A misaligned chirp will generate two peaks of the same frequency in two consecutive demodulation windows. Based on this observation, MaLoRaGW searches for the peaks of the same frequency in consecutive demodulation windows and removes those peaks before decoding the aligned chirps.

ii) Channel ratio: In addition to the fractional frequency offset, MaLoRaGW also uses the channel ratio as another feature for the classification and clustering of FFT peaks. Denote $[x_1, x_2, \ldots, x_M]$ as an FFT peak observed from the M signal streams at the gateway, where M is the number of its antennas. Then, $[1, \frac{x_2}{x_1}, \ldots, \frac{x_M}{x_1}]$ represents the vector of channel ratios over the gateway's M antennas, which is relatively independent of the data bit, chirp misalignment, CFO, CTO, etc. Therefore, MaLoRaGW uses this vector as another feature to classify the FFT peaks. We note that this feature is similar to that in [88], where phase difference is used for peak

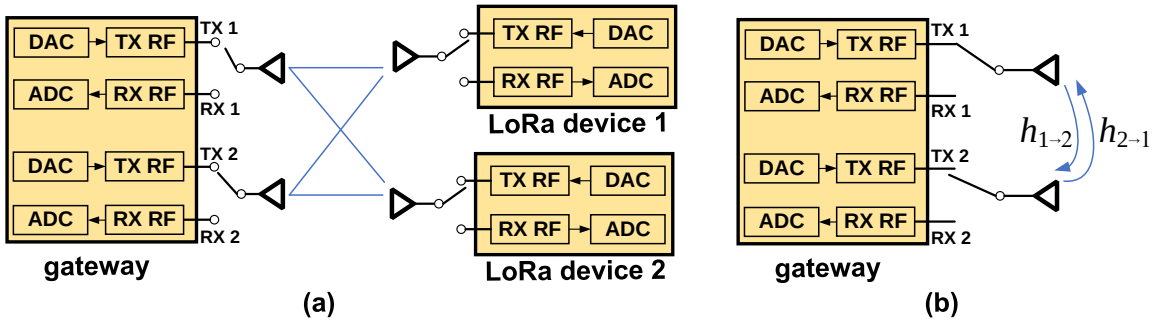


Figure 5.12: (a) Illustrating the relation between uplink and downlink channel in a two-user MIMO case. (b) Illustrating RF calibration at gateway.

classification instead of channel ratio.

Decoding Collided Packets. Once the FFT peaks are classified and clustered, it is straightforward for MaLoRaGW to decode the collided packets.

5.5.5 Downlink Beamforming

In this subsection, we present our approach for downlink beamforming. It comprises three steps: i) select a subset of LoRa devices for downlink MU-MIMO transmission; ii) infer downlink channels between the gateway and the selected LoRa devices; iii) precode baseband signals at the gateway. In what follows, we present them in detail.

User Selection. While an M-antenna gateway can decode more than M collided uplink packets, it can send at most M concurrent packets to user devices in downlink. If more than M user devices are involved in the uplink transmission, how to select a subset of user devices for downlink MU-MIMO transmission is an open problem. This problem is not in the scope of this work and will be studied in future. In this work, MaLoRaGW simply selects the $K' = \min(K, M)$ LoRa devices with strongest uplink signal strength for downlink MU-MIMO transmission.

Downlink Channel Inference. Another question we need to address is the difference between uplink and downlink channels. What we have is uplink channel (i.e., $\mathbf{H}_{\text{ul}}$); what we need is

downlink channel (i.e., $\mathbf{H}_{\text{dl}}$). In what follows, we use a small case when $K' = 2$ and $M = 2$ to illustrate our approach for the inference of downlink channel. But our approach is generic and can apply to a general case where $K' \leq M$.

Referring to Fig. 5.12(a), at a wireless receiver, the observed/estimated channel coefficient comprises transmitter's RF response, over-the-air response, and receiver's RF response. At the gateway, the observed/estimated uplink channel $\mathbf{H}_{\text{ul}}$ can be written as:

$$\begin{aligned}\mathbf{H}_{\text{ul}} &= \begin{bmatrix} R_1^{\text{grx}} R_{11}^{\text{ota}} R_1^{\text{ltx}} & R_1^{\text{grx}} R_{12}^{\text{ota}} R_2^{\text{ltx}} \\ R_2^{\text{grx}} R_{21}^{\text{ota}} R_1^{\text{ltx}} & R_2^{\text{grx}} R_{22}^{\text{ota}} R_2^{\text{ltx}} \end{bmatrix} \\ &= \begin{bmatrix} R_1^{\text{grx}} & 0 \\ 0 & R_2^{\text{grx}} \end{bmatrix} \begin{bmatrix} R_{11}^{\text{ota}} & R_{12}^{\text{ota}} \\ R_{21}^{\text{ota}} & R_{22}^{\text{ota}} \end{bmatrix} \begin{bmatrix} R_1^{\text{ltx}} & 0 \\ 0 & R_2^{\text{ltx}} \end{bmatrix}, \end{aligned} \tag{5.14}$$

where R denotes response and its superscript/subscript denotes the corresponding device. Specifically, 'g' and 'l' in its superscript represent gateway and LoRa device, respectively; 'tx' and 'rx' represent transmitter and receiver, respectively; numbers in its subscript represent either gateway's antenna index or LoRa device index. Superscript 'ota' represents the corresponding over-the-air channel response.

While the compound uplink and downlink channels are not reciprocal, their over-the-air components are identical in the coherence time. Therefore, we have

$$\begin{aligned}\mathbf{H}_{\text{dl}} &= \begin{bmatrix} R_1^{\text{lrx}} R_{11}^{\text{ota}} R_1^{\text{gtx}} & R_1^{\text{lrx}} R_{21}^{\text{ota}} R_2^{\text{gtx}} \\ R_2^{\text{lrx}} R_{12}^{\text{ota}} R_1^{\text{gtx}} & R_2^{\text{lrx}} R_{22}^{\text{ota}} R_2^{\text{gtx}} \end{bmatrix} \\ &= \begin{bmatrix} R_1^{\text{lrx}} & 0 \\ 0 & R_2^{\text{lrx}} \end{bmatrix} \begin{bmatrix} R_{11}^{\text{ota}} & R_{12}^{\text{ota}} \\ R_{21}^{\text{ota}} & R_{22}^{\text{ota}} \end{bmatrix}^{\top} \begin{bmatrix} R_1^{\text{gtx}} & 0 \\ 0 & R_2^{\text{gtx}} \end{bmatrix}. \end{aligned} \tag{5.15}$$

Based (5.14) and (5.15), we have:

$$\begin{aligned}\mathbf{H}_{\mathrm{dl}} &= \begin{bmatrix} \frac{R_1^{\mathrm{lrx}}}{R_1^{\mathrm{ltx}}} & 0 \\ 0 & \frac{R_2^{\mathrm{lrx}}}{R_2^{\mathrm{ltx}}} \end{bmatrix} \mathbf{H}_{\mathrm{ul}}^{\top} \begin{bmatrix} \frac{R_1^{\mathrm{gtx}}}{R_1^{\mathrm{grx}}} & 0 \\ 0 & \frac{R_2^{\mathrm{gtx}}}{R_2^{\mathrm{grx}}} \end{bmatrix} \\ &= \underbrace{\frac{R_1^{\mathrm{lrx}} R_1^{\mathrm{gtx}}}{R_1^{\mathrm{ltx}} R_1^{\mathrm{grx}}}}_{(a)} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \frac{R_2^{\mathrm{lrx}} R_1^{\mathrm{ltx}}}{R_2^{\mathrm{ltx}} R_1^{\mathrm{lrx}}} \end{bmatrix}}_{(b)} \mathbf{H}_{\mathrm{ul}}^{\top} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \frac{R_1^{\mathrm{grx}} R_2^{\mathrm{gtx}}}{R_1^{\mathrm{gtx}} R_2^{\mathrm{grx}}} \end{bmatrix}}_{(c)}. \end{aligned} \tag{5.16}$$

In (5.16), part (a) is a complex scale, which changes the signal strength for both LoRa devices. Part (b) is a diagonal matrix, which changes the signal strength difference between two devices. Therefore, both parts (a) and (b) do not change the signal beamforming directions and thus can be ignored in the beamforming process. Now, the question is how to find unknowns in part (c). To address this question, we propose an RF calibration scheme for the gateway, as shown in Fig. 5.12(b). The gateway first uses its first antenna to send a signal to its second antenna; denote the observed channel as $h_{1\to 2}$. Then, it uses its second antenna to send a signal to its first antenna; denote the observed channel as $h_{2\to 1}$. Then, we have $\frac{h_{2\to 1}}{h_{1\to 2}} = \frac{R_1^{\mathrm{grx}} R^{\mathrm{ota}} R_2^{\mathrm{gtx}}}{R_2^{\mathrm{grx}} R^{\mathrm{ota}} R_1^{\mathrm{gtx}}} = \frac{R_1^{\mathrm{grx}} R_2^{\mathrm{gtx}}}{R_1^{\mathrm{gtx}} R_2^{\mathrm{grx}}}$. Therefore, we have

$$\widetilde{\mathbf{H}}_{\mathrm{dl}} \triangleq \mathbf{H}_{\mathrm{ul}}^{\top} \begin{bmatrix} 1 & 0 \\ 0 & \frac{R_1^{\mathrm{grx}} R_2^{\mathrm{gtx}}}{R_1^{\mathrm{gtx}} R_2^{\mathrm{grx}}} \end{bmatrix} = \mathbf{H}_{\mathrm{ul}}^{\top} \begin{bmatrix} 1 & 0 \\ 0 & \frac{h_{2\to 1}}{h_{1\to 2}} \end{bmatrix}, \tag{5.17}$$

where $\triangleq$ indicates the ignorance of parts (a) and (b) in (5.16).

Eq. (5.17) shows our approach for gateway's RF calibration. It can be easily extended to a generic case. We have two remarks based on our experiments. First, the RF calibration can be done by the gateway in a standalone mode. Second, $\frac{R_1^{\mathrm{grx}} R_2^{\mathrm{gtx}}}{R_1^{\mathrm{gtx}} R_2^{\mathrm{grx}}}$ is stable over time and only need to calibrate at a low frequency.

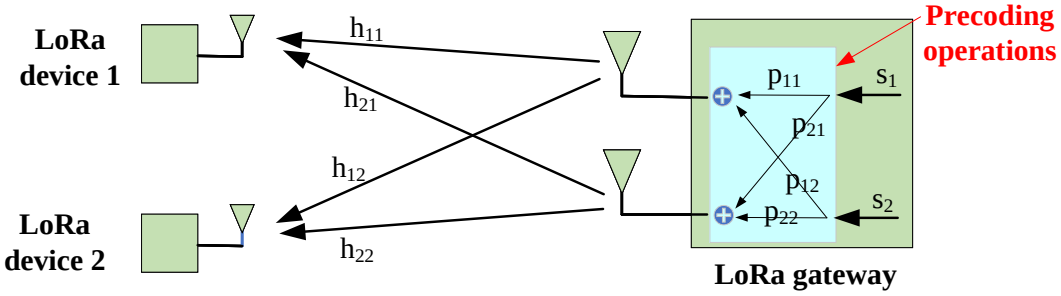
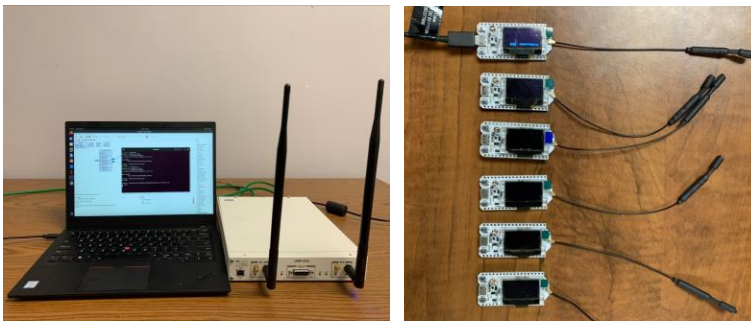


Figure 5.13: Downlink beamforming operation for downlink MU-MIMO transmission.



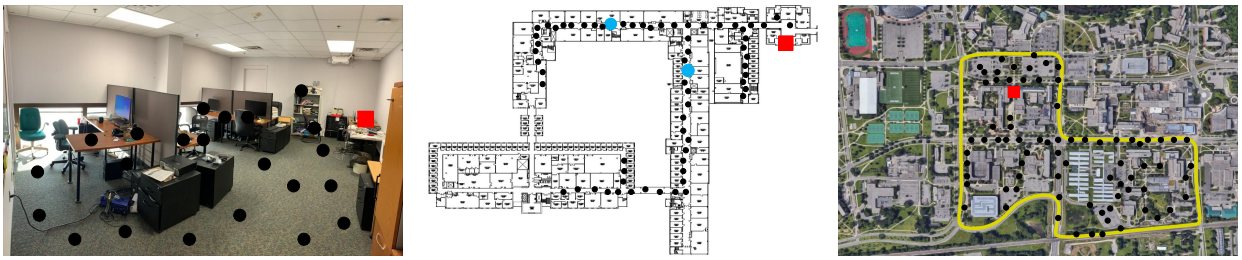
(a) (b)

Figure 5.14: (a) LoRaWAN gateway; (b) LoRa user devices.

Precoding for Downlink MU-MIMO Transmission. The purpose of precoding is to separate the signal streams in the over-the-air channel so that each LoRa device only receives its intended signal. Fig. 5.13 illustrates the precoding operation at the gateway in a toy-sized network. Recall that $\mathbf{H}_{\text{dl}} \in \mathcal{C}^{K' \times M}$ is the real downlink channel and $\widetilde{\mathbf{H}}_{\text{dl}} \in \mathcal{C}^{K' \times M}$ is the inferred downlink channel. Then, the zero-forcing precoder can be computed as follows:

$$\mathbf{P} = (\widetilde{\mathbf{H}}_{\text{dl}}^{\mathsf{H}} \widetilde{\mathbf{H}}_{\text{dl}})^{-1} \widetilde{\mathbf{H}}_{\text{dl}}^{\mathsf{H}}. \tag{5.18}$$

Denote $\mathbf{s} = [s_1, s_2, \ldots, s_{K'}]^\top$ as the baseband signals of data packets that the gateway wants to deliver to the K' LoRa devices. Then, the precoding operation can be expressed as $\mathbf{Ps}$. It can be verified that $\mathbf{H}_{\text{dl}}\mathbf{P}$ is a diagonal matrix. This means that $\mathbf{P}$ can pre-cancel inter-user interference in downlink MU-MIMO transmission.



(a) Lab (b) Building (c) Campus

Figure 5.15: Experimental evaluation scenarios.

5.6 Experimental Evaluation

In this section, we build a prototype of two-antenna MaLoRaGW and evaluate its performance with off-the-shelf LoRa devices in realistic wireless environments.

5.6.1 Implementation

Two-Antenna MaLoRaGW. We implement MaLoRaGW on a USRP X310, which has two antennas for transmission and reception. Fig. 5.14(a) shows the hardware for MaLoRaGW. We implemented the proposed PHY-layer design and MAC-layer design (see Fig. 5.6) on a laptop in C++ using GNU Radio OOT modules. The transmit power of USRP X310's each RF channel is set to 16 dBm, and the carrier frequency is set to 900 MHz. The spreading factor is set to 10. Three channel bandwidths, 125 kHz, 250 kHz, and 500 kHz, are used to study their impacts on uplink detection and downlink beamforming. The sampling rate is 1 MSps.

LoRa Devices. We use HELTEC ESP32 wireless modules (developed upon LoRa SX1276 chipset) shown in Fig. 5.14(b) as LoRa user devices. HELTEC ESP32 has an omni-directional antenna with 3 dBi gain. The receiver sensitivity of the board is -140 dBm, and its maximum transmit power is 20 dBm. We use open-source Arduino software to drive the modules and set the communication parameters. Particularly, we configure the modules to continuously transmit their

uplink packets. The modules send their packets independently without timing synchronization. We assign a unique word to each module. The word has 30 symbols, which are modulated by CSS with 10 SF. In downlink, the gateway concurrently transmits two independent packet streams to two selected LoRa modules. Each LoRa module measures its RSSI, counts the number of packets that are matched with its unique word, calculates the PER, and prints out the results on screen as shown in Fig. 5.14(b).

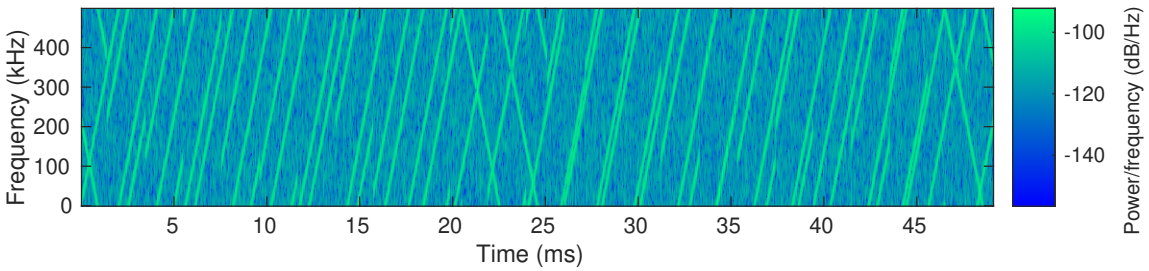
5.6.2 Experimental Setup, Metrics, and Baselines

We evaluated the performance of MaLoRaGW in three different scenarios: lab, office building, and university campus, as shown in Fig. 5.15. The gateway is placed at the spot marked by a red square in each scenario, while user devices are at some spots marked with solid dots. Particularly, Fig. 5.15(c) marks out the boundary of our test area using a yellow curve.

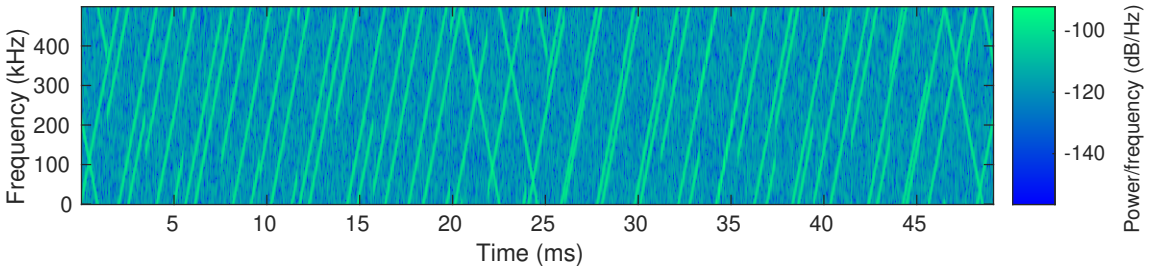
The following metrics will be used to evaluate the performance of MaLoRaGW: Received signal strength indicator (RSSI) in dBm, PER, and network throughput in kbps. Four existing LoRa gateway designs will be used as a comparison baseline to evaluate the performance of MaLoRaGW's uplink: conventional LoRa gateway, Choir [81], FTrack [75], and PCube [88]. We note that no prior work has considered concurrent downlink transmission. So we use a conventional LoRa gateway as the comparison baseline for MaLoRaGW's downlink transmission.

5.6.3 A Case Study

To understand how MaLoRaGW works, we consider a case study by placing MaLoRaGW and two LoRa devices in the building scenario shown in Fig. 5.15(b). The two LoRa devices are placed at the spots marked by blue circles. Only for this case, we use two USRP N210 devices as two



(a) The received signal on first gateway's antenna



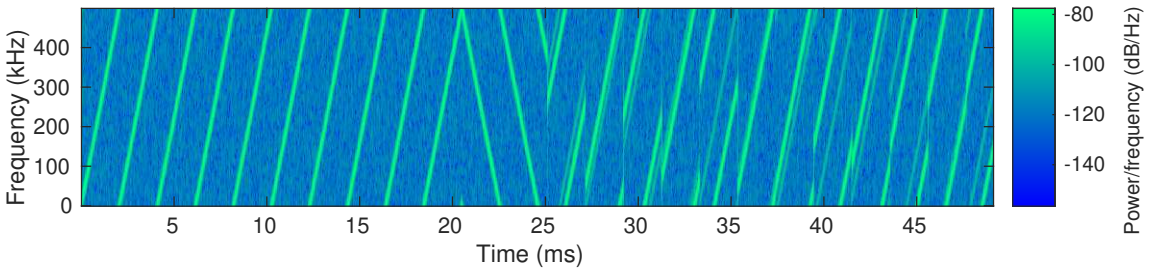
(b) The received signal on second gateway's antenna

Figure 5.16: Spectrogram of received signals at MaLoRaGW when two LoRa devices concurrently transmit their packets.

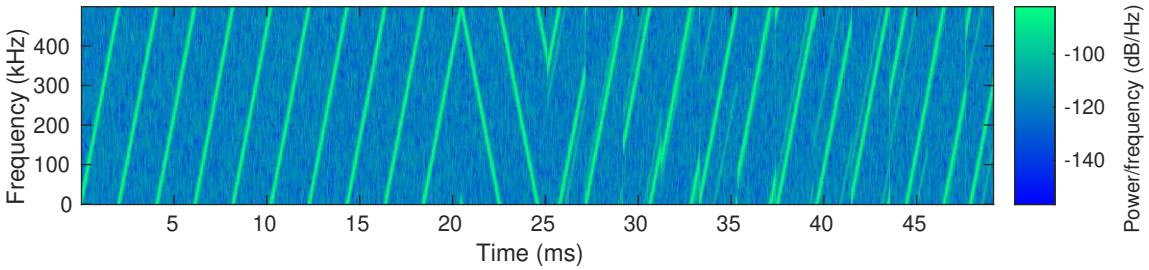
LoRa devices. This is because off-the-shelf LoRa devices are a closed system and do not provide detailed information. Using N210 allows us to examine the received signals and other PHY-layer parameters.

Uplink Performance. In the uplink, the two LoRa devices concurrently transmit their packets to MaLoRaGW. Fig. 5.16 plots the spectrogram of the collided signals at MaLoRaGW's two antennas. It can be clearly seen that two packet collision occurs. MaLoRaGW performs PCA-based synchronization, channel estimation, and packet decoding. The estimated CFO values are 2734 Hz and -3284 Hz for two LoRa users. The measured CTO values after preamble detection and frame extractions are 5.2 μs and 2.4 μs for the received signals from two devices. MaLoRaGW successfully decodes the two collided packets. The measured PER is zero for both users, and the uplink throughput is 4.9 kbps.

Downlink Performance. After decoding the uplink packets, MaLoRaGW takes advantage



(a) The received signal by LoRa device #1



(b) The received signal by LoRa device #2

Figure 5.17: Spectrogram of received signals at two LoRa user devices when concurrently served by MaLoRaGW in downlink.

of the channels estimated in uplink to perform beamforming for downlink transmission. Two independent packet streams are sent to those two LoRa devices (USRP N210 in this case). Fig. 5.17 plots the received signals by the two LoRa devices in downlink. While the desired signal pattern is very clear in the figure, a trace of inter-user interference can be seen alongside the desired signal at both devices. The interference can be attributed to the channel estimation error, calibration error, and other circuit imperfections. As it can be seen in the figure, the interference is much weaker than the desired signal.

Accuracy of Estimated Channels. Since the channel information is critical for downlink beamforming, it would be interesting to quantify the accuracy of the channel coefficients estimated in uplink. Unfortunately, this is impossible because the ground truth of channel coefficients is unknown. To circumvent this challenge, we measure the stability of the measured channel coefficients over a short period of time. Fig. 5.19 presents our experimental results. It shows

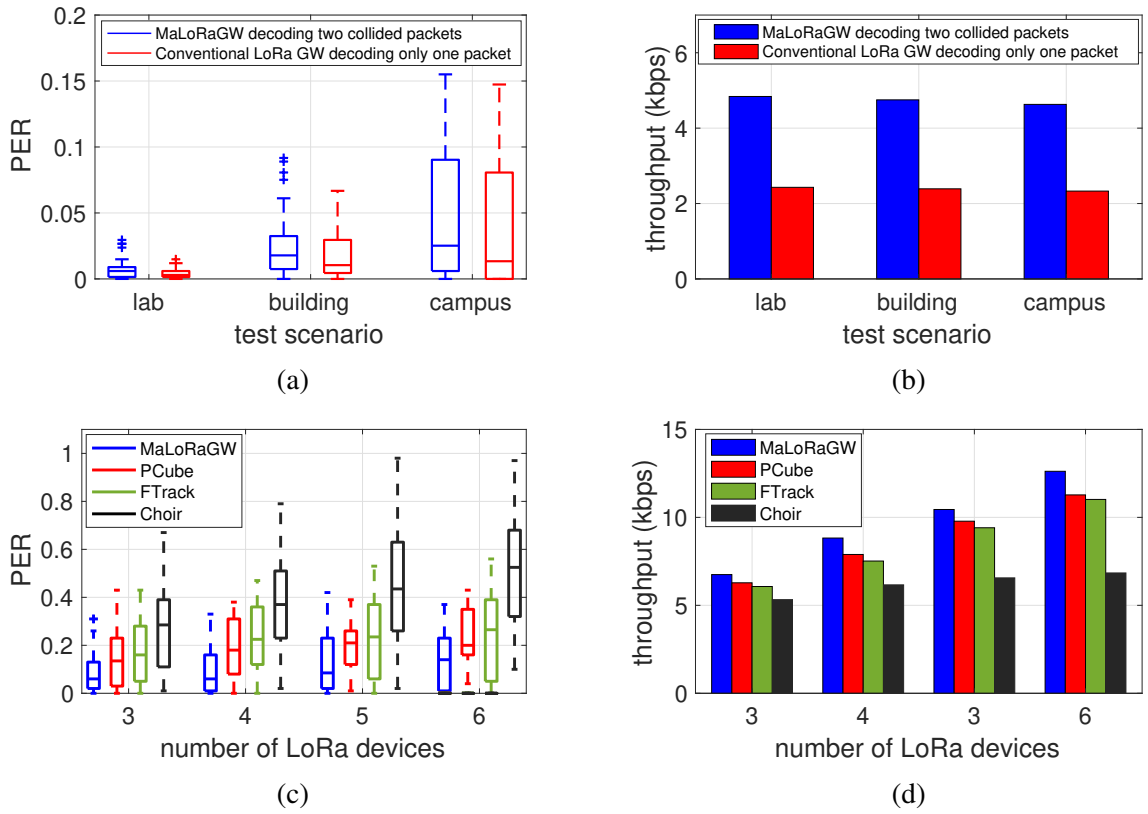
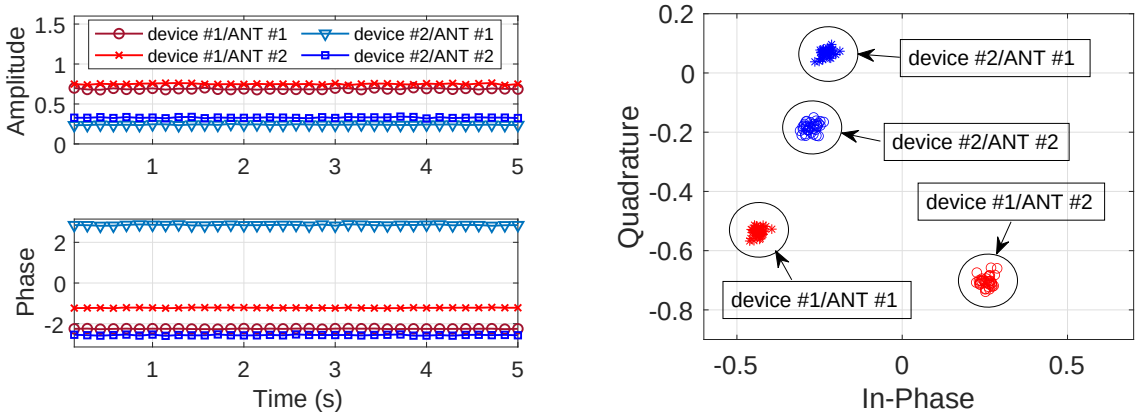


Figure 5.18: Uplink performance: (a) The PER performance when MaLoRaGW decodes two collided packets and a conventional LoRa gateway decodes a collision-free packet; (b) The throughput comparison between MaLoRaGW and a conventional LoRa gateway; (c) The PER performance comparison between MaLoRaGW and prior work; (d) The throughput comparison between MaLoRaGW and prior work.

the four channel coefficients measured from 40 consecutive packets (spanning over 5 seconds) in both temporal and I/Q domains. We can see that the channel coefficients remain stable over time. This indirectly shows the accuracy of our channel estimation method. Another fact that shows the channel estimation accuracy is the successful packet decoding at two LoRa devices. If the estimated channel coefficients were not accurate, the two LoRa devices would not be capable of decoding their own packets.



(a) (b)

Figure 5.19: (a) The amplitude and phase of four channel coefficients estimated at MaLoRaGW over 40 consecutive packets; (b) The I/Q scatters of the four estimated channel coefficients.

5.6.4 Uplink Performance

MaLoRaGW vs. Single-User LoRa Gateway. We compare the following two cases to quantify the performance of MaLoRaGW in uplink: i) MaLoRaGW is equipped with two antennas, and it decodes two collided packets from two uncoordinated COTS LoRa devices placed in all spots in Fig. 5.15. ii) A conventional LoRa gateway is equipped with one antenna. It decodes packets from only one active COTS LoRa device placed over all spots in Fig. 5.15. There is no collision in this case.

Fig. 5.18(a) presents the measured PER of MaLoRaGW and that of the single-user LoRa gateway. The mean measured PER at MaLoRaGW is 0.7% in lab, 2.6% in building, and 5.1% on campus. The mean measured PER at the single-user LoRa gateway is 0.5% in lab, 1.9% in building, and 4.4% on campus. The performance difference between MaLoRaGW and single-user LoRa gateway is 0.3% in lab, 0.7% in building, and 0.7% on campus. The experimental results indicate that a two-antenna LoRa gateway can double the number of serving users compared to a single-antenna LoRa gateway.

Fig. 5.18(b) plots the total uplink throughput achieved by MaLoRaGW and the single-user

LoRa gateway. The average uplink throughput achieved by MaLoRaGW is 4.9 kbps in lab, 4.8 kbps in building, and 4.6 kbps on campus. In contrast, the average uplink throughput achieved by the conventional single-user gateway is 2.4 kbps in lab, 2.4 kbps in building, and 2.3 kbps on campus. The results reveal that MaLoRaGW almost doubles the uplink throughput compared to a one-antenna LoRa gateway working in non-collision mode.

MaLoRaGW vs. Existing Schemes. Above we studied the performance of MaLoRaGW when it decodes two collided packets. We increase the number of packets in collision to see how MaLoRaGW performs. To do so, we place K LoRa dongles (see Fig. 5.14(b)) at K different spots in Fig. 5.15, where $3 \leq K \leq 6$. The K LoRa dongles are configured to transmit their unique words (packets) to the gateway simultaneously and independently. We repeat the above measurement many times to cover all those marked spots in Fig. 5.15. We compare MaLoRaGW against Choir [81], FTrack [75] and PCube [88]. Choir and FTrack have one antenna on their gateways, while MaLoRaGW and PCube have two antennas on their gateways.

Fig. 5.18(c) shows the measured PER of MaLoRaGW, PCube, FTrack, and Choir when the K LoRa dongles simultaneously send their packets to the gateway in the campus scenario. For MaLoRaGW, its average PER is 7.9% when $K = 3$, 9.6% when $K = 4$, 13.8% when $K = 5$, and 14.4% when $K = 6$. For PCube, its average PER is 14.3% when $K = 3$, 19.0% when $K = 4$, 19.8% when $K = 5$, and 23.1% when $K = 6$. For FTrack, its average PER is 17.1% when $K = 3$, 22.9% when $K = 4$, 23.0% when $K = 5$, and 24.8% when $K = 6$. For Choir, its average PER is 27.3% when $K = 3$, 36.9% when $K = 4$, 46.3% when $K = 5$, and 53.5% when $K = 6$. On average, MaLoRaGW reduces the average PER by 40.5% compared to PCube, 48.4% compared to FTrack, and 72.0% compared to Choir.

Fig. 5.18(d) plots the average uplink throughput achieved by MaLoRaGW, PCube, FTrack, and Choir when K ranges from 3 to 6. Specifically, the average uplink throughput achieved by

MaLoRaGW is 6.7 kbps when $K = 3$, 8.8 kbps when $K = 4$, 10.4 kbps when $K = 5$, and 12.6 kbps when $K = 6$. The average uplink throughput achieved by PCube is 6.2 kbps when $K = 3$, 7.8 kbps when $K = 4$, 9.7 kbps when $K = 5$, and 11.2 kbps when $K = 6$. The average uplink throughput achieved by FTrack is 6.1 kbps when $K = 3$, 7.5 kbps when $K = 4$, 9.4 kbps when $K = 5$, and 11.0 kbps when $K = 6$. The average uplink throughput achieved by Choir is 5.3 kbps when $K = 3$, 6.2 kbps when $K = 4$, 6.6 kbps when $K = 5$, and 6.8 kbps when $K = 6$. On average, MaLoRaGW has 10.8% throughput gain compared to PCube, 13.5% throughput gain compared to FTrack and 53.4% throughput gain compared to Choir. The throughput gain over Choir and FTrack is from the gateway's multiple antennas, which allow us to exploit the spatial feature (channel ratio) for resolving packet collision. We believe the throughput gain over PCube is mainly from the spatial signal projection operation, which reduces the near-far effect of collided packets and thus improves the packet decoding probability.

5.6.5 Downlink Performance

Prior work does not support concurrent downlink transmission in LoRaWANs. Actually, concurrent downlink transmission has not been studied in the literature. We therefore compare MaLoRaGW with a conventional LoRa gateway in downlink, where MaLoRaGW is equipped with two antennas and the conventional LoRa gateway with one antenna. We wish to see the performance gain that can be obtained by adding one more antenna on a LoRa gateway.

In the experiments, we first trigger two LoRa dongles for concurrent uplink transmission. MaLoRaGW estimates their channels and decodes their packets. It then performs beamforming for downlink MU-MIMO transmission. Surprisingly, the channels remain coherent for a pretty long time (e.g., 5 seconds without Tx, Rx, and surrounding movement) in most cases. In the following results, the overhead of gateway's RF calibration was not considered as it only needs to be done at

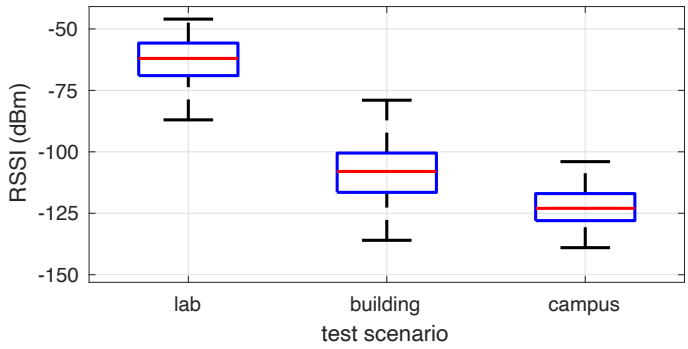


Figure 5.20: Measured RSSI at two LoRa dongles when they are concurrently served by MaLoRaGW in downlink.

a low frequency.

Measured RSSI. Fig. 5.20 shows the measured RSSIs displayed by the two LoRa dongles' screen (see Fig. 5.14(b)) in downlink MU-MIMO transmission when those two LoRa dongles were placed at the locations marked in Fig. 5.15. The average RSSI value is -62.9 dBm in lab scenario, -107.6 dBm in building scenario, and -122.9 dBm in campus scenario. Per LoRa dongle's manual, the chip's receiver sensitivity is -140 dBm. It means that the LoRa dongles should be able to decode most of their packets. This inference is consistent with our experimental observation.

PER. In this measurement, MaLoRaGW performs downlink MU-MIMO transmission with two LoRa dongles, while the conventional LoRa gateway performs downlink transmission with a single LoRa dongle. Both gateways send 1000 packets to their dongles, and the LoRa dongles report the number of packets being successfully decoded. We then calculate their PER and throughput. Fig. 5.21(a) plots the PER performance of LoRa dongles when served by MaLoRaGW and the conventional LoRa gateway in those three scenarios. When served by MaLoRaGW, the average PER of LoRa dongles is 1.0% in lab scenario, 2.6% in building scenario, and 5.4% in campus scenario. In contrast, when served by the conventional LoRa gateway, the average PER at the LoRa dongle is 0.4% in lab, 1.8% in building, and 4.7% in campus scenario. Numerically, the PER difference between MaLoRaGW and a conventional LoRa gateway is less than 1% (0.6% in

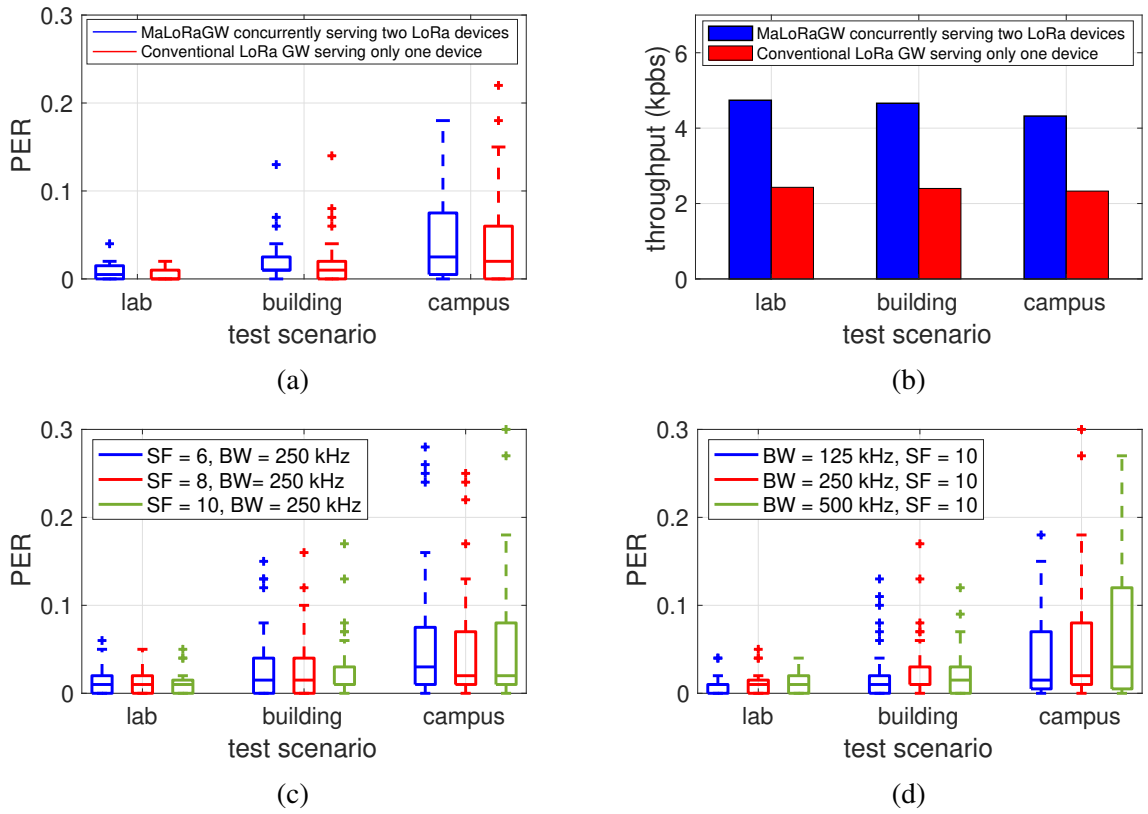


Figure 5.21: Downlink performance: (a) Measured PER at LoRa dongles when they are served by MaLoRaGW (using MU-MIMO) and a conventional LoRa gateway; (b) The throughput comparison between MaLoRaGW and a conventional LoRa gateway; (c) The PER performance of MaLoRaGW when different spreading factors (SF) are used; (d) The PER performance of MaLoRaGW when different bandwidth is used.

lab, 0.8% in building, and 0.7% in campus). This indicates that, from LoRa users' perspective, employing MU-MIMO at a gateway only slightly degrades their performance. This also indicates that employing MU-MIMO at a gateway is transparent to LoRa users.

Throughput. Fig. 5.21(b) shows the downlink throughput of MaLoRaGW and its conventional counterpart. The achievable network throughput in the downlink by the proposed MU-MIMO scheme The throughput achieved by MaLoRaGW is 4.7 kbps in lab, 4.6 kbps in building, and 4.5 kbps in campus. In contrast, the throughput achieved by a conventional LoRa gateway is 2.4 kbps in lab, 2.4 kbps in building, and 2.3 kbps in campus. The measurement reveals that a two-antenna MaLoRaGW almost doubles ($1.95\times$ actually) the downlink throughput compared to a

single-user LoRa gateway. This shows the potential of MU-MIMO in LoRa downlink transmission.

Impact of Spreading Factor (SF). We evaluate the PER performance of MaLoRaGW in downlink when using three different SF values, namely, 6, 8, and 10. Fig. 5.21(c) plots the PER measured at two LoRa dongles when served by MaLoRaGW in downlink MU-MIMO. When SF is set to 6, the average PER is 1.3% in lab, 3.2% in building, and 6.4% in campus. When SF is set to 8, The average PER is 1.2% in lab, 2.8% in building, and 6.2% in campus. When SF is set to 10, the average PER is 1.0% in lab, 2.6% in building, and 5.4% in campus. As expected, the PER at LoRa dongles decreases as the SF value increases. This is because a larger SF brings a higher spreading gain for packet detection.

Impact of Bandwidth. LoRa supports different bandwidth: 125 kHz, 250 kHz, and 500 kHz. We now conduct experiments to study the performance of MaLoRaGW with these three different bandwidths. We set SF to 10. Fig. 5.21(d) presents the measured PER at the two LoRa devices when served by MaLoRaGW in downlink. When bandwidth is 125 kHz, the average PER is 0.9% in lab, 2.3% in building, and 4.5% in campus. When bandwidth is 250 kHz, the average PER is 1.0% in lab, 2.6% in building, and 5.4% in campus. When bandwidth is 500 kHz, the average PER is 1.1% in lab, 2.8% in building, and 6.0% in campus. It can be seen that a smaller bandwidth offers a better PER performance. The reasons are twofold. First, a system with a smaller bandwidth will use more radio power to carry each of its bits, thereby reducing PER at its receiver. Second, a system with a smaller bandwidth tends to experience a frequency-flatter channel, which reduces the leakage of inter-user interference in downlink MU-MIMO transmission.

5.6.6 Limitations and Discussions

Antenna Limitation. While MaLoRaGW slightly improves the throughput of LoRa uplink transmission compared to the state-of-the-art, it significantly improves the throughput of LoRa downlink

transmission. However, our experiments of MaLoRaGW are limited to the two-antenna case. Theoretically, MaLoRaGW should work for a gateway with more (≥ 3) antennas, and it would provide higher throughput gain compared to single-antenna LoRa gateway. Validating the scalability of MaLoRaGW (over its antenna number) requires to substantiate efforts in system implementation, which will be carried out in our future work.

Hardware Cost and Energy Consumption. Compared to single-antenna LoRa gateways, MaLoRaGW has higher hardware cost and energy consumption. Fortunately, MaLoRaGW is backward compatible with commodity off-the-shelf LoRa devices and will not increase their hardware cost and energy consumption. Considering the fact that most LoRa gateways have sufficient power supplies, we believe the performance gain of MaLoRaGW outweighs the increase of its costs.

5.7 Chapter Summary

In this chapter, we presented MaLoRaGW, the first-of-its-kind LoRa gateway that enables MU-MIMO transmission in LoRa networks. MaLoRaGW features a joint design for uplink packet detection and downlink beamforming, enabling it to concurrently serve multiple LoRa user devices in both uplink and downlink. The key component of MaLoRaGW is a joint baseband signal design for uplink packet detection and downlink beamforming, which are underpinned by three modules: spatial signal projection, accurate channel estimation, and implicit beamforming. We have evaluated a two-antenna MaLoRaGW in realistic scenarios of different scales. It has been validated that MaLoRaGW is backward compatible with COTS LoRa devices. It further demonstrates 10% throughput gain in uplink and 95% throughput gain in downlink when compared to the state-of-the-art.

Chapter 6

Jamming-Resilient VANETs

Current data-driven intelligent transportation systems are mainly reliant on IEEE 802.11p to collect and exchange information. Despite promising performance of IEEE 802.11p in providing low-latency communications, it is still vulnerable to jamming attacks due to the lack of a PHY-layer countermeasure technique in practice. In this chapter, we propose JammingBird, a novel receiver design that tolerates strong constant jamming attacks. The enablers of JammingBird are two MIMO-based techniques: Jamming-resistant synchronizer and jamming suppressor. Collectively, these two new modules are able to detect, synchronize, and recover desired signals under jamming attacks, regardless of the PHY-layer technology employed by the jammers. We have implemented JammingBird on a vehicular testbed and conducted extensive experiments to evaluate its performance in three common vehicular scenarios: Parking lots (0$\sim$15 mph), local traffic areas (25$\sim$45 mph), and highways (60$\sim$70 mph). In our experiments, while the jamming attacks degrade the throughput of conventional 802.11p-based receivers by 86.7%, JammingBird maintains 83.0% of the throughput on average. Experimental results also show that JammingBird tolerates the jamming signals with 25 dB stronger power than the desired signals.

6.1 Introduction

The efficiency of transportation systems is not merely about building better highways anymore; it is about intelligence. An intelligent transportation system (ITS) is a data-driven infrastructure

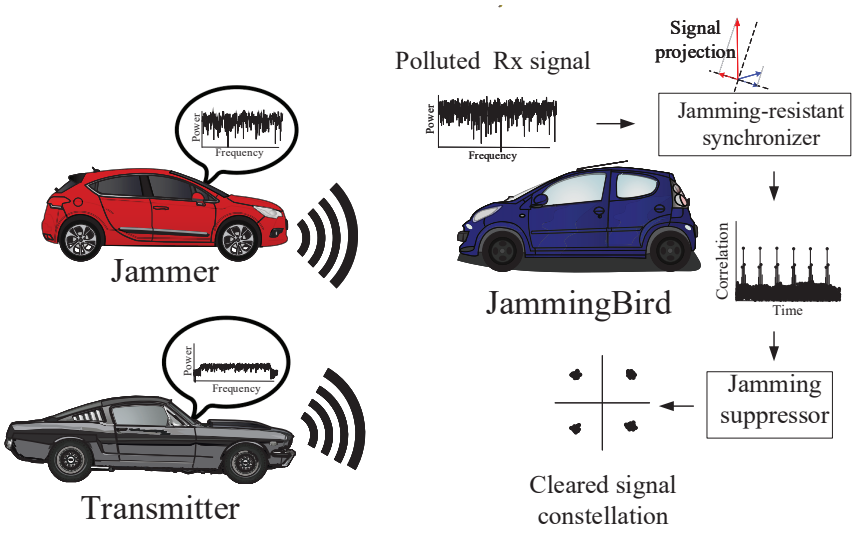


Figure 6.1: JammingBird recovers packets buried in strong jamming signals with the aid of two modules. Jamming-resistant synchronizer identifies and synchronizes legitimate jammed packets. Jamming suppressor removes the jamming signal and recovers the desired packets.

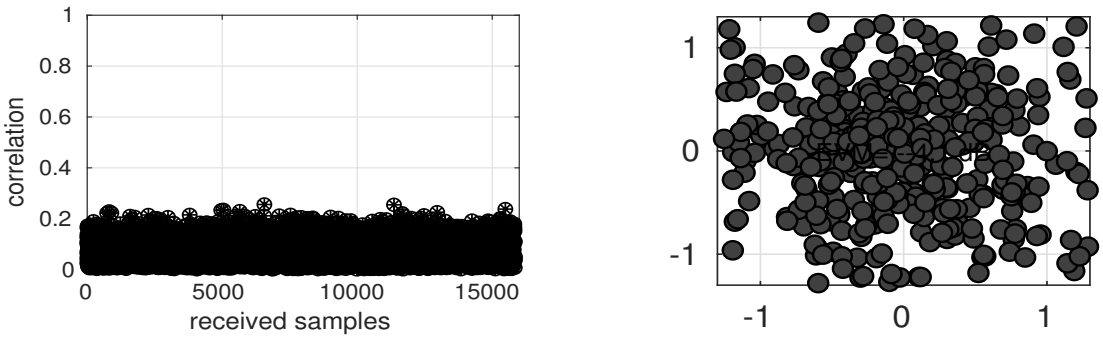
that significantly contributes in improving public safety [105], economy [106], environmental ecosystems [107] of developed societies. To realize such a data-centric ITS, vehicular ad hoc networks (VANETs) are the primary mean of collecting data. VANETs offer efficient vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications for safety and non-safety data exchange [108]. IEEE 802.11p is the pervasive technology that provides low-latency wireless communications for VANETs. IEEE 802.11p amends IEEE 802.11 standard to meet the requirements of ITS applications in 5.9 GHz frequency band. Compared to legacy Wi-Fi, it uses 10 MHz bandwidth for better mobility management and enjoys higher maximum transmit power [109]. It can also establish communications with out-of-network users with wildcard BSSID for broadcasting time-intensive data, such as crashes and traffic congestion messages.

Despite all the amendments to meet the timing and throughput needs of ITS applications, IEEE 802.11p has remained almost defenseless for a decade when it confronts jamming attacks. In fact, even a simple constant jamming attack can be regarded as a big security threat for VANETs [10]. Causing denial of service at network users, such a jamming attack ages safety-related information

and finally makes it outdated [110]. Thus, it is of great importance to reinforcing VANETs against jamming attacks [111]. In response to this urgent need, we have proposed JammingBird, as shown in Fig. 6.1. JammingBird rectifies the vulnerabilities of IEEE 802.11p at the PHY layer and effectively subverts strong constant jamming attacks. JammingBird can recover desired signals which are drowned into powerful jamming signals, a task that cannot be accomplished by conventional 802.11p-based receivers.

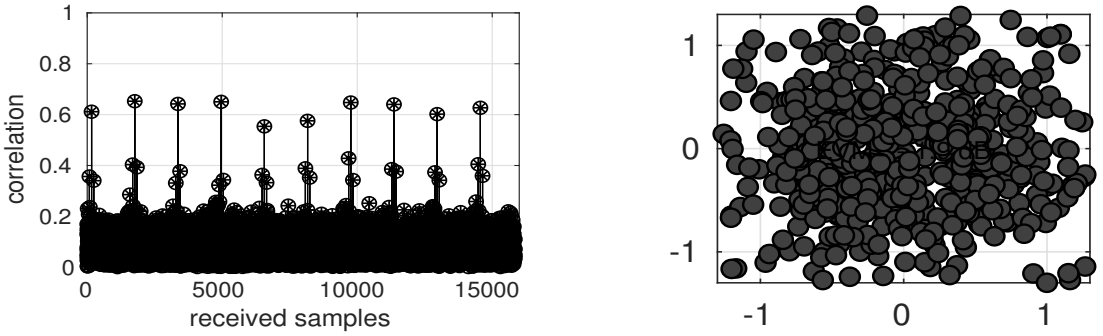
Vulnerability of IEEE 802.11p to Jamming Attacks. We have conducted a preliminary experiment on a conventional 802.11p-based receiver to show how vulnerable it is when facing jamming attacks. We have implemented the legitimate transmitter, conventional receiver, and jammer using USRP devices and laptops. First, we have considered a harsh test environment where all users moved on a highway in the same direction at $60\sim70$ mph. The legitimate users were 150 ft apart, and the jammer was located in between. The jammer was sending a noise-like signal to interrupt legitimate transmissions. In our experiments, an average of 20 dB jamming to signal ratio (JSR) was observed at the receiver side. Fig. 6.2 shows the hardship of 802.11p-based receiver in decoding its desired signals. As shown in Fig. 6.2(a), it fails in coarse time synchronization. The receiver cannot notice the existence of desired packets. Even if the receiver is forced to proceed with the highest correlation peak (the most probable starting sample for a legitimate packet) over a long time window, the decoded signal will be erroneous as shown in Fig. 6.2(b). The signal is not recovered as it is highly polluted by the jamming signal.

We have repeated our experiments in a benign test environment, where all the nodes were static in an open parking lot. Jammer and the legitimate transmitter were located 100 ft away from the receiver. The observed JSR was about 0 dB. Fig. 6.3 show the performance of 802.11p-based receiver. Despite a marginal improvement in synchronization, the receiver was still unable to decode the desired signal. Clearly, the constant jamming attack completely brought down the



(a) Synchronization (b) Decoded signal

Figure 6.2: Performance of conventional 802.11p-based receiver at a highway when JSR=20 dB.



(a) Synchronization (b) Decoded signal

Figure 6.3: Performance of conventional 802.11p-based receiver at a parking lot when JSR=0 dB.

802.11p-based communication. One may think this issue can be easily treated by adjusting transmit power on the legitimate transmitter. However, high transmit power levels unnecessarily densify the networks, exacerbate undesirable events like broadcast storms, and beget large backoff windows across the legitimate users.

JammingBird. We propose JammingBird, a new wireless receiver design to recover desired data packets in the presence of constant jamming attacks. As shown in Fig. 6.1, JammingBird takes advantage of recent advances in MIMO technology to mitigate unknown jamming signals and decodes the data packets. It addresses the underlying challenges of 802.11p-based receivers in encountering jamming attacks: packet detection, synchronization, and data recovery. Specifically, JammingBird reinforces 802.11p receivers with two novel modules.

JammingBird approaches the packet detection and synchronization problems with *jamming-resistant synchronizer*. This module comprises spatial projection filters to alleviate jamming signals by destructively combining those signals over different antennas. The spatial projection averts the jamming effect, making JammingBird able to notice the existence of a legitimate packet and find its starting sample. The legitimate, yet polluted, packets will be passed to *jamming suppressor* module. This module leverages the IEEE 802.11p frame structure and offers a blind jamming mitigation technique. It does not need the CSI between the jammer and receiver. Instead, it uses the short training field (STF) in the legitimate frames to calculate the jamming CSI ratio and recovers the desired packets. Collectively, these two modules lay the foundation of JammingBird.

We have implemented JammingBird on a proof-of-the-concept vehicular wireless testbed and evaluated its performance on three common scenarios in VANETs: parking lots ($0\sim15$ mph), local traffic areas ($25\sim45$ mph), and highways ($60\sim70$ mph). For these cases, JammingBird respectively reaches 26.2 Mbps, 22.2 Mbps, and 19.5 Mbps. In our experiments overall cases, while the jammer degrades the throughput of regular 802.11p-based receivers by 86.7%, JammingBird maintains 83.0% of throughput when facing jamming signals. The experimental result proves the efficacy of JammingBird in mitigating unknown and strong constant jamming attacks.

6.2 Related Work

In our literature review, we focus on two trajectories. We first review jamming attacks and their countermeasures designed for VANETs. Then, we review MIMO-based anti-jamming techniques.

Jamming Attacks and Anti-Jamming Strategies in VANETs. The security threats in VANETs can be divided into three types of attacks: (i) attacks on vehicular systems, (ii) attacks on information, and (iii) attacks on infrastructure [112]. In the first type of attack, the attacker may target interrupting

the social engineering, malware integration, sensor impersonation, and bogus information. The second type attempts to attack the information circulating through the VANETs using jamming attacks, spoofing attacks, fake information, and false position attack. The third type considers attacking the back-end and the network, such as the bogus information between the roadside units and central entities. In [110, 113], Punal *et al.* evaluated the vulnerability of the V2V communications in the face of different class of jamming attacks, including reactive jamming attacks and periodic jamming attacks.

Very limited work has been done so far to countermeasure jamming attacks in VANETs. There are two basic approaches in the literature to do so. As the first approach, the users in a jammed area can use an alternative infrastructure (e.g., cellular networks) for their communications. The authors in [114, 115] proposed to use unmanned aerial vehicles (UAVs) as relays to reroute the users' data traffic to alternative roadside infrastructure when the serving one is out of service due to jamming attacks. As the second approach, detection mechanisms might be used to detect and/or localize the jammer within the network [116]. In [117–119], a series of different techniques were proposed for jamming detection in VANETs. In [120, 121], learning-based approaches were proposed to detect and localize the jamming attacks in VANETs.

MIMO-based Anti-jamming Techniques. In [36,37], Yan *et al.* proposed a jamming cancellation technique for 802.11-based communications against reactive jamming attacks using a MIMO receiver design. The proposed scheme requires the legitimate transmitter's channel knowledge and frame structure modification to insert user-defined pilot signals for jamming mitigation purposes. In [35], Zeng *et al.* proposed a MMSE-based jamming mitigation solution for 802.11-based communications against constant jamming attacks. In [16], Pirayesh *et al.* proposed a MIMO-based jamming-resilient receiver to secure ZigBee communications against constant jamming attacks. The authors used an online learning approach to decode the ZigBee packets in the face of

the jamming signal. In particular, they designed a light-weight neural network and used the received ZigBee preamble signal within the packet for training the network. In [122, 123], jamming-resistant schemes were devised to secure massive MIMO uplink communications against constant jamming attacks. The schemes leverage the jamming CSI to design a jamming cancellation receiver in the spatial domain. However, a prior knowledge of the received jamming signal power on all antennas was required to estimate the jammer CSI.

JammingBird differs from the aforementioned MIMO-based and VANET-oriented countermeasures in the following aspects: First, JammingBird does not require any prior knowledge about the jammer, including its waveform, maximum transmit power, and signaling technology. Second, JammingBird does not require any modification in the standard framework of 802.11p-based transceivers in terms of extra signaling overhead for jamming mitigation purposes. Instead, it uses the 802.11p PHY-layer protocol and leverage MIMO technology to suppress the jamming signal and recover data.

6.3 System Model

We suit JammingBird for a V2X communication link between a single-antenna transmitter and a two-antenna receiver, as shown in Fig. 6.1. The transmitter and receiver can be either an onboard unit on a vehicle or a roadside unit connected to the backbone infrastructure. Although it is a miniature networking scenario, it is the most common V2X case, given the simplicity of both onboard and roadside units in typical VANETs. The V2X communications are established using OFDM modulation specified in IEEE 802.11p PHY. Let us assume that $\mathrm{X}[l,k] \in \mathcal{CN}(0,1)$ is the legitimate message transmitted over the lth OFDM symbol and the kth subcarrier. A single-antenna jammer disrupts the V2X communications by constantly sending powerful arbitrary signals. Let

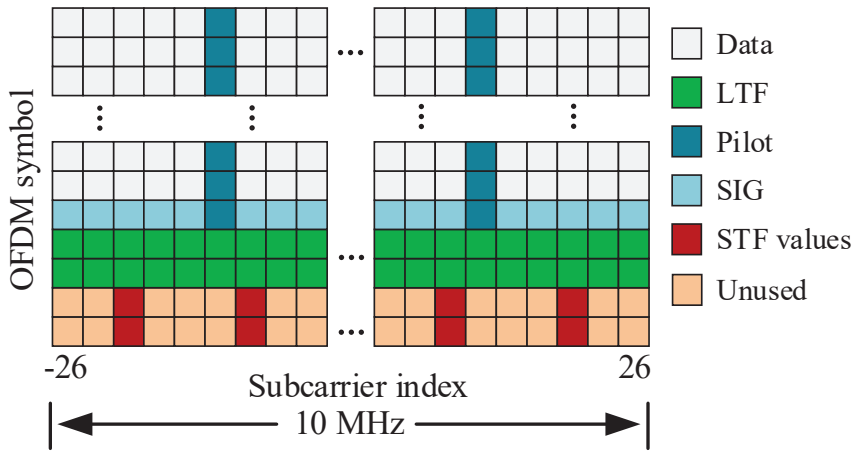


Figure 6.4: IEEE 802.11p frame format for V2X communications.

us consider that the legitimate message $\mathrm{X}[l,k]$ is obscured with $\mathrm{X}_j[l,k] \in \mathbb{C}$ from the jammer. Please note that it does not mean the jammer uses OFDM modulation. Instead, $\mathrm{X}_j[l,k]$ translates the effect of jamming signal in frequency domain.

We assume block fading channel within a packet. This is a mild assumption in VANETs, given the very short length of packets. We denote the channel gain between the ith antenna of receiver and the legitimate transmitter by $\mathrm{H}_i[k] \in \mathbb{C}$ over kth subcarrier. We also denote channel gain between ith antenna of receiver and the jammer by $\mathrm{H}_{j,i}[k] \in \mathbb{C}$ over kth subcarrier. Therefore, on the lth OFDM symbol and the kth subcarrier, the received signal by legitimate receiver can be expressed as:

$$\mathbf{Y}[l,k] = \mathbf{H}[k]\mathrm{X}[l,k] + \mathbf{H}_j[k]\mathrm{X}_j[l,k] + \mathbf{Z}[l,k], \tag{6.1}$$

where $\mathbf{Y}[l,k] = [\mathrm{Y}_1[l,k], \mathrm{Y}_2[l,k]]^{\mathrm{T}} \in \mathbb{C}^{2\times 1}$ denotes the received signal on both antennas. $\mathbf{H}[k] \triangleq [\mathrm{H}_1[k],\ \mathrm{H}_2[k]]^{\mathrm{T}} \in \mathbb{C}^{2\times 1}$ is the compound channel between the receiver and legitimate transmitter, and $\mathbf{H}_j[k] \triangleq [\mathrm{H}_{j,1}[k],\ \mathrm{H}_{j,2}[k]]^{\mathrm{T}} \in \mathbb{C}^{2\times 1}$ is the compound channel between receiver and the jammer. $\mathbf{Z}[l,k]$ stands for AWGN noise.

The jammer can arbitrarily choose its signal type. For instance, jammer can leverage noise-like, LTE-like, or CDMA-like signals to interfere the legitimate transmissions over the entire bandwidth of interest. V2X communications, on the other hand, leverage IEEE 802.11p frame format. Fig. 6.4 depicts the frame format used by V2X communications over 10 MHz at 5.9 GHz band. The frame comprises 64 subcarriers, including 12 null subcarriers and 52 valid subcarriers to carry data and pilot signal samples. The frame consists of a preamble field, signal field (SIG), and data field. While preamble and signal fields are of fixed length, the length of data field depends on the payload size. The preamble field is mainly used for packet detection, time and frequency synchronizations, and channel estimation. It consists of two identical short training field (STF) OFDM symbols and two identical long training field (LTF) OFDM symbols. In the frequency domain, each STF symbol only uses 12 subcarriers, as shown in Fig. 6.4. LTF symbols use a sequence of $+1$ and -1 values and populate all 52 valid subcarriers. The SIG field carries the modulation and coding scheme index and also determines the length of the frame. The data is mapped into 48 subcarriers, and the remaining four subcarriers are assigned to pre-known pilot signals for phase offset correction at the receiver.

6.4 Problem Description

In order to study the vulnerability of the 802.11p-based receiver, we briefly describe the main steps followed by the conventional receiver for recovering its desired signals in a non-hostile environment. As shown in Fig. 6.5, 802.11p-based receiver resembles the legacy Wi-Fi's receiver.

When a signal is received and sampled at the receiver, it first determines the existence of legitimate packets within the stored signal. Upon confirmation, it leverages correlation-based synchronization modules for correcting time and frequency offsets. The valid and corrected portion

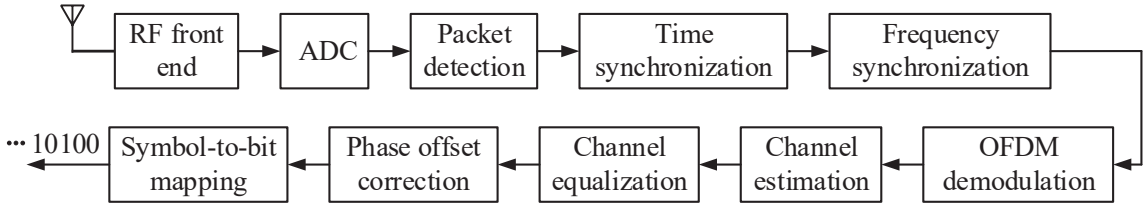


Figure 6.5: Conventional 802.11p-based receiver for V2X communications.

of the received signal is then converted into the frequency domain by the OFDM demodulation. The receiver employs LTF symbols in the preamble to estimate the channel at each subcarrier. The channel estimation takes place once for the entire frame. Then, the channel equalization subverts the effect of the estimated channel and recovers modulated symbols. Lastly, the phase offset is corrected using the pilot samples embedded into the frame as phase references. Now, the critical question is how a simple jamming attack can bring down the entire signal reception and recovery mechanisms.

6.4.1 Achilles heels of 802.11p-based receiver

Despite its subtle design, the 802.11p-based receiver experiences a hard time when it encounters jammers. In fact, the jammers impact multiple Achilles heels of the 802.11p-based receiver.

Physical Carrier Sensing. At both transmitter and receiver sides, the medium access of IEEE 802.11p is based on physical carrier sensing in part. At a time instance, the channel status is detected as occupied if a considerable energy level is detected over the spectrum. Under such a circumstance, the receiver actively looks for legitimate packets, and a potential transmitter postpones its transmission. As such, a jamming attack not only ages the information at the transmitters, it keeps the receivers unnecessarily active and inflicts a computational burden to them.

Packet Detection. The receiver uses auto-correlation of the received time-domain STF signal to detect the presence of a legitimate packet. Given the limited length of STF signal and possible

dominance of jamming signals, auto-correlation result could be too ambiguous and full of spurious spikes. Consequently, the false alarm rate in packet detection tends to be drastically high.

Synchronization.: Let us assume the receiver detects the presence of a legitimate packet by any means, such as a visible and sudden change in energy of received signal. Finding the start of the frame, which is buried in jamming signal, is a tedious task. Furthermore, the frequency synchronization is prone to large errors, and itself may cause additional frequency offset.

Channel Equalization. To avert the distortions from wireless medium, the channel gains should be estimated first. This is a challenging task under jamming attacks, as the packets' preambles are highly polluted by jamming signals. Also, the channel of jamming signal cannot be estimated in general, as the jammer can use any signals for jamming purposes.

These weak spots make the conventional 802.11p-based receiver vulnerable to jamming attacks. As shown by our preliminary experiments, the receiver may fail to decode signals when exposed to hostile environments.

6.4.2 Design Objectives and Challenges

JammingBird is designed as a treatment to Achilles heels of 802.11p-based receiver. JammingBird needs to address the shortcomings of the 802.11p-based receiver in packet detection, synchronization, and channel equalization.

First, JammingBird should be able to detect the presence of legitimate packets which are likely drowned into strong and unknown jamming signals. Second, if the existence of legitimate signals is confirmed, it must detect start of packets at sample level and correct frequency offset. These tasks are cumbersome due to the dominance of jamming signal, finite length of preamble, and offset injection from the jammer. The limited length of the preamble prevents correlation-based synchronization to bare clear correlation spikes as expected in asymptomatic cases (even under

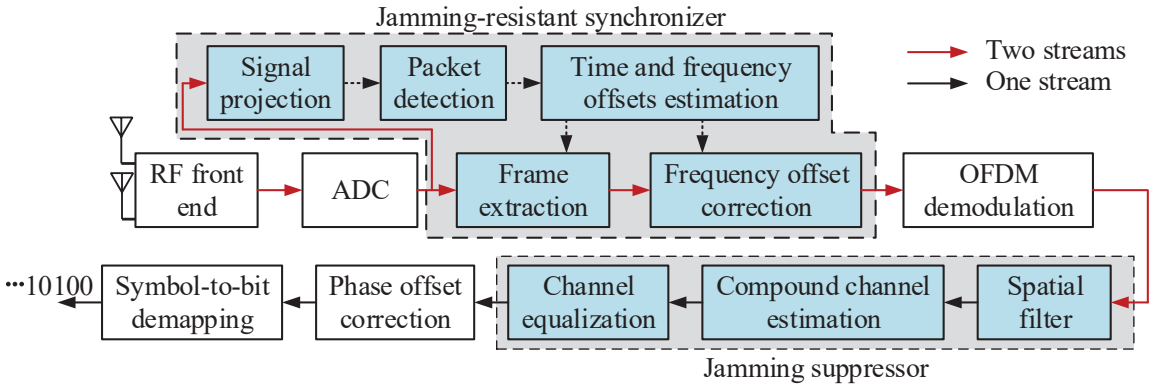


Figure 6.6: The structure of JammingBird with its new modules.

strong jamming attacks). Also, it is quite possible that frequency offset traces from jammer mislead the synchronization module in compensating the frequency offset between legitimate transmitter and the receiver.

When legitimate portion of the signal is detected, and offsets are compensated, JammingBird should suppress the jamming signal, equalize the channels, and recover the desired packets. This is another challenging task. It is not possible to acquire the channel gains between the jammer and receiver as the jammer is not bounded to any specific communication technology. Hence, its power level, frame format, and waveform are arbitrary. Also, due to the strong and uncontrolled power emitted by the jammer, the estimation of channels between legitimate users is very challenging, if not impossible.

6.5 JammingBird: A Jamming-Resilient Receiver

JammingBird bears high-power and unknown constant jamming attacks. It is blessed with two new modules as shown in Fig. 6.6, namely *jamming-resistant synchronizer* and *jamming suppressor*. Jamming-resistant synchronizer enables the receiver to detect the packets and successfully perform time and frequency synchronizations in the presence of a jamming attack. Once the start of a

legitimate packet is identified, and the offsets are compensated, the polluted signal will be translated into the frequency domain and passed to the jamming suppressor. The jamming suppressor first removes the effect of jamming signals from the received signal with the aid of a spatial filter. The cleared signal is then used for channel estimation and equalization. These two brand new modules effectively subvert the jamming signals, enabling JammingBird to survive under strong jamming attacks. In the following, each new module is presented in detail.

6.5.1 Jamming-Resistant Synchronizer

The jamming-resistant synchronizer identifies and extracts the portion of received time-domain signals containing legitimate IEEE 802.11p packets. Thereafter, the frequency offset between the legitimate transmitter-receiver pair is compensated. The synchronizer module leverages a spatial filter to determine the existence and beginning of packets in the presence of jamming attacks. The design of this filter is not contingent on channel knowledge and can be accomplished blindly. The filter alleviates the impact of jamming signals, allowing us to apply conventional correlation-based packet detection and synchronization techniques.

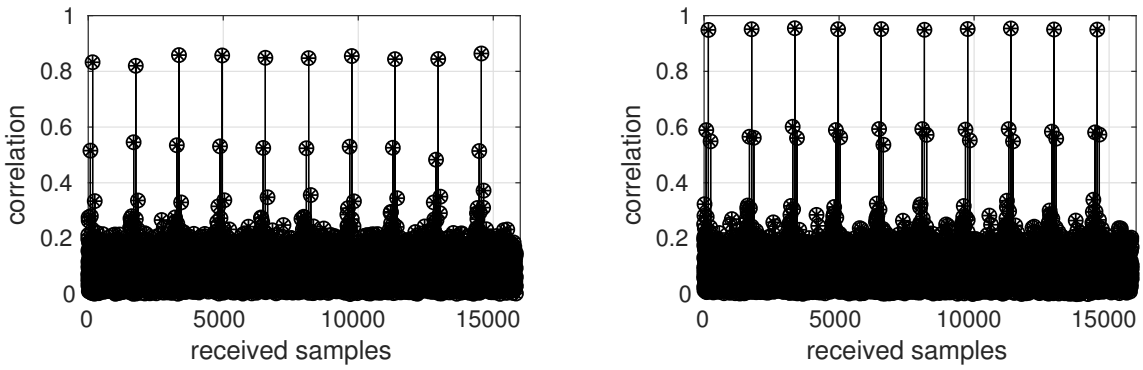
To compute the filter, we perform an eigenvalue decomposition (EVD) on the received signals from both antennas. Let us denote $\mathbf{y} \in \mathbb{C}^{2\times N_s}$ as the received time-domain signals on both receiving antennas and denote N_s as the number of collected samples on each antenna. Auto-correlation of the received samples is $\mathbf{R}_{\mathrm{yy}} = \mathbb{E}\{\mathbf{y}\mathbf{y}^{\mathrm{H}}\}$, where the symbols $[\cdot]^{\mathrm{H}}$ and $\mathbb{E}\{\cdot\}$ represent the conjugate transpose and expectation operators, respectively. EVD of the $\mathbf{R}_{\mathrm{yy}}$ can be expressed as $[\mathbf{Q}, \Lambda, \mathbf{Q}^{-1}] = \mathrm{EVD}(\mathbf{R}_{\mathrm{yy}})$, where Λ is the diagonal matrix comprising the eigenvalues of $\mathbf{R}_{\mathrm{yy}}$ on its diameter. The columns of $\mathbf{Q} \in \mathbb{C}^{2\times 2}$ are the eigenvectors of $\mathbf{R}_{\mathrm{yy}}$.

The signaling space can be completely spanned by columns of $\mathbf{Q}$ as its two bases. We decompose the signaling space into two complementary subspaces, each spanned by a column

vector in $\mathbf{Q}$. Assume that the desired signal subspace is spanned by $\mathbf{Q}_s \in \mathbb{C}^{2\times 1}$, and the jamming subspace is spanned by $\mathbf{Q}_j \in \mathbb{C}^{2\times 1}$, and $\mathbf{Q} = [\mathbf{Q}_s, \mathbf{Q}_j]$. Then, we apply the signal subspace basis as the projection filter over the received signal by letting $\mathbf{y}_p = \mathbf{Q}_s^{\mathsf{H}}\mathbf{y}$. To find the basis of desired signal subspace, we examine both available bases and look up the cross-correlation results in synchronization. If visible spikes are witnessed, the vector that achieves a higher correlation peak will be selected as the spatial projection filter. At the same time, the emergence of correlation spikes reveals the existence of legitimate packets. Once a legitimate packet is detected, the projected signal onto desired signal subspace will be subjected to conventional time synchronization. The extracted signal is used for computing the carrier frequency offset as $\theta = 1/64 \cdot \angle(\sum_{n=M}^{n=M+64} y_p[n] y_p[n+63]^{\mathsf{H}})$, where $\angle(\cdot)$ is the angle of a complex number, and M is the position of the first LTF sample, and $\mathbf{y}_p[n]$ is the nth column of $\mathbf{y}_p$. The frequency offset is then compensated by multiplying $e^{-j\theta n}$ to synchronized signal.

To illustrate the effectiveness of jamming-resistance synchronizer, we have repeated the experiment conducted in Section 6.1 under the same networking scenario and communication environment. This time, we have leveraged a jamming-resistant synchronizer. Fig. 6.7 shows the performance of our proposed synchronizer. Comparing Fig. 6.7(a) with Fig. 6.2(a), it is evident that the jamming-resistant synchronizer is able to successfully detect and synchronize legitimate packets in a very hostile environment, where the transmissions undergo strong jamming attacks on a highway. The conventional receiver, however, fails to do so. Comparing Fig. 6.7(b) with Fig. 6.3(a), it can bee seen that our proposed synchronizer outperforms conventional 802.11p-based receiver in finding and synchronizing packet at a parking lot where JSR equals to 0 dB.

Albeit successful in synchronization and packet detection, the proposed spatial filter is not capable of suppressing the jamming signal and clearing it for final signal recovery. As such, we use another module to mitigate the jamming signals.



(a) Synchronization result (b) Synchronization result

Figure 6.7: Jamming-resistant synchronizer at: (a) highway with JSR=20 dB and (b) parking lot with JSR=0 dB.

6.5.2 Jamming Suppressor

We describe two main steps that jamming suppressor takes toward recovering the desired signals in the following.

Jamming Signals Filtering. We design a spatial filter and apply it to the received signal in (6.1). The filter is able to remove the effect of jamming signal effectively. Let us denote the jamming suppressor filter as $\mathbf{U}[k] = [\mathrm{U}_1[k], \mathrm{U}_2[k]]^{\mathrm{T}} \in \mathbb{C}^{2\times 1}$. For such a filter, we have the following proposition.

Proposition 1: The jamming suppressor filter $\mathbf{U}[k] = [1,\ -\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]]^{\mathsf{H}}$ *completely removes the jamming signal* $\mathrm{X}_j[l,k]$ *from the received signal if noise is negligible.*

Proof: Upon applying the jamming suppressor filter on the received signal as $\mathrm{Y}_c[l,k] = \mathbf{U}^{\mathsf{H}}[k]\mathbf{Y}[l,k]$, the filtered signal on subcarrier k and OFDM symbol l can be expressed as:

$$\begin{aligned}\mathrm{Y}_c[l,k] =& \mathbf{U}^{\mathsf{H}}[k]\mathbf{H}[k]\mathrm{X}[l,k] + \mathbf{U}^{\mathsf{H}}[k]\mathbf{H}_j[k]\mathrm{X}_j[l,k] \\ &+ \mathbf{U}^{\mathsf{H}}[k]\mathbf{Z}[l,k].\end{aligned} \tag{6.2}$$

To clear the effect of jamming signal in (6.2), the term $\mathbf{U}^{\mathsf{H}}[k]\mathbf{H}_j[k]\mathrm{X}_j[l,k]$ needs to be nullified.

It is equivalent to letting $\mathrm{U}_1^{\mathsf{H}}[k]\mathrm{H}_{j,1}[k] + \mathrm{U}_2^{\mathsf{H}}[k]\mathrm{H}_{j,2}[k] = 0$. Such a condition will be easily met if $\mathrm{U}_1^{\mathsf{H}}[k] = 1$, and $\mathrm{U}_2^{\mathsf{H}}[k] = -\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$. This completes the proof and confirm the efficacy of the design presented in Proposition 1.

When the jamming suppressor filter is designed, it nullifies $\mathbf{U}^{\mathsf{H}}[k]\mathbf{H}_j[k]\mathrm{X}_j[l,k]$ on subcarrier k. Therefore, (6.2) can be represented as:

$$\mathrm{Y}_c[l,k] = \mathrm{H}_c[k]\mathrm{X}[l,k] + \mathrm{Z}_c[l,k], \tag{6.3}$$

where $\mathrm{H}_c[k] \triangleq \mathrm{H}_1[k] - \mathrm{H}_2[k].\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$ and $\mathrm{Z}_c[l,k] \triangleq \mathrm{Z}_1[l,k] - \mathrm{Z}_2[l,k].\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$. The desired signal in (6.3) can be recovered with equalizing $\mathrm{H}_c[k]$ over subcarrier k. It is evident that neither jamming suppression step nor signal recovery step is reliant on the exact values of $\mathrm{H}_{j,1}[k]$ and $\mathrm{H}_{j,2}[k]$. Jamming suppressor, instead, uses the CSI ratio of $\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$ for both eliminating the jamming signal and recovering the desired one. Due to the unknown PHY technology used by the jammer, it is not possible to compute the CSI ratio by dividing the individual CSI values at two antennas of the receiver. However, it is possible to directly calculate the CSI ratio using IEEE 802.11p frame format.

We take advantage of the unused time-frequency resources within the 802.11p frames to estimate the required jammer CSI ratio over each subcarrier. Denote $\gamma[l,k]$ as the jammer CSI ratio for the signal sample received on OFDM symbol l and subcarrier k. Then, we estimate $\gamma[l,k]$ as:

$$\gamma[l,k] = \frac{\mathrm{Y}_1[l,k]}{\mathrm{Y}_2[l,k]}, \quad \text{for } k \in \mathcal{K} \text{ and } l \in \mathcal{L}, \tag{6.4}$$

where $\mathcal{K}$ is the set of 40 valid subcarriers in IEEE 802.11p frames that are not assigned to the STF sequence. $\mathcal{L} = \{1, 2\}$ refers to the first two OFDM symbols shown in Fig. 6.4.

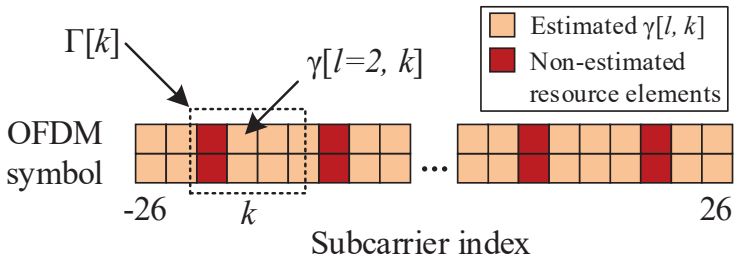


Figure 6.8: An illustration of the averaging filter applied for estimating $\tilde{\gamma}[k]$ on subcarrier k.

We face the following two challenges regarding the estimation of $\gamma[l,k]$ in (6.4).

Challenge 1. $\gamma[l,k]$ cannot be estimated over all the subcarriers using the received IEEE 802.11p frames. One may argue that $\gamma[l,k]$ can be calculated when there is no IEEE 802.11p frame inside the received signal. This is not possible in practice as the receiver would not carry out the necessary signal processing steps unless it detects the presence of a legitimate IEEE 802.11p frame.

Challenge 2. From (6.4), it is evident that $\gamma[l,k]$ is equal to $\mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$ only when $\mathrm{Z}_1[k,l]$ and $\mathrm{Z}_2[k,l]$ are zero. As this is not the case in real wireless environments, $\gamma[l,k]$ includes the effect of noise as the estimation error.

To overcome these challenges, we use an averaging filter to interpolate the values of $\gamma[l,k]$ for the missing subcarriers and reduce the impact of noise. Fig. 6.8 shows an instance of the averaging filter on subcarrier k. The averaging process on subcarrier k can be expressed as:

$$\tilde{\gamma}[k] = \frac{1}{|\Gamma[k]|} \sum_{k \in \Gamma[k]} \gamma[k,l], \tag{6.5}$$

where $\Gamma[k] = \{k - k_l \leq k \leq k + k_u \text{ and } l = 1,2\}$, in which k_l and k_u define the lower and upper bounds of the averaging window, respectively.

We resort to simulation in order to evaluate the performance of averaging filter in interpolating the missing subcarriers and canceling noise. Fig. 6.9 shows the amplitude and phase of the interpolated $\tilde{\gamma}[k]$ for all the subcarriers when the averaging filter in (6.5) is applied and noise is

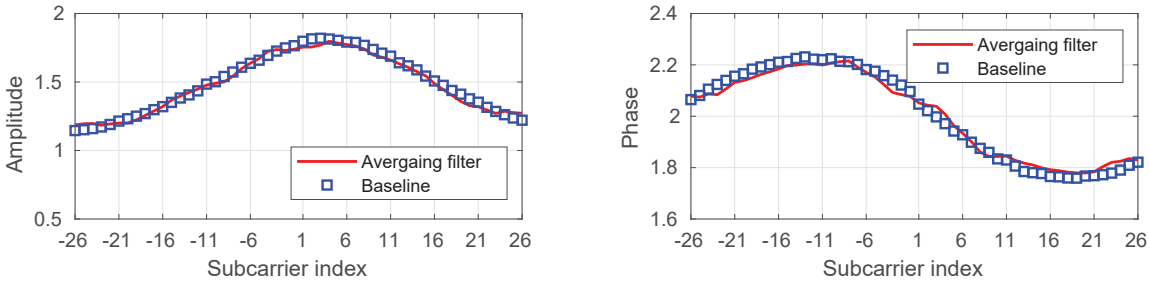


Figure 6.9: Amplitude and phase of interpolated $\tilde{\gamma}[k]$ v.s. ideal estimation of $\gamma[k]$ for all subcarriers.

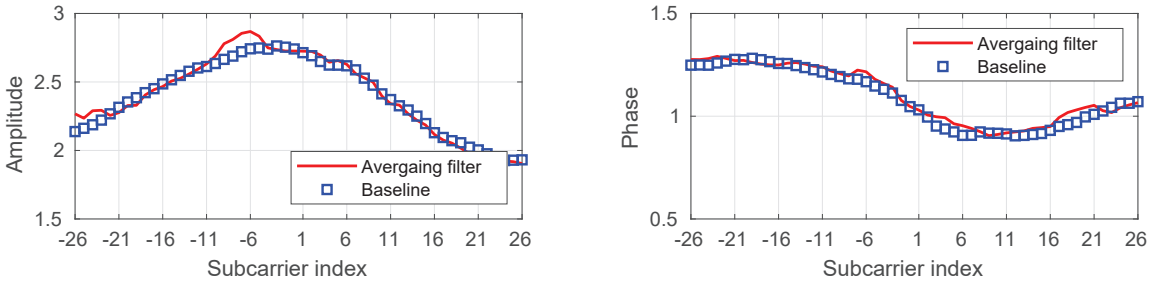


Figure 6.10: Amplitude and phase of interpolated $\tilde{\gamma}[k]$ when JNR is 20 dB v.s. ideal estimation of $\gamma[k]$ in noise-free scenario.

negligible. As the baseline, we use actual $\gamma[k]$ under the same channel realization. Fig. 6.10 shows the amplitude and phase of the estimated $\tilde{\gamma}[k]$ when the averaging filter is applied, and the jamming signal power to noise power ratio (JNR) is 20 dB. The baseline shows the results for the same setting in the noise-free scenario. The simulation results in Fig. 6.9 and Fig. 6.10 show that we can successfully interpolate the $\tilde{\gamma}[k]$ values for missing subcarriers and reduce the impact of the noise by using the filter presented in (6.5).

Desired signal recovery. Once the filter is applied to the received signal, the jamming suppressor follows its second task. Estimating the compound channel $\mathrm{H}_c[k]$, it recovers the desired signal $\mathrm{X}[l,k]$ over OFDM symbol l and subcarrier k. To do so, it leverages LTF symbols embedded into the preamble of IEEE 802.11p frames and uses the linear least-square method to estimate the compound channel. The estimated channel on subcarrier k can be expressed as $\hat{\mathrm{H}}_c[k] = (\mathrm{Y}_c[3,k] + \mathrm{Y}_c[4,k]) \,/\, (2.\mathrm{X}[3,k])$ for all k. Please note that the LTF OFDM symbols are

identical. As such, $\mathrm{X}[3,k] = \mathrm{X}[4,k]$.

We have assumed perfect estimation of $\gamma[k] = \mathrm{H}_{j,1}[k]/\mathrm{H}_{j,2}[k]$ so far. However, this is not a pragmatic assumption. To point out this issue, we replace the $\gamma[k]$ by $\gamma[k]+\varepsilon$, where ε denotes the estimation error. Then, the filtered signal in (6.3) can be rewritten as:

$$\mathrm{Y}_c[l,k] = \mathrm{H}_c^{'}[k]\mathrm{X}[l,k] + \varepsilon \mathrm{H}_{j,2}[k]\mathrm{X}_j[l,k] + \mathrm{Z}_c^{'}[l,k], \tag{6.6}$$

where $\mathrm{H}_c^{'}[k] \triangleq \mathrm{H}_1[k] - (\gamma[k]+\varepsilon)\mathrm{H}_2[k]$ and $\mathrm{Z}_c^{'}[l,k] \triangleq \mathrm{Z}_1[l,k] - (\gamma[k]+\varepsilon)\mathrm{Z}_2[l,k]$. The second term in (6.6) shows the jamming signal scales with ε and introduces an additional error in compound channel estimation. To reduce the impact of this undesired jamming signal as well as the additive noise in channel estimation process, we use channel smoothing technique. We bound $2 \sim 4$ subcarriers together and estimate the compound channel. Once the compound channel is estimated for all subcarriers, its effect is equalized and the signal is recovered as $\hat{X}[l,k] = \mathrm{Y}_c[l,k]/\mathrm{H}_c[k]$ for all k.

6.6 Experimental Evaluation

We have built JammingBird on a proof-of-the-concept vehicular testbed and evaluated its performance on real-world scenarios. We first describe the testbed and test scenarios in detail. Then, we present the performance metrics and experimental results.

6.6.1 Experimental Setting

Prototype. We have implemented JammingBird using a laptop and an X310 USRP device equipped with two antennas. For the legitimate transmitter, we use an N210 USRP device and a laptop. We

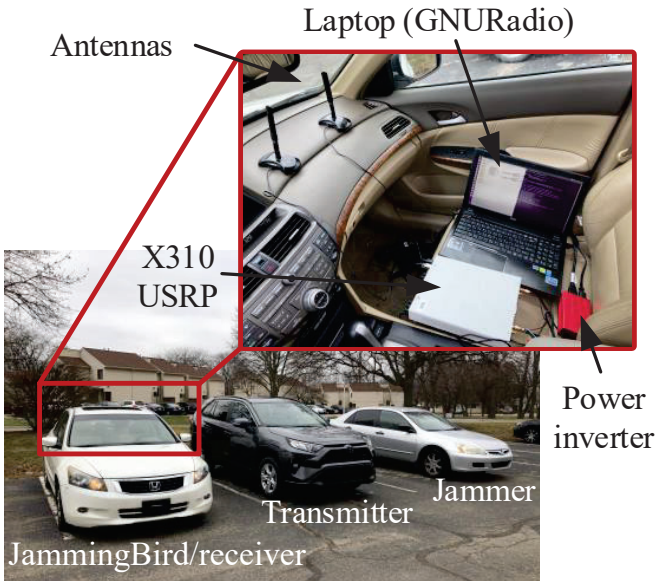


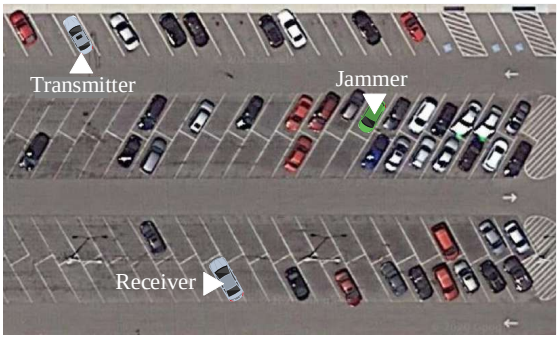
Figure 6.11: Our vehicular testbed for evaluating JammingBird.

use GNURadio protocol stack to drive the USRP devices and carry out the signal processing. The carrier frequency is set to 5.810 GHz, and the sampling rate is set to 10 MSps. The transmit power is also set to 13 dBm. Additionally, we have prototyped a jammer using an N210 USRP device connected to a laptop. The jammer constantly sends noise-like signals with a power of 20 dBm. Each of the legitimate users and the jammer has individually been mounted on a vehicle, as shown in Fig. 6.11.

Test Scenarios. Fig. 6.12 shows satellite pictures of our experimental environments: (i) parking lot scenario (Fig. 6.12(a)) where all three vehicles are mobile at speed of $0\sim15$ mph; (ii) local street scenario (Fig. 6.12(b)) where all vehicle move at speed of $25\sim45$ mph; and (iii) highway scenario (Fig. 6.12(c)) where the vehicles drive at relatively high speed of $60\sim70$ mph.

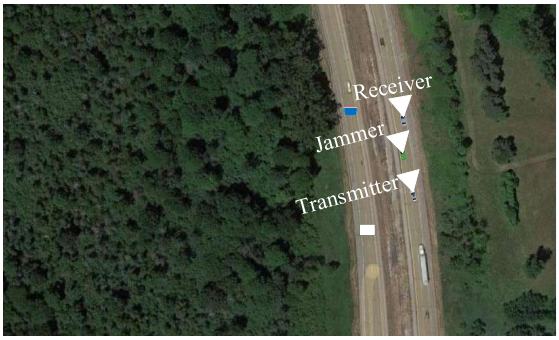
6.6.2 Performance Metrics and Baseline

Performance Metrics. We use EVM and throughput, as defined in Section 2.9.1 (equations (2.18) and (2.19)), to evaluate the performance of JammingBird and conventional 802.11p-based receivers.



(a) Parking lot scenario.

(b) Local street scenario.



(c) Highway scenario.

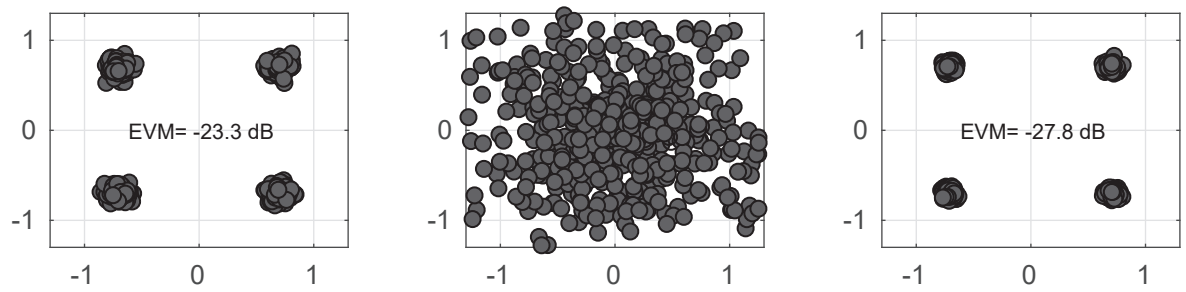
Figure 6.12: Three outdoor scenarios for evaluating the performance of JammingBird and conventional 802.11p-based receiver.

The achievable throughput in this case can be calculated by: $r = \frac{48}{80} \times 10 \times \gamma(\text{EVM})$ Mbps, where 48 is the number of payload subcarriers, 80 is the points of one OFDM symbol (including CP), 10 is the signal sampling rate (in Msps), and $\gamma(\text{EVM})$ is the average number of bits carried by one symbol and its possible values are given in Table 2.1.

Performance Baseline. As the baseline, we use the performance of a conventional 802.11p-based receiver in the jamming-free scenario, where the jammer is turned off.

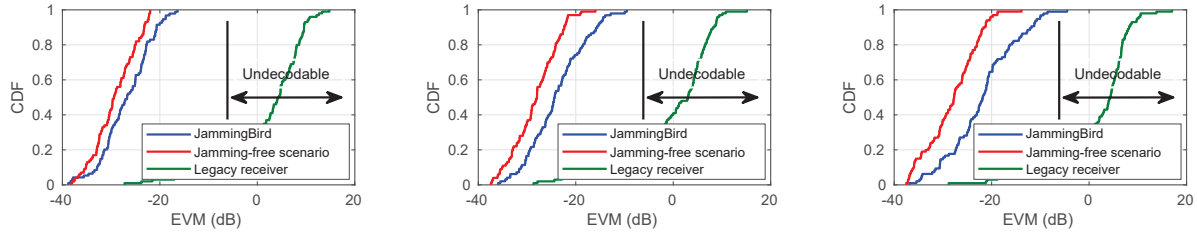
6.6.3 A Case Study

We conducted a case study to delineate the evaluation process of JammingBird in detail. We consider a local street scenario as shown in Fig. 6.12(b). JammingBird effectively recovers the



(a) (b) (c)

Figure 6.13: The constellation of the decoded signal by: (a) JammingBird under attack; (b) conventional received under attack; and (c) conventional receiver in jamming-free environment.



(a) Parking lot scenario. (b) Local street scenario. (c) Highway scenario.

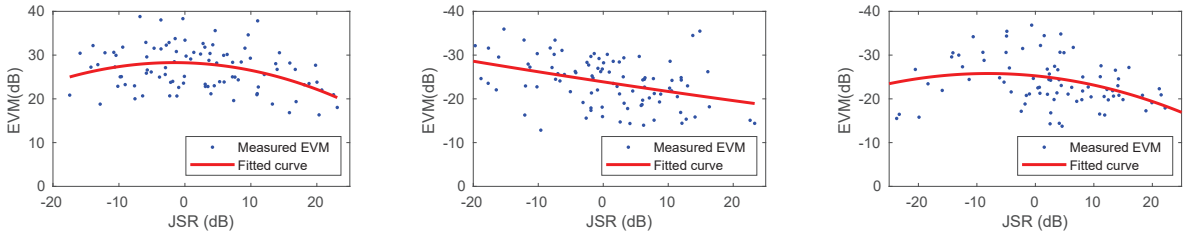
Figure 6.14: The distribution of the measured EVMs in different scenarios.

desired signal as shown in Fig. 6.13(a), where the measured EVM and throughput are -23.3 dB and 24 Mbps, respectively. Fig. 6.13(b) shows that the conventional receiver fails to decode the desired signal under the same jamming attack. As the baseline, Fig. 6.13(c) shows the constellation of decoded signal by a conventional receiver in a jamming-free environment. The EVM and corresponding throughput of the baseline are -27.8 dB and 30 Mbps, respectively. We can see that JammingBird is capable of mitigating the jamming signal and achieve 80.0% of the jamming-free throughput.

6.6.4 Experimental Results

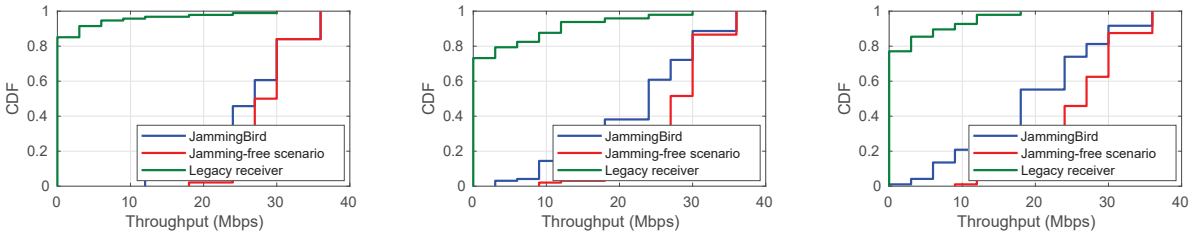
We have performed the previous test for all the test scenarios and recorded the following experimental results.

EVM. Fig. 6.14 shows the cumulative distribution function (CDF) of the measured EVMs for



(a) Parking lot scenario. (b) Local street scenario. (c) Highway scenario.

Figure 6.15: The measured EVMs versus JSR values.



(a) Parking lot scenario. (b) Local street scenario. (c) Highway scenario.

Figure 6.16: The achieved throughput at different test scenarios.

100 random realizations in each of the three test scenarios.

As illustrated in Fig. 6.14, JammingBird successfully suppresses the jamming signals and follows the baseline with a slight degradation in all test scenarios. In particular, the gap between the performance of JammingBird and the baseline is respectively 3.5 dB, 3.8 dB, and 6.3 dB, in parking lot, local street, and highway scenarios. The average measured EVM for the conventional 802.11p-based receiver is 4.5 dB in the parking lot scenario, 3.0 dB in the local street scenario, and 4.2 dB in highway scenario, respectively. The results indicate complete failure of conventional 802.11p-based receiver since data can be recovered if achieved EVM is less than -8.0 dB.

We sorted the measured EVMs of the decoded packets for different JSR values, as shown in Fig. 6.15. The fitted curves reflect the overall behavior of the measured EVMs. We can see that as the JSR values increase, the fitted curves of the measured EVMs gradually increase. This is mainly due to the error caused by the non-ideal jamming suppressor filter design and the interpolation error in practice, which are magnified as the power of jamming signal increases. Also, as shown

in Fig. 6.15, JammingBird can recover the desired data in the face of jamming signals with 25 dB stronger power than the desired ones.

Throughput. Fig. 6.16 shows the achieved throughput by JammingBird and conventional receiver in different test scenarios. The average achievable throughput of JammingBird is 26.2 Mbps in the parking lot, 22.2 Mbps in the local street, and 19.5 Mbps in the highway. Under the same test settings, the throughput of the conventional receiver is 6.4 Mbps in the parking lot, 3.1 Mbps in the local street, and 1.8 Mbps in the highway. On average, JammingBird achieves 18.9 Mbps higher throughput than the conventional receiver under constant jamming attacks. The average achievable throughput in the jamming-free scenario is 28.4 Mbps in the parking lot, 27.4 Mbps in the local street, and 26.0 Mbps in the highway. As such, while attacked by strong jamming signals, JammingBird can reach about 83.0% of the throughput of the conventional receiver in a jamming-free scenario.

6.7 Chapter Summary

In this chapter, we have designed JammingBird, a jamming-resilient receiver to secure vehicular communications against high-power constant jamming attacks. The enablers of JammingBird are two MIMO-based techniques: Jamming-resistant synchronizer and jamming suppressor. These two modules enable JammingBird to detect, synchronize, and recover desired signals under strong constant jamming attacks. We have implemented JammingBird on a proof-of-the-concept vehicular testbed and conducted extensive experiments to evaluate its performance under common vehicular test scenarios. Experimental results show that JammingBird is able to maintain 83.0% of throughput when legitimate communications undergo strong constant jamming attacks.

Chapter 7

Jamming-Resilient ZigBee Communications

ZigBee is a wireless communication technology that has been widely used to provide low-bandwidth wireless services for IoT applications such as building automation, medical data collection, and industrial equipment control. As ZigBee operates in the ISM radio frequency bands, it may suffer from unintentional interference from coexisting radio devices (e.g., WiFi and Bluetooth) and/or radio jamming attacks from malicious devices. Although many results have been produced to enhance ZigBee security, there is no technique that can secure ZigBee against jamming attack. In this chapter, we propose a new ZigBee receiver by leveraging MIMO technology, which is capable of decoding its desired signal in the presence of constant jamming attack. The enabler is a learning-based jamming mitigation method, which can mitigate the unknown interference using an optimized neural network. We have built a prototype of our proposed ZigBee receiver on a wireless testbed. Experimental results show that it is capable of decoding its packets in the face of 20 dB stronger jamming. The proposed ZigBee receiver offers an average of 26.7 dB jamming mitigation capability compared to off-the-shelf ZigBee receivers.

7.1 Introduction

ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create wireless local area networks for home automation, industrial equipment control, medical data collection, and other low-bandwidth needs. It is typically used for low data rate

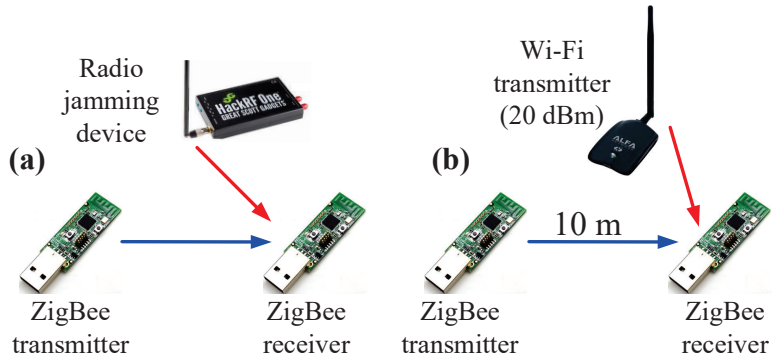


Figure 7.1: Illustrating the vulnerability of ZigBee communications. (a) ZigBee is under jamming attack; (b) ZigBee is under cross-network interference.

applications, with a defined data rate of 250 kbps. Its transmission range varies from 10 to 20 meters, depending on output power and environmental characteristics. ZigBee operates in the industrial, scientific, and medical (ISM) radio frequency bands. While other frequency bands are possible, most countries and regions in the world use 2.4 GHz for commercial ZigBee devices in indoor environments. With the rapid proliferation of IoT devices, ZigBee communications have become an important component of the telecommunication infrastructure in our society.

As ZigBee has been used for many crucial applications in real world, it is of great importance to secure ZigBee communications for reliable wireless connection. However, similar to other wireless technologies, ZigBee faces two challenges in practice. First, ZigBee devices share ISM radio frequency bands with other types of radio devices (e.g., WiFi and Bluetooth), and, therefore, suffer from unintentional interference from those coexisting devices. For example, a ZigBee device may suffer from interference from its co-located WiFi devices, and the interference may disrupt its communication. Second, due to the openness of wireless medium, ZigBee communications are vulnerable to radio jamming attacks. When a malicious device emits high-power jamming signal, all the ZigBee devices in its proximity will be unable to communicate.

One may think that ZigBee communications use spectrum spreading at PHY layer and, therefore, a ZigBee receiver is resilient to intentional or unintentional interference. This perception is not

correct. In ZigBee standard [124], the length of spectrum-spreading code sequence is 32 for every 4 bits. The jamming mitigation capability that it can offer is about $10\log_{10}(32/4) \approx 9$ dB, which is very limited. Fig. 7.1(a) shows a commercial ZigBee receiver in the face of a jamming device, which constantly sends 5 MHz noise-like interference. Our tests show that the ZigBee receiver frequently fails to decode its packets when its JSR is greater than -1.6 dB. Fig. 7.1(b) shows a commercial ZigBee receiver in the proximity of a commercial off-the-shelf WiFi device[1] that is *constantly* sending WiFi data packets. Our tests show that, when their distance is less than 5 meters, the ZigBee receiver suffers from larger than 90% packet error rate.

Although many results have been produced to enhance ZigBee security, there is no solution that can secure ZigBee against jamming attacks. The existing results in this domain are either focused on the enhancing the effectiveness of jamming attacks [125–127] or limited to the interference cancellation for cooperative devices such as WiFi [128–130]. Little progress has been made so far in the design of practical solutions to secure ZigBee against jamming attacks. The lack of effective solutions underscores the critical needs and grand challenges in this task.

In this chapter, we introduce a practical scheme to secure ZigBee communications against radio jamming attack (or unknown cross-network interference on ISM bands). The enabler is a new physical-layer design for a ZigBee receiver, making it capable of decoding its data packets in the presence of unknown interference. Our design relies on the assumption that a ZigBee device is equipped with two antennas. It leverages the spatial degrees of freedom (DoF) provided by its antennas to mitigate interference and decode its desired signal. One may argue that many ZigBee devices are powered by battery and, therefore, unsuited for multiple antennas. In fact, with the advancement of semiconductor and antenna technologies in the past decades, two antennas can be

[1]The WiFi device is an Alfa AWUS036NHA wireless USB adapter. We modified its firmware and driver in Linux to disable its carrier sense so that it can constantly send data packets at 20 dBm transmit power.

easily installed on a battery-powered ZigBee device. Moreover, many ZigBee-based IoT devices (e.g., electronic switches and industrial equipment) have sufficient power supply for their operations. Therefore, it is a mild assumption that a ZigBee device has two antennas in future IoT systems.

To decode ZigBee signal in the presence of unknown interference (jamming signal), we propose a learning-based method for jamming mitigation using a neural network at the physical layer. This neural network works as a linear spatial filter to suppress interference while not requiring any knowledge of the interference. A challenge in this method is the way of training the neural network so that it can decode the packets in real time. To address this challenge, we adopt a small-sized neural network that does not have hidden layers and optimize it by exploiting the inherent relationship of network weights to speed up the training process. ZigBee packet preamble (4 bytes or 32 bits) is then used to train the optimized neural network.

In addition to signal detection, another challenge in the design of jamming-resilient ZigBee receiver is time and frequency synchronization, where time synchronization is to search for the first chip of a packet and frequency synchronization is to compensate the frequency offsets. In the presence of interference, conventional correlation-based synchronization approach does not work. To address this challenge, we propose a projection-based approach for the synchronization component, which first projects received signals in the spatial domain and then employs the conventional approach to compensate the time and frequency offsets.

We have built a prototype of ZigBee receiver on a wireless testbed to validate our design in real-world wireless environments and evaluated its performance in the presence of a malicious device that emits different types of radio jamming signals. We placed the ZigBee transmitter, receiver, and jamming device at 20 different locations in a smart home environment. We examined three cases where a malicious radio attacker interferes with ZigBee receiver using WiFi-like, CDMA-like, or noise-like signal over full ZigBee spectrum. Experimental results show that our prototyped

ZigBee receiver offers an addition of 26.7 dB (on average) jamming mitigation capability (JMC) in comparison with an off-the-shelf ZigBee receiver. The results suggest that our designed ZigBee receiver can successfully decode ZigBee packets even if jamming signal is 20 dB stronger than ZigBee signal.

This work advances the state-of-the-art in the following aspects: i) We have proposed a learning-based jamming mitigation method using an optimized neural network, which is capable of decoding ZigBee signal in the presence of unknown interference. ii) Based on the learning-based jamming mitigation, we have designed a ZigBee receiver to decode its data packets in the face of malicious jamming attack. iii) We have built a prototype of our proposed ZigBee receiver and demonstrated its effectiveness in real-world wireless environments.

7.2 Related Work

We survey the prior research efforts in relevant to our work in the following three domains.

Jamming and Anti-jamming in ZigBee: While the security problems in Wi-Fi and cellular networks have received a large amount of research efforts and produced a large volume of research results (see, e.g., [35, 36, 131–134]), the security problems in ZigBee networks are highly overlooked. This stagnation is reflected by the lack of advances in the design of jamming-resistant ZigBee communications. Clearly, the existing anti-jamming schemes are limited to spectrum sharing (DSSS) technique. These schemes would not work when jamming signal is stronger than ZigBee signal at ZigBee receiver.

In contrast, our anti-jamming scheme takes advantage of recent advances in MIMO technology and renders much better ability of securing ZigBee communications in the presence of jamming attack.

Interference Cancellation in ZigBee Coexistence. Another research line related to this work is interference cancellation in the coexistence of ZigBee. In [128] and [129], the authors proposed WizBee, a coexistence scheme of ZigBee and WiFi, where the ZigBee device has a single antenna. They assumed that Wi-Fi signal is about 5 to 20 dB stronger than ZigBee signal, and thus employs interference cancellation to mitigate WiFi signal for ZigBee signal detection. This method does not apply to jamming defense because the ZigBee receiver does not have knowledge about the jamming signal. In [130], the authors studied the vulnerability of ZigBee devices to interference from 802.11 devices and proposed a solution for minimizing interference from 802.11 in ZigBee medical sensors. However, the proposed solutions are limited at the MAC layer and unsuited for jamming defense.

Learning-based Interference Management. Recently, machine learning (ML) becomes popular for wireless networking design, and there are many research results on learning-based interference management [135–138]. For example, in [135], the authors studied blind interference alignment (BIA) in wireless networks and proposed two reinforcement learning algorithms for selecting the best antenna configuration for BIA. These works, however, are limited to analytical study. So far, we find no prior work that employs neural network for real-time interference mitigation.

7.3 Problem Description

7.3.1 Jamming Attack Model

We consider the ZigBee network, as shown in Fig. 7.2, where a ZigBee router serves one or multiple ZigBee devices. At one moment, the ZigBee router communicates with a single ZigBee user device. In this network, there is a malicious device that continuously emits jamming signal

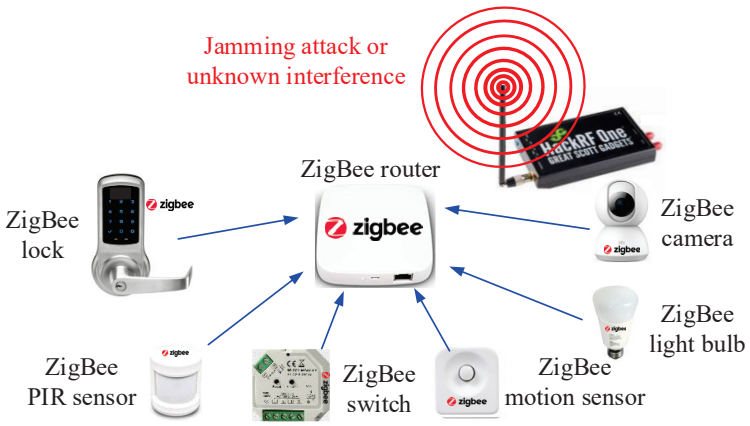


Figure 7.2: ZigBee for device and appliance control in the face of jamming attacks or unknown interference.

to disrupt the ZigBee communications, and we have the following assumptions on the jamming attack: i) The ZigBee devices have no knowledge about jamming signal, including its bandwidth, waveform, and frame format. ii) The bandwidth of jamming signal could be larger than, equal to, or less than the bandwidth of ZigBee signal. iii) The waveform of jamming signal may vary over time.

In real world, some ZigBee devices are not constrained by their physical size and their power consumption while playing a critical role in their applications. For example, many ZigBee-based electronic switches are connected to a main power supply and have a large size. These switches are widely used to control factory equipment and machines. In addition, ZigBee routers, which serve as the Internet gateway for ZigBee users as shown in Fig. 7.2, are not constrained by their physical size or power consumption. On such ZigBee devices, we can install multiple (two) antennas for radio signal transmission and reception.

Our objective is to secure the radio communications for the ZigBee devices, that have two or more antennas, against radio jamming attacks. Specifically, for the ZigBee devices that are equipped with two or more antennas, we design an efficient scheme to decode data packets in the presence of unknown interference, while not requiring any knowledge of interference.

7.3.2 Background of ZigBee Communications

Before presenting our design, we first offer a review of ZigBee PHY and MAC layers, which is essential for understanding of our new ZigBee receiver.

PHY-Layer Specs. ZigBee is based on IEEE 802.15.4 standard, which specifies operation in the unlicensed 2.4 to 2.4835 GHz (worldwide), 902 to 928 MHz (North America and Australia), and 868 to 868.6 MHz (Europe) ISM bands. Sixteen channels are allocated in the 2.4 GHz band. These channels are spaced 5 MHz apart, though using only 2 MHz of bandwidth. The radios use direct-sequence spread spectrum (DSSS) coding, and the spectrum-spreading code sequence comprises pre-defined 32 chips, as specified in Table 7.1. For ZigBee devices working in the 2.4 GHz band, offset quadrature phase-shift keying (O-QPSK) is used. In O-QPSK, two chips are modulated onto the in-phase and q-phase carriers, and the over-the-air data rate is 250 kbps per channel. For indoor applications at 2.4 GHz, transmission distance ranges from 10 to 20 meters, depending on the construction materials, the number of walls to be penetrated, and the output power permitted in that geographical location.

MAC Protocols. The current IEEE 802.15.4 standards [139] support two types of networks: Beacon-enabled and non-beacon-enabled networks. In non-beacon-enabled networks, CSMA/CA is used for medium access control. In this type of network, at least one ZigBee device keeps its radio receiver active, listening to possible packets from other ZigBee devices; while other ZigBee devices would remain asleep until they are commanded to transmit. The typical example of such a network is a wireless light switch controller: The ZigBee chipset inside a lamp may continuously receive signals, since it is connected to the main supply, while a battery-powered wireless remote controller would remain asleep until the switch is triggered. The remote controller then wakes up to send a command packet to the lamp, and returns to sleep after receiving an acknowledgment.

Table 7.1: The mapping from data bits to chip sequence [124].

Binary data ($b_0b_1b_2b_3$)	Symbol value	Chip values ($c_0c_1c_2...c_{32}$)
0000	0	11011001110000110101001000101110
1000	1	11101101100111000011010100100010
0100	2	00101110110110011100001101010010
1100	3	00100010111011011001110000110101
0010	4	01010010001011101101100111000011
1010	5	00110101001000101110110110011100
1110	6	11000011010100100010111011011001
1110	7	10011100001101010010001011101101
0001	8	10001100100101100000011101111011
1001	9	10111000110010010110000001110111
0101	10	01111011100011001001011000000111
1101	11	01110111101110001100100101100000
0011	12	00000111011110111000110010010110
1011	13	01100000011101111011100011001001
0111	14	10010110000001110111101110001100
1111	15	11001001011000000111011110111000

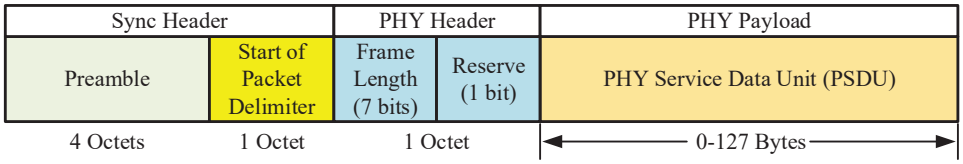


Figure 7.3: The frame structure of ZigBee communications.

In beacon-enabled networks, the special network nodes called ZigBee routers transmit periodic beacons to announce their presence to the other nodes. Beacon intervals depend on data rate; they may range from 15.4 milliseconds to 251.6 seconds at 250 kbps. Nodes may sleep between beacons, thus lowering their duty cycle and prolonging their battery lifetime.

ZigBee Frame Structure. Fig. 7.3 shows the frame structure of a ZigBee data packet at the physical layer, which comprises three parts: Sync header, PHY header, and PHY payload. Particularly, a ZigBee frame has a preamble in its sync header, which consists of 4 pre-defined Octets (32 bits). The preamble is used by the ZigBee receivers to obtain chip and symbol synchronization for an incoming message. In the standards, the preamble is composed of 32 binary zeros. As we shall see, this preamble plays a key role in our design of jamming-resilient ZigBee receiver, which

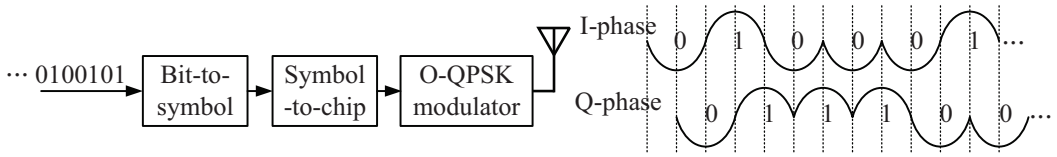


Figure 7.4: The PHY-layer diagram of a conventional ZigBee transmitter and an example of O-QPSK waveform.

uses the preamble to train an optimized neural network for jamming mitigation.

ZigBee Transmitter Diagram. Fig. 7.4 shows the PHY-layer diagram of a conventional ZigBee transmitter and an example of generated O-QPSK signal. As shown in the figure, the bit-to-symbol module first groups every 4 bits as a symbol, with its value in the range from 0 to 15. Then, each of the resulting symbols is mapped to a sequence of predefined 32 chips, as specified in Table 7.1. Finally, the sequence chips are O-QPSK modulated using half-sine pulse shaping filter, and the resulting I/Q signals are sent for radio frequency transmission.

ZigBee Receiver Diagram. Fig. 7.5 shows the diagram of a conventional ZigBee receiver. The RF front-end module first converts a radio signal to the corresponding baseband signal, followed by a module for energy detection. Then, the analog signal is converted to digital samples using $12\times$ oversampling rate. A matched filter is used to suppress noise and $3\times$ down-sample the digital signal. After that, frequency synchronization and timing recovery are performed to decode the chips, which are further used for symbol detection (preamble detection and phase ambiguity elimination). Finally, the decoded chips are despread to estimate the original chips. Similar to other wireless receivers, conventional ZigBee receivers are vulnerable to both jamming attacks and unknown interference.

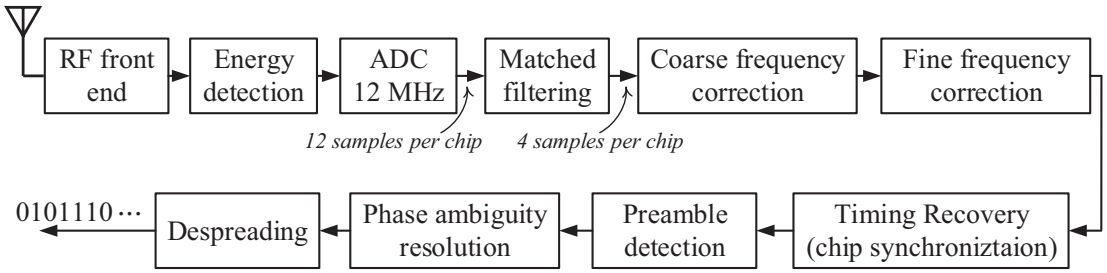


Figure 7.5: The PHY-layer diagram of a conventional ZigBee receiver.

7.4 A New ZigBee Receiver Design

To enable ZigBee communication in the presence of jamming attack, we need a ZigBee receiver that is immune to unknown interference. In what follows, we first describe the basic idea of our design and then present its key components.

7.4.1 Basic Idea

The basic idea of our design is to install two antennas on a ZigBee device by leveraging the recent advances in semiconductor and antenna technologies. This is possible for many ZigBee devices that are not constrained by their physical size or power consumption (e.g., ZigBee hubs and ZigBee electronic switches). For a ZigBee device with two or more antennas, we design a new baseband signal processing pipeline to mitigate the jamming signal and recover ZigBee signal.

Fig. 7.6 shows the diagram of our proposed signal processing scheme for a ZigBee receiver. Compared to the conventional ZigBee receiver in Fig. 7.5, it has two new modules: Synchronization module and jamming mitigation module. Other modules remain the same as those in conventional ZigBee receiver. In what follows, we focus on these two new modules.

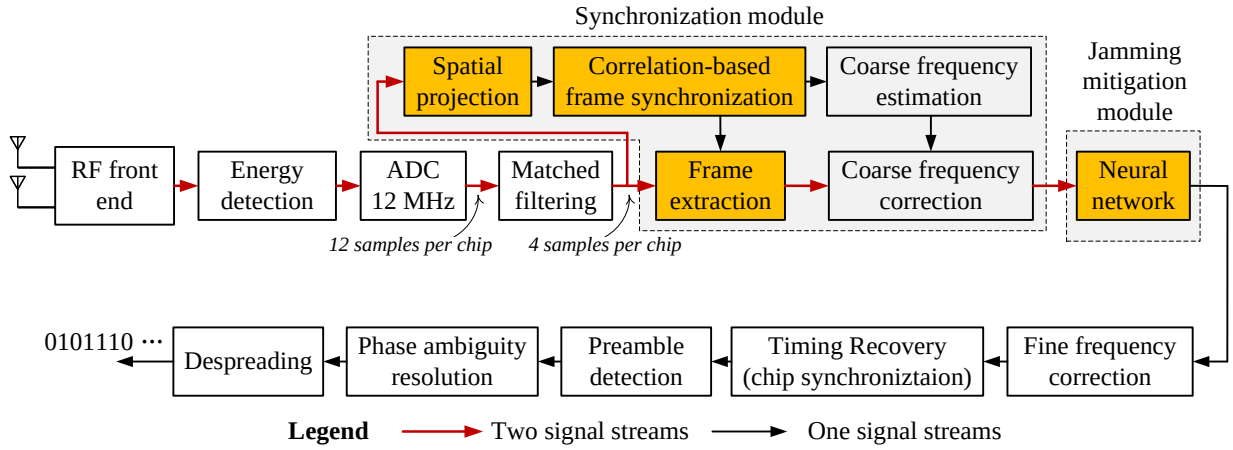


Figure 7.6: The diagram of our proposed ZigBee receiver for decoding ZigBee packets in the face of jamming signal.

7.4.2 Synchronization Module

In the conventional ZigBee receiver, the sync module has two purposes: i) Estimate the carrier frequency offset and compensate the coarse frequency offset for the received signal; and ii) identify the beginning of a signal frame. To achieve these two purposes, the conventional method performs FFT operation to estimate the frequency offset and use correlation to estimate the time offset. This method, however, does not work for a ZigBee receiver in the face of unknown interference, necessitating a new sync method to estimate the frequency and time offset for the received signal.

To address this challenge, we propose a projection-based method for the alleviation of jamming signal in the spatial domain. Here, projection refers to a filtering operation on the two data streams using a linear spatial vector. Fig. 7.7 illustrates the basic idea of our method. Consider the network in Fig. 7.7(a), where a two-antenna ZigBee receiver suffers from jamming attacks. If the jamming signal is much stronger than the ZigBee signal, the frequency and timing offset cannot be accurately estimated at the ZigBee receiver and, as a result, the ZigBee signal cannot be decoded. To alleviate the jamming signal, we project the received signal to a spatial direction using a spatial filter $\mathbf{p}$, as shown in Fig. 7.7(b). If we can find a good projection direction (e.g., the one perpendicular

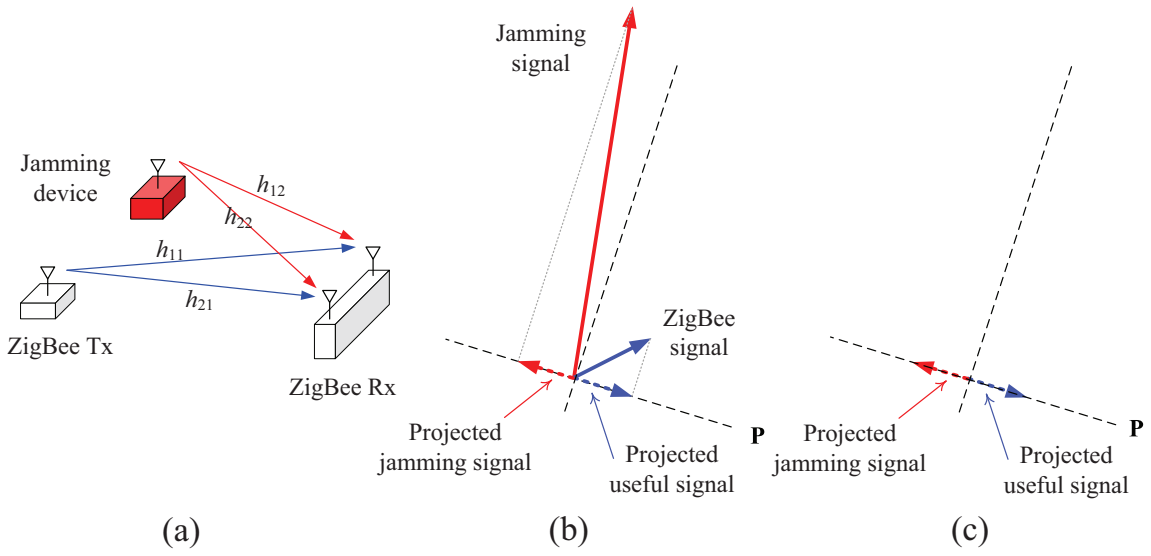


Figure 7.7: Illustrating the basic idea of our synchronization method. (a) An example of toy-sized network consisting of a ZigBee transmitter, a ZigBee receiver, and high-power jammer. (b) Signal projection in the time domain at the ZigBee receiver, where projection filter $\mathbf{p}$ is a 2×1 complex vector. (c) Amplitudes of jamming and ZigBee signals after projection.

to the jamming signal direction), then the jamming signal would be significantly weakened on the projection direction, as illustrated in Fig. 7.7(c).

Now, the question is how to find a good direction for signal projection. We resort to matrix decomposition, and it turns out that a singular vector of the received signals is an effective direction for jamming alleviation. Mathematically, denote $\mathbf{y}(n) \in \mathbb{C}^{2\times 1}$ as the input of the "signal projection" module in Fig. 7.6, and $y(n) \in \mathbb{C}$ as the output of this module. Then, we construct the projection filter by $\mathbf{p} = singularvector(\sum_{n=1}^{N_{\text{sync}}} \mathbf{y}(n)\mathbf{y}(n)^{\mathsf{H}})$, where N_{sync} is the number of received signal samples for projection. It can be set empirically (e.g., 30). After constructing $\mathbf{p}$, we project the two signal streams by letting $y(n) = \mathbf{p}^{\mathsf{H}}\mathbf{y}(n)$. For this synchronization module, we have the following lemma.

Lemma 1 If the wireless channels are frequency-flat[2] and there is no noise, then the signal-to-jamming ratio (SJR) after signal projection is greater than or equal to 0 dB, regardless of the

[2]Frequency-flat wireless channel is a channel where all frequency components of a signal experience the same response.

jamming signal power before the projection.

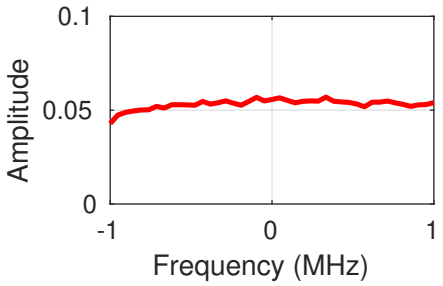
The proof of this lemma is given in Appendix. Fig. 7.8 shows the measured wireless channel over 10 m line-of-sight (LoS) distance, and Fig. 7.9 shows the measured wireless channel over 10 m non-line-of-sight (NLoS) distance. We can see that the channels are relatively frequency-flat. Moreover, since ZigBee is for short-range communication, the noise is small in many scenarios. Therefore, we expect this projection-based method has a performance close to its theoretical limit in (1). It is noteworthy that the conventional sync method is resilient to the interference that has similar power as the signal (i.e., SJR is about 0 dB).

7.4.3 Jamming Mitigation Module

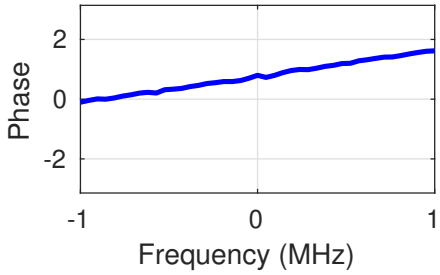
Following the signal processing pipeline in Fig. 7.6, after the compensation of coarse time and frequency offsets, the two signal streams are fed into a neural network, which is used to mitigate jamming signal for the recovery of the ZigBee signal. After the mitigation of jamming signal, the rest of the signal processing modules remain the same as those in the conventional ZigBee receiver as shown in Fig. 7.5. Then, the key question is how to design the neural network such that it can mitigate the jamming signal to the maximum extent while preserving the ZigBee signal. We address this question in the next section.

7.5 Learning-based Jamming Mitigation

In this section, we present a learning-based method to mitigate the jamming signals. We first formulate the jamming mitigation problem as a mathematical problem and then propose a learning-based method for jamming mitigation.

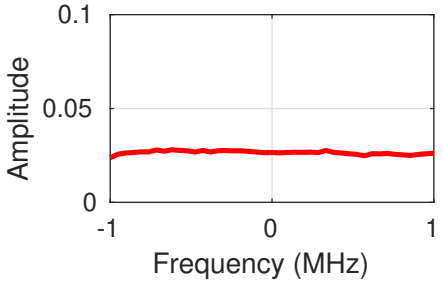


(a) Channel amplitude.

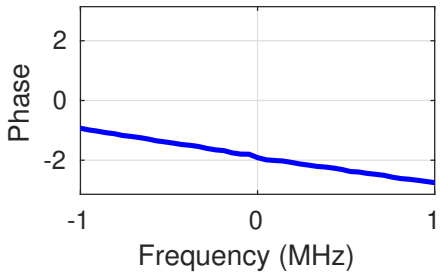


(b) Channel phase.

Figure 7.8: An instance of measured line-of-sight wireless channel in ZigBee communication.



(a) Channel amplitude.



(b) Channel phase.

Figure 7.9: An instance of measured non-line-of-sight wireless channel in ZigBee communication.

7.5.1 Problem Formulation

While ZigBee channels are spaced 5 MHz, ZigBee signal bandwidth is about 2 MHz. Moreover, ZigBee is typically used for short-range communications. Therefore, we assume that the radio signals in ZigBee communications experience frequency-flat wireless channels. As shown in Fig. 7.8 and Fig. 7.9, wireless channels are pretty flat over frequency in both real-world LoS and NLoS scenarios.

Based on this assumption, we formulate the jamming mitigation problem as a mathematical problem. Denote $\mathbf{y}(n) \in \mathbb{C}^{2\times N}$ as the two input signal streams of the neural network module in Fig. 7.6, where N is the number of samples in a ZigBee frame as shown in Fig. 7.3. Denote $x(n) \in \mathbb{C}^{1\times N}$ as the transmitted ZigBee signal. Denote $z(n) \in \mathbb{C}^{1\times N}$ as the transmitted jamming

signal. Then, the received signal $\mathbf{y}(n)$ can be expressed as:

$$\mathbf{y}(n) = \mathbf{h}_1 x(n) + \mathbf{h}_2 z(n) + w(n), \tag{7.1}$$

where $\mathbf{h}_1 = [h_{11} \ h_{21}]^\mathsf{T}$ is the channel coefficients between ZigBee transmitter and ZigBee receiver, $\mathbf{h}_2 = [h_{12} \ h_{22}]^\mathsf{T}$ is the channel coefficients between jamming device and ZigBee receiver, as shown in Fig. 7.7. $w(n)$ is the noise at the ZigBee receiver.

To mitigate jamming signal, we need to find a filter $\mathbf{g} = [g_1 \ \ g_2] \in \mathbb{C}^{1\times 2}$ that can mitigate the jamming signal through properly combining the two signal streams. We intend to design filter satisfying the following requirements: $\mathbf{g}\mathbf{h}_1 = 1$ and $\mathbf{g}\mathbf{h}_2 = 0$. If we could obtain channel coefficients $\mathbf{h}_1$ and $\mathbf{h}_2$, then it is a trivial task to compute $\mathbf{g}$. After obtaining $\mathbf{g}$, we can mitigate the jamming signal by letting $\hat{x}(n) = \mathbf{g}\mathbf{y}(n)$, where $\hat{x}(n)$ is the signal after jamming mitigation, (i.e., the output of the neural network module).

However, in real systems, due to the lack of knowledge about jamming signal, there is no solution that can estimate the channel coefficients in the presence of jamming signal, making it challenging to mitigate jamming signal.

7.5.2 Optimized Neural Network for Jamming Mitigation

Challenge and Setting. To mitigate jamming signal, we resort to a neural network to replace the spatial filter $\mathbf{g}$. A challenge in this method is that the neural network should work in real time to decode the ZigBee packets. In other words, the neural network should be capable of mitigating the jamming signal for each individual ZigBee packet. To address this challenge, we adopt a small-size neural network that does not have hidden layers as shown in Fig. 7.10. This neural network works with real numbers, where the input is the real and imaginary parts of two signal streams (i.e.,

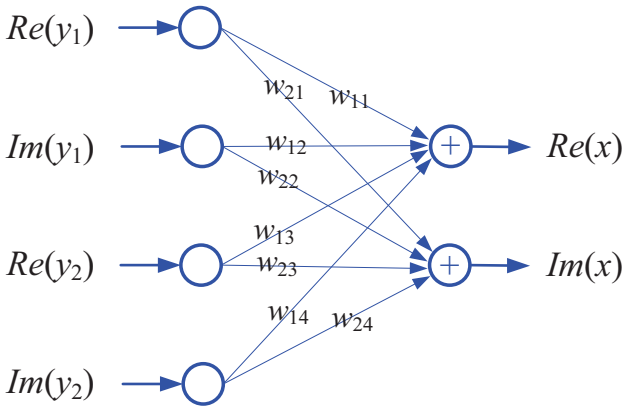


Figure 7.10: A neural network for jamming mitigation.

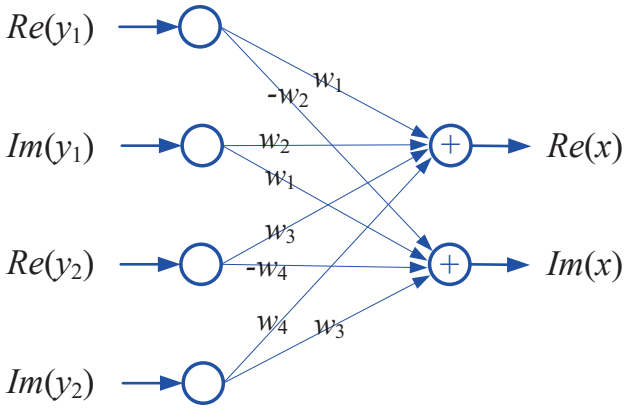


Figure 7.11: A simplified neural network for jamming mitigation.

$\mathbf{y}(n) = [y_1(n) \;\; y_2(n)]$) and the output is the real and imaginary parts of one signal stream (i.e., $x(n)$). As neural network works with real numbers by nature, we decompose a complex number into two real numbers, with $Re(\cdot)$ and $Im(\cdot)$ being its real and imaginary parts.

Data for Training. This simple neural network is trained by each individual packet. To train this network, we use the preamble in the ZigBee frame. As shown in Fig. 7.3, a ZigBee frame has a preamble field, which comprises 4 pre-defined Octets (32 bits). For these 32 bits, every four are modulated to a spectrum-spreading code sequence of 32 chips. Therefore, the preamble of a ZigBee frame has 256 chips, which we use to train the neural network.

Neural Network Optimization. To realize real-time packet detection, we propose a scheme to

speed up the training process. Our method takes advantage of the inherent relation of the network weights. Suppose that the noise is negligible. Then, we have

$$\begin{aligned}
\mathbf{gy} &= g_1 y_1 + g_2 y_2 \\
&= [Re(g_1) + iIm(g_1)][Re(y_1) + iIm(y_1)] \\
&\quad + [Re(g_2) + iIm(g_2)][Re(y_2) + iIm(y_2)] \\
&= \big[Re(g_1)Re(y_1) - Im(g_1)Im(y_1) \\
&\quad + Re(g_2)Re(y_2) - Im(g_2)Im(y_2)\big] \\
&\quad + i\big[Re(g_1)Im(y_1) + Im(g_1)Re(y_1) \\
&\quad + Re(g_2)Im(y_2) + Im(g_2)Re(y_2)\big]. \qquad (7.2)
\end{aligned}$$

Define four real numbers as follows: $w_1 = Re(g_1)$, $w_2 = -Im(g_1)$, $w_3 = Re(g_2)$, and $w_4 = -Im(g_2)$. Then, (7.2) can be written to

$$\begin{aligned}
\mathbf{gy} = &\underbrace{\big[w_1 Re(y_1) + w_2 Im(y_1) + w_3 Re(y_2) + w_4 Im(y_2)\big]}_{Re(x)} \\
&+ i\underbrace{\big[w_1 Im(y_1) - w_2 Re(y_1) + w_3 Im(y_2) - w_4 Re(y_2)\big]}_{Im(x)}. \qquad (7.3)
\end{aligned}$$

Based on (7.3), the weights in the neural network in Fig. 7.10 can be re-written as those in Fig. 7.11. It is evident that the new neural network has only four weights, less than the number of weights in Fig. 7.10.

Training Method. When a ZigBee device receives a packet as shown in Fig. 7.3, it uses the packet preamble to train the neural network. Specifically, to train weights $\{w_1, w_2, w_3, w_4\}$ as shown in Fig. 7.11, we transform the network into that shown in Fig. 7.12. For this neural network,

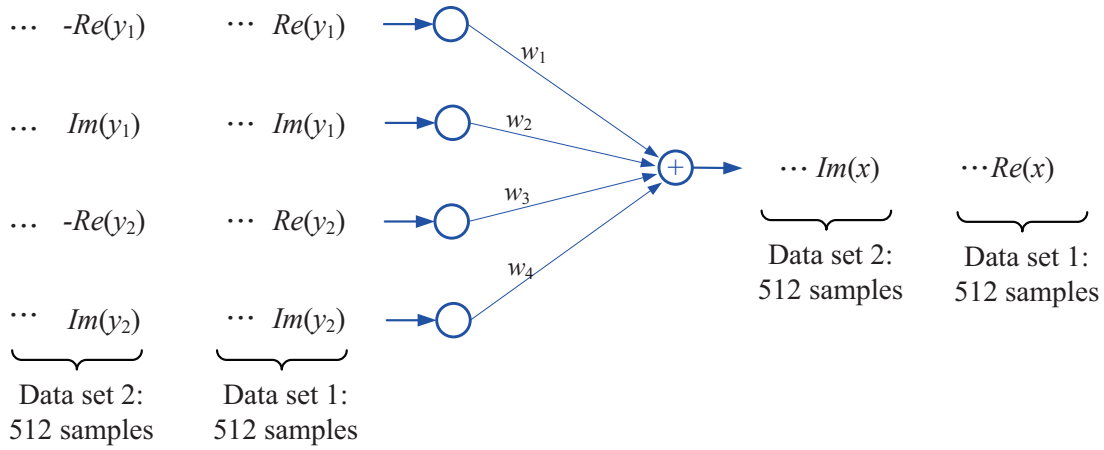


Figure 7.12: An optimized neural network used for weight training.

Algorithm 1 Neural network training process.

1: **Input:** Transmitted preamble $x(n)$ of a ZigBee packet, and received preamble $\mathbf{y}(n)$ of the ZigBee packet, $1 \leq n \leq 512$
2: **Output:** The weights of neural network $[w_1 \;\; w_2 \;\; w_3 \;\; w_4]$
3: Obtain $[Re(x(n)) \;\; Im(x(n))]$, $1 \leq n \leq 512$
4: Obtain $[Re(y_1(n)) \;\; Im(y_1(n)) \;\; Re(y_2(n)) \;\; Im(y_2(n))]$, $1 \leq n \leq 512$
5: **for** $1 \leq n \leq 512$ **do**
6: *// the nth iteration*
7: Train the neural network using the following data: $[Re(y_1(n)) \;\; Im(y_1(n)) \;\; Re(y_2(n)) \;\; Im(y_2(n))]$ for input and $Re(x(n))$ for output
8: Train the neural network using the following data: $[Im(y_1(n)) \;\; -Re(y_1(n)) \;\; Im(y_2(n)) \;\; -Re(y_2(n))]$ for input and $Im(x(n))$ for output
9: **end for**
10: Return $[w_1 \;\; w_2 \;\; w_3 \;\; w_4]$

we have a total of $128 \times 4 \times 2 = 1024$ samples to update its four weights, where 128 is the number of complex symbols in the preamble, 4 is oversampling rate, and 2 is the number of components in a complex number (real and imaginary). As shown in the figure, the first 512 samples correspond to the real part of the 256 chips in the frame preamble, and the second 512 samples correspond to the imaginary part of the 256 chips in the frame preamble.

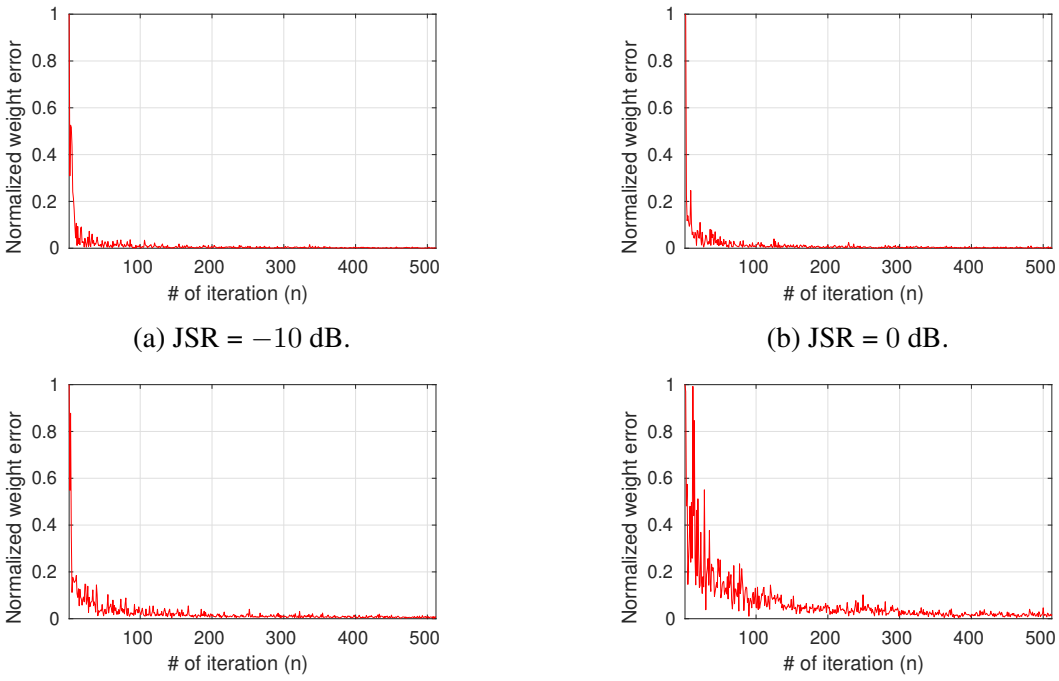
Alg. 1 shows our training method for the neural network. In this algorithm, we use backpropagation to train the weights for the neural network and the squared error as neural network cost

function. In the training process, we use an adaptive step size for the update of weights. Specifically, for the 512 samples in dataset 1 in Fig. 7.12, we update the weights as follows: $\{w_1, w_2, w_3, w_4\} \leftarrow \{w_1, w_2, w_3, w_4\} + \lambda(n)\,[Re(y_1(n)) \;\; Im(y_1(n)) \;\; Re(y_2(n)) \;\; Im(y_2(n))][Re(x(n)) - Re(\tilde{x}(n))]$, where $\lambda(n)$ is the step size for the nth iteration and $Re(\tilde{x}(n))$ is the forward computation output of the nth iteration. For the 512 samples in dataset 2 in Fig. 7.12, we update the weights as follows: $\{w_1, w_2, w_3, w_4\} \leftarrow \{w_1, w_2, w_3, w_4\} + \lambda(n)[Im(y_1(n)) \;\; -Re(y_1(n)) \;\; Im(y_2(n)) \;\; -Re(y_2(n))][Im(x(n)) - Im(\tilde{x}(n))]$, where $Im(\tilde{x}(n))$ is the forward computation output of the nth iteration. In the nth iteration, we set $\lambda(n) = \frac{w_1^2+w_2^2+w_3^2+w_4^2}{10+n}$, $1 \leq n \leq 512$.

One may wonder why the preamble of each packet is enough for the training of neural network. The reasons are actually twofold. First, the neural network is of small size. It does not have hidden layer, and it has only 4 weights to train. Given its small size, it can be envisaged that the training process converges fast over training samples. Second, as we detailed before, the preamble of each ZigBee packet has 512 independent data samples that we can use to train the neural network. This number is not small, and it is sufficient to train those 4 weights.

It is noteworthy that, different from conventional neural networks, which use a portion of data for training and the remainder for test, our neural network uses all the data (packet preamble) for training and does not have test phase. This is because our neural network is of very small size and required to be run in real time. In essence, it is a heuristic algorithm.

As shown in Alg. 1, the training involves 512 iterations, each of which has 8 multiplications and 10 additions in the forward calculation, as well as 21 multiplications and 12 additions in the backward calculation. Collectively, the training algorithm requires 14,848 multiplications and 11,264 additions. We note that, given the advances in semiconductor and battery technologies in the past decades, the training algorithm can be easily carried out by a battery-powered ZigBee device. Hence, the power consumption of our proposed ZigBee receiver should not be an issue in



(a) JSR = -10 dB.

(b) JSR = 0 dB.

(c) JSR = 10 dB.

(d) JSR = 20 dB.

Figure 7.13: Normalized weight errors over the number of training iterations in different JSR scenarios.

practice. We also note that the new design of our proposed ZigBee receiver lies at the physical layer, which will produce no impact on the topology of a ZigBee network.

Jamming Mitigation. With the weights $\{w_1, w_2, w_3, w_4\}$ from the neural network, we use them to construct the spatial filter $\mathbf{g} = [w_1 - iw_2 \;\; w_3 - iw_4]$. Then, the jamming mitigation is conducted by $\hat{x}(n) = \mathbf{g}\mathbf{y}(n)$, where $\hat{x}(n)$ is the output of the neural network module in Fig. 7.6. As shown in Fig. 7.6, output signal stream $\hat{x}(n)$ is sent to the fine frequency correction module, timing recovery module, preamble detection module, phase ambiguity module, and despreading module. These modules are identical to those in conventional ZigBee receivers, as shown in Fig. 7.5.

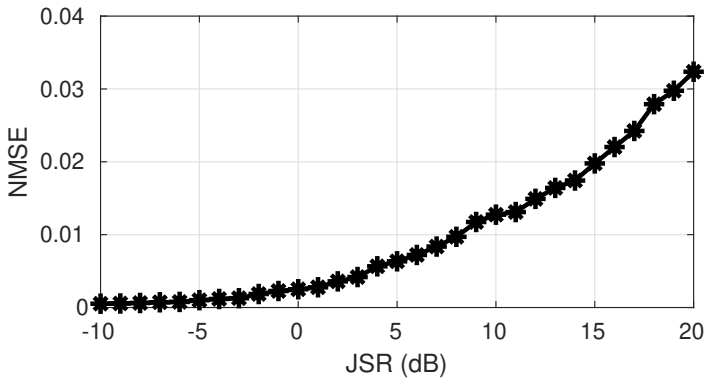


Figure 7.14: Normalized mean square error (NMSE) of decoded signal in the presence of jamming attack when using the proposed neural network for signal detection.

7.5.3 Performance Analysis

In what follows, we study the performance of this learning-based jamming mitigation in the scenarios without noise. For its performance in realistic noisy scenarios, we resort to experimental evaluation, as presented in Section 7.6.

Convergence. We first study the convergence of the neural network by observing the fluctuation of its weights. Specifically, we define the normalized weight error as follows:

$$error = \sqrt{\frac{\sum_{i=1}^{4}[w_i(n) - w_i(n-1)]^2}{\sum_{i=1}^{4} w_i(n)^2}}, \tag{7.4}$$

where n is the number of training iterations in Alg. 1. Based on this definition, we implement this neural network and observe its weights over training iterations in a case study. Fig. 7.13 shows the normalized weight error over the number of training iterations in different JSR scenarios. It is evident that the weights converge quickly. With 512 samples for training, the fluctuation of weight errors is limited 0.1% when JSR is -10 dB, 0.1% when JSR is 0 dB, 0.9% when JSR is 10 dB, and 1.7% when JSR is 20 dB.

Learning-based Signal Detection. We now study the performance of the neural network in

signal detection using extensive simulation. Denote x as the original symbol at ZigBee Tx. Denote $\hat{x}$ as the estimated symbol in the presence of jamming signal at ZigBee Rx. We define normalized mean square error (NMSE) as $\mathrm{NMSE} = (\mathbb{E}\big|x - \hat{x}\big|^2)/(\mathbb{E}\big|x\big|^2)$. Our simulation results show that the NMSE is less than 4% when JSR is 20 dB. This means that the proposed learning-based method can decode ZigBee packet in zero-noise scenarios even if jamming signal is 20 dB stronger than ZigBee signal.

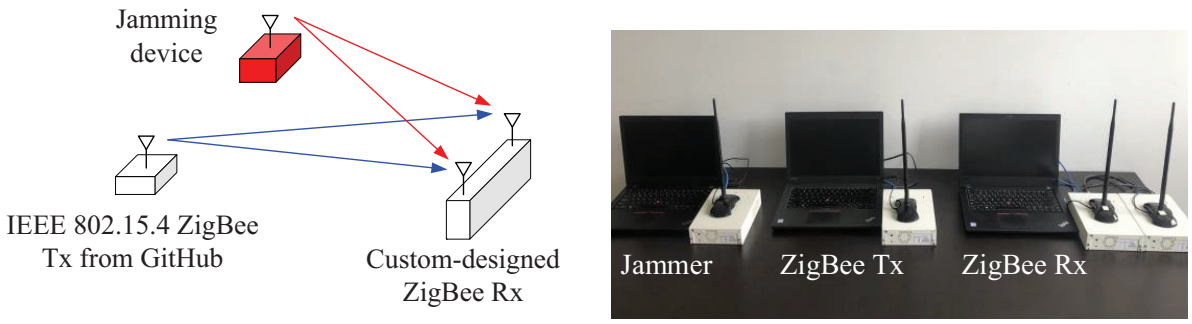
7.6 Experimental Evaluation

7.6.1 Implementation and Experimental Settings

We built a prototype of the ZigBee communication network, as shown in Fig. 7.15, to evaluate the proposed ZigBee receiver.

ZigBee Transmitter. We built the ZigBee transmitter using an USRP N210 device [39] and a laptop with GNURadio software package [140]. We have used the IEEE 802.15.4 ZigBee open-source code from GitHub [141] to implement the ZigBee transmitter. The open-source ZigBee code employs the PHY protocols as that of commercial off-the-shelf ZigBee transmitters. The frame structure in Fig. 7.3 is used for data transmission, where the length of PHY Service Data Unit (PSDU) is 32 bytes. We have run the ZigBee transmitter using O-QPSK modulation with over-the-air bit rate of 250 kbps. The carrier frequency is 2.48 GHz, and the sampling rate is 12 MHz. The transmit power of this ZigBee transmitter was set 13 dBm.

ZigBee Receiver. We have implemented two types of ZigBee receiver. i) Conventional single-antenna ZigBee device: For this ZigBee receiver, we implemented it by installing the IEEE 802.15.4 ZigBee open-source code from GitHub [141] on an USRP N210 device. ii) Our prototype of two-antenna ZigBee receiver: We built this ZigBee receiver using two USRP N210 devices,



(a) Network setting. (b) Network devices.

Figure 7.15: Network setting and devices used in our experiments.

which were connected through a MIMO cable, as shown in Fig. 7.15(b). These two USRP N210 devices were connected to a laptop, on which we implemented our design shown in Fig. 7.6.

Radio Jamming Device. We have built a radio jamming device using a USRP device and GNURadio software package. The jamming device was able to employ three types of waveforms as follows:

- **WiFi-like Jamming:** We use the legacy WiFi frame consisting of 4 OFDM symbols as preamble and 16 OFDM symbols as random payload. The total number of subcarriers are 64, and 52 subcarriers are used to carry payload. The effective bandwidth is about 4.1 MHz, and the symbol duration is 16 μs (12.8 μs OFDM symbol prepended by a 3.2 μs cyclic prefix).

- **CDMA-like Jamming:** A random bit stream is constantly modulated onto the carrier frequency using QPSK modulation and rectangular I/Q pulse shaping filters. The effective bandwidth is 5 MHz.

- **Noise-like Jamming:** A zero-mean complex Gaussian signal is modulated onto the carrier frequency. The symbol duration is 0.2 μs, and the effective bandwidth is 5 MHz.

The transmit power of the jamming device was set to 20 dBm, and its bandwidth was set to 5 MHz. We note that, since the bandwidth of jamming signal is larger than that of ZigBee signal,

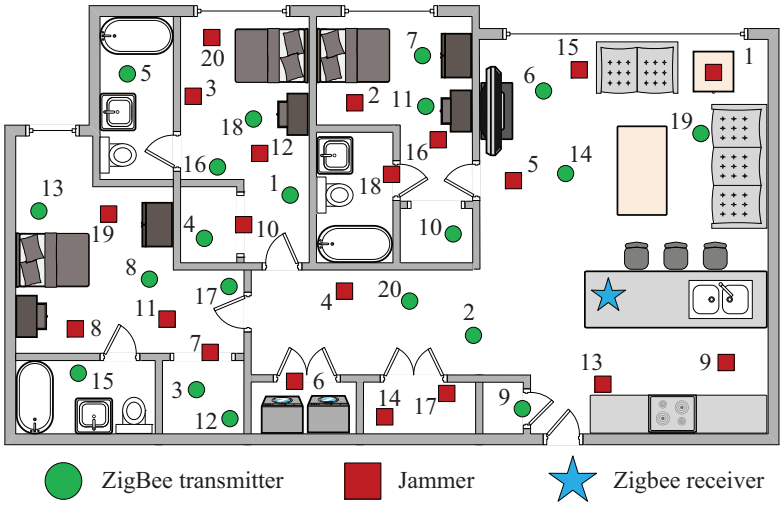


Figure 7.16: Floor plan of our experimentation.

this bandwidth is sufficient for jamming attack. In real systems, any out-of-band jamming signal will be filtered out by ZigBee device's RF filter and produce no impact on ZigBee device.

Experimental Settings. Fig. 7.16 shows the testbed settings of our experiments. The ZigBee receiver was placed on an office desk, as marked a blue star in the figure. For ease of experimentation, the ZigBee receiver was stationary without movement throughout our experiments. This setting could be justified by real-world ZigBee applications, where most of ZigBee Hubs are placed at a fixed spot to provide services. The ZigBee transmitter was placed at one of the 20 locations marked green circles in the figure, and the radio jamming device was placed at one of the 20 locations marked red boxes. These 20 green/red boxes were randomly selected on the floor to cover the whole home area. The settings of each individual devices in our experiments were specified previously.

7.6.2 Performance Metrics

We evaluate the performance of the ZigBee receiver using the following four metrics:

Jamming-to-Signal Ratio (JSR). Focusing on the radio signal received by the ZigBee receiver

Table 7.2: Empirical PRR of our proposed and conventional ZigBee receivers.

Location index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Conventional receiver	0	0.6173	0.6174	0	0	0.6173	0	0	0	0.1756	0.6922	0.1755	0	0.9506	0	0	0	0	0.6168	0.9009
Proposed receiver	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

before jamming mitigation, we define JSR as

$$\mathrm{JSR} = 10\log_{10}\left(\frac{\sum_{n=1}^{N_m}|\mathbf{y}_j(n)|^2}{\sum_{n=1}^{N_m}|\mathbf{y}_s(n)|^2}\right), \tag{7.5}$$

where $\mathbf{y}_j(n)$ is the received jamming signal at the ZigBee receiver when ZigBee transmitter has been turned off, $\mathbf{y}_s(n)$ is the received ZigBee signal at the ZigBee receiver when jamming device has been turned off, and N_m is the number of measured signal samples (e.g., $N_m = 2000$).

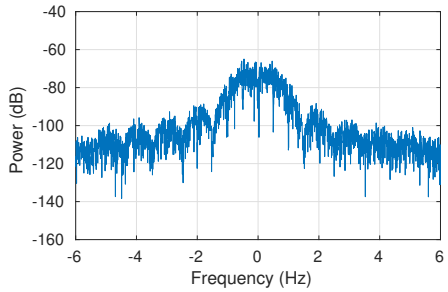
EVM. Focusing on the signal at the ZigBee receiver after jamming mitigation, we define EVM as: $\mathrm{EVM} = 10\log_{10}\left(\frac{\sum_{n=1}^{N_c}\left|x(n)-\hat{x}(n)\right|^2}{\sum_{n=1}^{N_c}\left|x(n)\right|^2}\right)$, where $x(n)$ is the original chips at the ZigBee transmitter, $\hat{x}(n)$ is the estimated chips at the ZigBee receiver, and N_c is the number of chips.

Packet Reception Rate (PRR). PRR is defined as the ratio of successfully decoded packets per total transmitted packets.

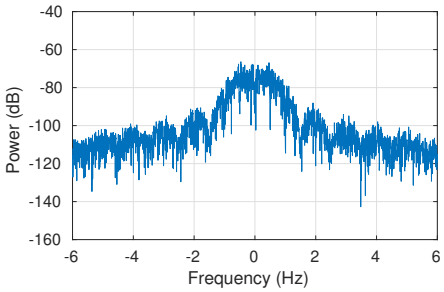
Jamming Mitigation Capability (JMC). Based on the measured JSR and EVM, we define JMC as the gap between JSR and EVM, i.e., $\mathrm{JMC} = \mathrm{JSR} - \mathrm{EVM}$.

7.6.3 A Case Study

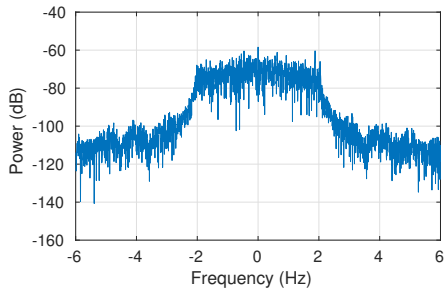
To evaluate the performance of the designed ZigBee receiver, we consider the case where the ZigBee transmitter is placed at location 1 (small green circle) and the jamming device is placed at location 1 (small red square) in Fig. 7.16. The jamming device emits WiFi-like jamming signal to disrupt ZigBee communications. We first study the performance of proposed ZigBee receiver at the designated location. Fig. 7.17(a) and (b) show the received power spectrum for ZigBee signal



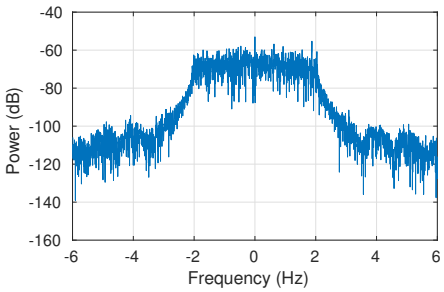
(a) Received ZigBee signal on antenna 1.



(b) Received ZigBee signal on antenna 2.



(c) Received ZigBee and jamming signals on antenna 1.



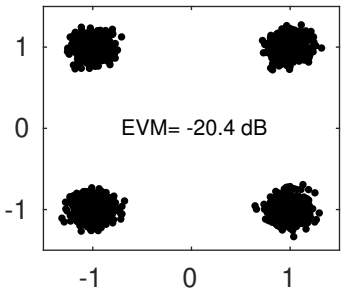
(d) Received ZigBee and jamming signals on antenna 2.

Figure 7.17: The power spectrum of received signals on the ZigBee receiver's two antennas with and without jamming signal.

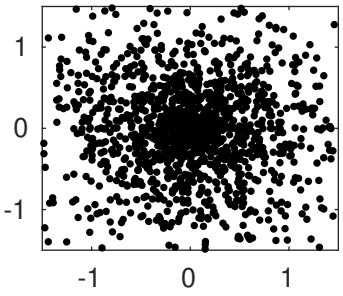
in the absence of jamming signal, and Fig. 7.17(c) and (d) show the received power spectrum for ZigBee and jamming signals at the ZigBee receiver. It is easy to see that the jamming signal is stronger than the ZigBee signal. According to (7.5), the measured JSR is 9.6 dB in this case.

We then process the received ZigBee and jamming signals using our proposed scheme in Fig. 7.6. Fig. 7.18(a) shows the constellation of the ZigBee signal (without despreading) decoded by our proposed ZigBee receiver. We can see that it can successfully decode the ZigBee packet in the presence of jamming attack. EVM of the decoded ZigBee signal is -20.4 dB. This means that the jamming mitigation capability of our design is 30.0 dB.

In contrast, Fig. 7.18(a) shows the constellation of the ZigBee signal decoded by conventional ZigBee receiver. It is evident that a conventional ZigBee receiver fails to decode its desired data packet in the presence of jamming attack.



(a) Signal decoded by the proposed ZigBee receiver.



(b) Signal decoded by the conventional ZigBee receiver.

Figure 7.18: Our proposed ZigBee receiver versus the conventional ZigBee receiver.

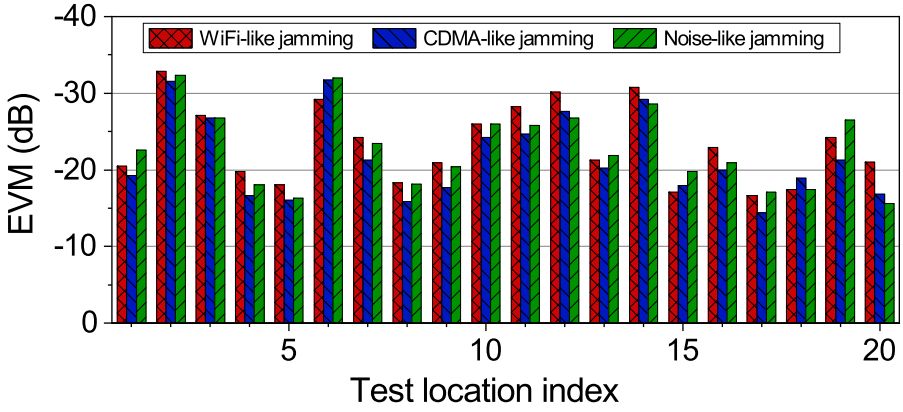


Figure 7.19: Measured EVM of our proposed ZigBee receiver.

7.6.4 Extensive Results

We measure the performance of the proposed ZigBee receiver and the conventional ZigBee receiver at other 19 locations in the same building as shown in Fig. 7.16. We consider three jamming waveforms: WiFi-like, CDMA-like, and noise-like signals. We report our measured experimental results as follows.

EVM. Fig. 7.19 shows the measured EVM of our proposed ZigBee receiver when it is placed at the designated locations to decode the ZigBee signals in the face of jamming signals. We can see that for all the locations, the achieved EVM of our proposed ZigBee receiver ranges from -32.9 dB to -14.4 dB, with an average of -22.6 dB. As a comparison, we place a conventional ZigBee

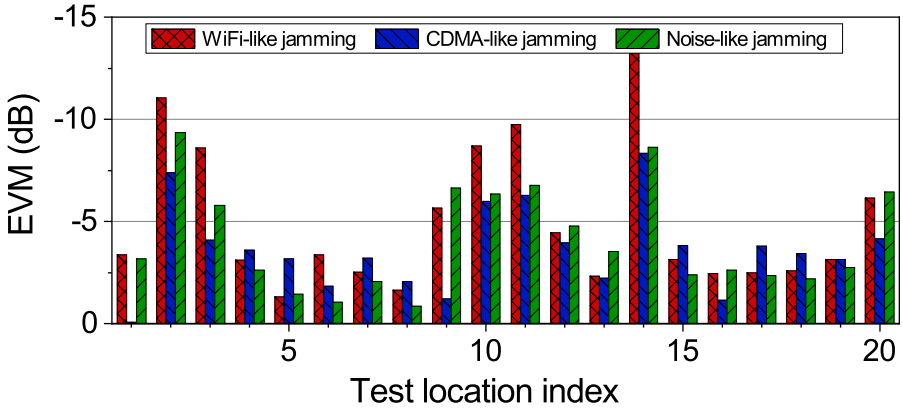


Figure 7.20: Measured EVM of conventional ZigBee receiver.

receiver at the same locations to decode the ZigBee signals in the presence of jamming signals. Fig. 7.20 shows our measured results. It is evident that our proposed ZigBee receiver significantly outperforms conventional ZigBee receiver, with an average EVM gain of 18.6 dB.

PRR. Based on the measured EVM, we calculate the average PRR at each location. Table 7.2 shows our results. We can see that the proposed ZigBee receiver achieves 100% PRR for all 20 locations, while the conventional receiver achieves 26.8% PRR on average.

JSR. To scrutinize the performance of our proposed ZigBee receiver, we measure its JSR according to (7.5). Fig. 7.21 shows our measured results. We can see that the measured JSR covers a highly dynamic range from -6 dB to 25.9 dB. Particularly, at location 5, the JSR is 25.9 dB, and its EVM is -16.0 dB. It means that the receiver achieves 41.9 dB JMC.

JMC. Based on the measured JSR and EVM, we calculate the JMC achieved by our proposed ZigBee receiver. Fig. 7.22 shows our results. The achieved JMC ranges from 21.0 dB to 41.9 dB, and the average of JMC at all the 20 locations is 26.7 dB.

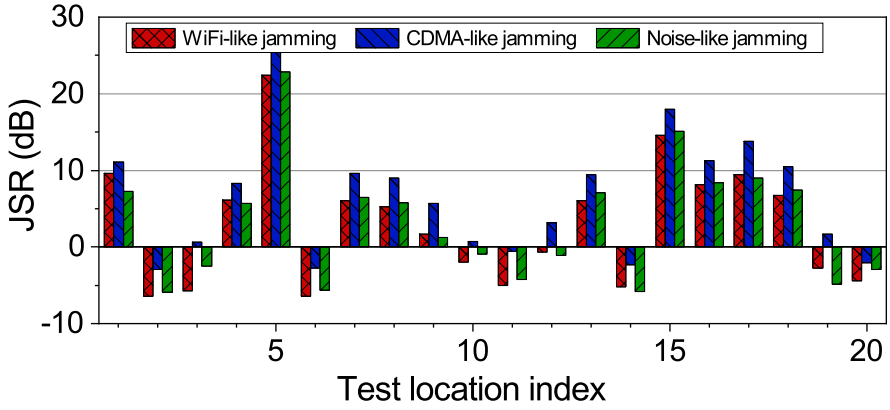


Figure 7.21: Measured JSR at our proposed ZigBee receiver.

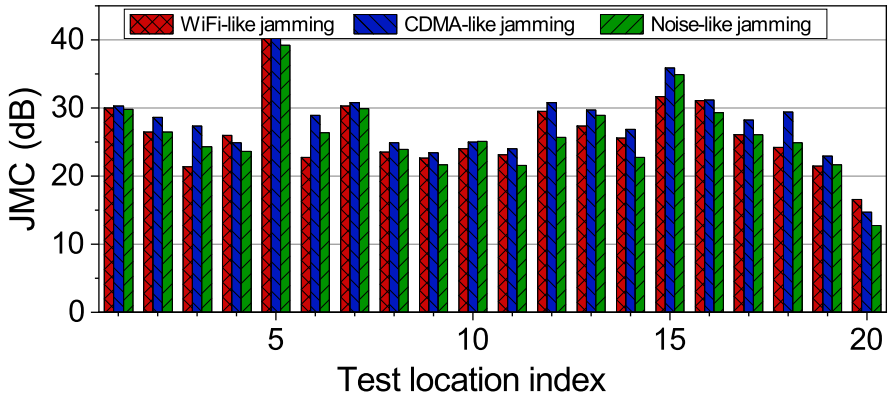


Figure 7.22: The JMC of the proposed ZigBee receiver in comparison with an off-the-shelf ZigBee receiver.

7.7 Chapter Summary

In this chapter, we introduced a jamming-resistant ZigBee receiver by leveraging recent advances in multi-antenna technology. The new ZigBee receiver is capable of decoding ZigBee packets in the presence of jamming signals. The key components of our design are a signal-projection-based sync module and a neural-network-based jamming mitigation module. To speed up its training process, we optimized the neural network by taking advantage of the inherent relations of its weights and used the preamble (256 chips) in each individual ZigBee packet for its training. We have built a prototype of our proposed ZigBee receiver and evaluated its performance in real-world wireless

environments. Experimental results show that our design can salvage ZigBee communications in the presence of jamming signals 20 dB stronger than ZigBee signals and that our design can offer 26.7 dB jamming mitigation capability on average.

Chapter 8

Summary and Outlook

8.1 Summary

In this thesis, we studied three sets of primary constraints in developing wireless IoT communications and networks; energy efficiency, spectral efficiency, and PHY-layer security. We introduced multiple solutions to enable efficient and ubiquitous wireless connectivity for IoT devices taking the constraints associated with IoT applications into account. We implemented the proposed schemes on real-world wireless testbeds and demonstrated their feasibility in practice. We showed that the proposed schemes are of higher capability compared to the state-of-the-art technologies and can be integrated into the existing wireless infrastructure. Specifically, this thesis introduced the following solutions.

- **Energy-Efficient IoT Communications for WLANs.** We introduced EE-IoT, an energy-efficient IoT communications scheme for WLANs that allows an OFDM-based AP to communicate with multiple QAM-based (non-OFDM) IoT devices at a very low sampling rate. The key enabler of EE-IoT is an asymmetric PHY design, in which the AP preserves its legacy hardware architecture to transmit/receive OFDM-modulated broadband signals, but each IoT device receives/transmits narrowband signal on a single subcarrier by tuning its carrier frequency to that subcarrier. By doing so, the IoT devices can use a much lower sampling rate (250 ksps) for signal transmission/reception and do not require computation-intensive

FFT/IFFT operations in their baseband signal processing, thereby leading to a significant reduction in their hardware complexity and power consumption. We built a prototype of EE IoT on USRP2 wireless testbed and evaluated its performance in an office building environment. Experimental results show that an AP can serve 24 IoT devices simultaneously and each IoT device can achieve more than 187 kbps in the downlink and more than 125 kbps in the uplink.

- **Coexistence of EE-IoT and Wi-Fi Devices.** We further developed EE-IoT scheme and enabled a transparent coexistence of Wi-Fi and IoT devices. We propose an SDMA-based approach that allows a multi-antenna AP to serve broadband Wi-Fi devices and narrowband IoT devices simultaneously. The key component of this scheme is a lightweight interference cancellation method, which can effectively mitigate mutual interference in practice by leveraging the spatial degrees of freedom provided by AP's multiple antennas. We built a prototype of WiFi-IoT on a GNURadio-USRP2 wireless testbed and evaluated its performance in an office building environment. Experimental results show that, using WiFi-IoT, an AP with two antennas can serve one Wi-Fi device and 24 IoT devices simultaneously in both uplink and downlink, with each IoT device achieving more than 375 kbps.

- **Uplink Distributed MIMO.** We presented UD-MIMO, a practical uplink-distributed MIMO scheme for WLANs. UD-MIMO improves the spectrum efficiency of WLANs by enabling concurrent data transmissions from multiple STAs to multiple APs. UD-MIMO is compatible with commercial off-the-shelf 802.11 devices (with modified driver). The enabler behind UD-MIMO is a new signal detection method that can decode concurrent data packets from asynchronous STAs. We have built a prototype of UD-MIMO on two wireless testbeds and demonstrated its compatibility with Qualcomm Atheros 802.11 devices. Our

experimental results show that UD-MIMO offers $3.4\times$ throughput compared to the CSMA-based interference-avoidance approach. Our experimental results also show that UD-MIMO achieves 82% throughput of MU-MIMO.

- **MU-MIMO Transmission for LoRa.** To improve the spectrum efficiency of LoRaWANs, we introduced MaLoRaGW that enables MU-MIMO transmission for LoRa by enabling it to concurrently serve multiple LoRa user devices in both uplink and downlink. The key component of MaLoRaGW is a joint baseband signal design for uplink packet detection and downlink beamforming, which are underpinned by three modules: spatial signal projection, accurate channel estimation, and implicit beamforming. We have evaluated a two-antenna MaLoRaGW in realistic scenarios of different scales. It has been validated that MaLoRaGW is backward compatible with COTS LoRa devices. It further demonstrates 10% throughput gain in uplink and 95% throughput gain in downlink when compared to the state-of-the-art.

- **Jamming-Resistant Communications for VANETs.** We presented JammingBird, a MIMO-based receiver structure, to secure VANETs against constant jamming attacks. The key enabler of JammingBird is a jamming suppressor module that uses the channel ratio between the jammer and the two receive antennas to eliminate jamming signal and recover the desired packet. JammingBird leverages unused time-frequency resources within the 802.11p frames to estimate the jammer's CSI ratio. The proposed receiver is capable of decoding desired packets under high-power constant jamming attacks, regardless of the PHY-layer technology employed by the jammer. We built JammingBird on a USRP-based wireless testbed and evaluated its performance in real-world outdoor scenarios. Our experimental results demonstrated that JammingBird can decode the desired packets in the face of 25 dB stronger jamming signal.

- **Jamming-Resilient Receiver Design for ZigBee Communications.** We presented a learning-based receiver to secure ZigBee communications against high-power in-band constant jamming attacks. We designed a lightweight neural network that serves as a linear spatial filter to suppress constant jamming attacks while not requiring any knowledge of the jamming signal. We built a prototype of ZigBee receiver on a wireless testbed to validate our design in real-world wireless environments and evaluated its performance in the presence of a malicious device that emits different types of radio jamming signals. The proposed ZigBee receiver offers an average of 26.7 dB jamming mitigation capability compared to off-the-shelf ZigBee receivers.

8.2 Lessons Learned

Despite the advancement of wireless technologies and the research efforts on the design of efficient and secure wireless connectivity for IoT devices, the performance of real-world IoT network systems is still far from satisfaction in practice. For example, commercial IoT technologies (e.g., Wi-Fi and ZigBee) can still be easily disrupted by malicious jamming emitters. Yet, packet collision (i.e., co-channel interference) limits the network throughput and spectrum efficiency of state-of-the-art technologies. This reality motivates us to rethink the missing parts in this field. This thesis details contributions that span both PHY/MAC-layer algorithm and protocol design and system implementation in developing wireless IoT communications and networks. This design approach led to important research contributions and lessons learned.

- **Systematic Design.** A wireless transceiver has several key components of both analog and digital circuits for transmission/reception, including power amplifiers, mixers, filters, ADC/DAC, digital modulation, packet detection, frequency synchronization, timing recov-

ery, and signal detection. In the literature, the majority of work focuses solely on one component, whereas the impacts and performance of other components on their design are highly overlooked. For example, in the analog domain, when the co-channel interference or jamming signal is strong, it will saturate the dynamic range of ADC, resulting in a significant quantization noise for the useful signal in the digital domain. In the digital domain, the frequency and timing offsets of a packet must be accurately estimated and compensated in order to decode this packet. However, these digital components are vulnerable to co-channel interference and jamming signals. It means that enabling efficient and secure wireless communications cannot be limited to a single component; rather, it needs a holistic and systematic design to mitigate jamming signal or concurrently decode collided packets received from multiple STAs. In this thesis, we showed how system implementation helped us learn the performance of different components in our design. This level of understanding allowed us to identify the performance bottleneck in our system and redesign algorithms and protocols to address that.

- **Computational Complexity.** Sophisticated digital signal processing schemes, albeit offering superior performance, cannot always be suited for real-world wireless transceivers. This is because the digital signal processing modules of a radio transceiver is typically implemented on application-specific integrated circuit (ASIC), making computational complexity a key factor to be considered for massive production. In this thesis, we showed how non-OFDM IoT devices can communicate with an OFDM-based WLAN AP using an asymmetric PHY design. In the proposed scheme, IoT devices will no longer require computation-intensive FFT/IFFT operations in their baseband signal processing. This will significantly reduce IoT devices' hardware complexity and power consumption.

8.3 Future Work

- **MU-MIMO for Passive RFID Systems.** Passive ultra high frequency (UHF) radio frequency identification (RFID) tags have been used in many sectors of our society, such as warehouses, libraries, retail stores, supply chains, and transportation. In real applications, RFID readers with a fast tag reading rate are always desirable. Such readers will directly enhance the efficiency of tag-intensive RFID systems. However, existing RFID readers have a very limited reading rate. For instance, Impinj Speedway RFID readers can read at most 35 tags/s in a read zone [142], which is far from enough for many applications. This limit stems from the PHY and MAC layers of RFID communication protocols. The coordination of tags' transmission and the management of tag collision consume a large portion of a reader's airtime, leading to a significantly reduced tag reading rate in real scenarios. One possible solution to increase the tag reading rate by multiple times is by enabling concurrent multi-tag reading via MU-MIMO technique. However, MU-MIMO application and implementation in passive RFID systems face new challenges in CSI estimation and beamforming design. This is because RFID tags have very limited communication and computation power; they are incapable of estimating the downlink CSI and reporting it to the reader. To address this challenge, we can employ implicit beamforming for downlink transmission by leveraging OTA channel reciprocity and design a collision recovery scheme to decode the collided tag's packets and estimate the uplink channels.

- **Cross-Domain Anti-Jamming Design.** Most existing anti-jamming techniques exploit the degree of freedom in a single (time, frequency, space, code, etc.) domain to decode in-the-air radio packets in the presence of interfering signals from malicious jammers. For instance, channel hopping, which is used in Bluetooth, manipulates radio signals in the frequency

domain to avoid jamming attack; spectrum spreading employs a secret sequence in the code domain to whiten the energy of narrowband jamming signal to enhance a wireless receiver's resilience to jamming attacks; MIMO-based jamming mitigation aims to project signals in the spatial domain so as to make useful signal perpendicular to jamming signals. However, these single-domain anti-jamming techniques appear to have a limited ability of handling jamming signals due to a number of factors, such as the available spectrum bandwidth, the computational complexity, the number of antennas, the resolution of ADC, the nonlinearity of radio circuit, and the packet delay constraint. One research direction toward enhancing a wireless network's resilience to jamming attacks is by jointly exploiting multiple domains for PHY-layer signal processing and MAC-layer protocol manipulation. This direction deserves more research efforts to explore practical and efficient anti-jamming designs.

BIBLIOGRAPHY

[1] Chin-Lung Hsu and Judy Chuan-Chuan Lin. An empirical examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives. *Computers in Human Behavior*, 62:516–527, 2016.

[2] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of Things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[3] Biljana L Risteska Stojkoska and Kire V Trivodaliev. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454–1464, 2017.

[4] Joel JPC Rodrigues, Dante Borges De Rezende Segundo, Heres Arantes Junqueira, Murilo Henrique Sabino, Rafael Maciel Prince, Jalal Al-Muhtadi, and Victor Hugo C De Albuquerque. Enabling technologies for the Internet of health things. *IEEE Access*, 6:13129–13141, 2018.

[5] Li Da Xu, Wu He, and Shancang Li. Internet of Things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[6] Juan Antonio Guerrero-Ibanez, Sherali Zeadally, and Juan Contreras-Castillo. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies. *IEEE Wireless Communications*, 22(6):122–128, 2015.

[7] Mathy Vanhoef and Frank Piessens. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of Computer Security Applications Conference*, pages 256–265, 2014.

[8] Matt Ettus. Universal software radio peripheral (USRP). *Ettus Research LLC http://www. ettus. com*, 2008.

[9] WARP: Wireless Open Access Research Platform. Warp v3, 2020. Available at: `https://warpproject.org/trac/wiki/HardwareUsersGuides/WARPv3` [Online; accessed 2020-09-17].

[10] Hossein Pirayesh and Huacheng Zeng. Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 24(2):767–809, 2022.

[11] Hossein Pirayesh, Pedram Kheirkhah Sangdeh, and Huacheng Zeng. EE-IoT: An energy-efficient IoT communication scheme for WLANs. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, pages 361–369, 2019.

[12] Hossein Pirayesh, Pedram Kheirkhah Sangdeh, and Huacheng Zeng. Coexistence of Wi-Fi and IoT communications in WLANs. *IEEE Internet of Things Journal*, 7(8):7495–7505, 2020.

[13] Hossein Pirayesh, Pedram Kheirkhah Sangdeh, Qiben Yan, and Huacheng Zeng. UD-MIMO: Uplink distributed MIMO for wireless LANs. In *Proc. of IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2021.

[14] Hossein Pirayesh, Shichen Zhang, Pedram Kheirkhah Sangdeh, and Huacheng Zeng. MaLoRaGW: Multi-user MIMO transmission for LoRa. In *Proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, page 179–192, 2023.

[15] Hossein Pirayesh, Pedram Kheirkhah Sangdeh, Shichen Zhang, Qiben Yan, and Huacheng Zeng. JammingBird: Jamming-resilient communications for vehicular ad hoc networks. In *Proc. of IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2021.

[16] Hossein Pirayesh, Pedram Kheirkhah Sangdeh, and Huacheng Zeng. Securing ZigBee communications against constant jamming attack using neural network. *IEEE Internet of Things Journal*, 8(6):4957–4968, 2020.

[17] 3GPP TR 45.820. Cellular system support for ultra low complexity and low throughput Internet of Things. V2.1.0, August 2015.

[18] Koushik Maharatna, Eckhard Grass, and Ulrich Jagdhold. A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM. *IEEE Journal of Solid-State Circuit*, 39(3):484–493, 2004.

[19] JC Jensen, R Sadhwani, AA Kidwai, B Jann, A Oster, M Sharkansky, I Ben-Bassat, O Degani, S Porat, A Fridman, et al. Single-chip WiFi b/g/n $1\times$ 2 SoC with fully integrated front-end & PMU in 90nm digital CMOS technology. In *2010 IEEE Radio Frequency Integrated Circuits Symposium*, pages 447–450, 2010.

[20] Texas Instruments. Analog-to-digital converters (ADCs)-products. *www.ti.com/data-converters/adc-circuit/products.html [Online; Accessed: 2019-04-12]*, 2019.

[21] Rapeepat Ratasuk, Nitin Mangalvedhe, Yanji Zhang, Michel Robert, and Jussi-Pekka Koskinen. Overview of narrowband IoT in LTE Rel-13. In *IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–7, 2016.

[22] Andreas Hoglund, Xingqin Lin, Olof Liberg, Ali Behravan, Emre A Yavuz, Martin Van Der Zee, Yutao Sui, Tuomas Tirronen, Antti Ratilainen, and David Eriksson. Overview of 3GPP release 14 enhanced NB-IoT. *IEEE Network*, 31(6):16–22, 2017.

[23] Sung-Min Oh and JaeSheung Shin. An efficient small data transmission scheme in the 3GPP NB-IoT system. *IEEE Communications Letters*, 21(3):660–663, 2017.

[24] Leif R Wilhelmsson, Miguel M Lopez, and Dennis Sundman. NB-WiFi: IEEE 802.11 and Bluetooth low energy combined for efficient support of IoT. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2017.

[25] Naveed Butt, Rocco Di Taranto, Dennis Sundman, and Leif Wilhelmsson. On the feasibility to overlay a narrowband IoT signal in IEEE 802.11. In *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, 2017.

[26] Yu Wang, Luis Felipe Del Carpio, Dennis Sundman, Divya Peddireddy, and Anna Larmo. MAC layer design and evaluation of a narrowband Wi-Fi system. In *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2017.

[27] Zhijun Li and Tian He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14, 2017.

[28] Song Min Kim and Tian He. Freebee: Cross-technology communication via free side-channel. In *Proceedings of ACM MobiCom*, pages 317–330, 2015.

[29] Theodore S Rappaport. *Wireless communications: Principles and practice*, volume 2. Prentice Hall PTR New Jersey, 1996.

[30] J-J Van de Beek, Per Ola Borjesson, M-L Boucheret, Daniel Landstrom, Julia Martinez Arenas, Per Odling, Christer Ostberg, Mattias Wahlqvist, and Sarah Kate Wilson. A time and frequency synchronization scheme for multiuser OFDM. *IEEE Journal on Selected Areas in Communications*, 17(11):1900–1914, 1999.

[31] Michele Morelli, C-C Jay Kuo, and Man-On Pun. Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review. *Proceedings of the IEEE*, 95(7):1394–1427, 2007.

[32] IEEE 802.11ac. IEEE standard for information technology local and metropolitan area networks part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput. *IEEE Standards 802.11ac*, 2014.

[33] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Huacheng Zeng, and Hongxiang Li. A practical underlay spectrum sharing scheme for cognitive radio networks. In *IEEE INFOCOM*, pages 2521–2529. IEEE, 2019.

[34] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Adnan Quadri, and Huacheng Zeng. A practical spectrum sharing scheme for cognitive radio networks: Design and experiments. *IEEE/ACM Transactions on Networking*, 28(4):1818–1831, 2020.

[35] Huacheng Zeng, Chen Cao, Hongxiang Li, and Qiben Yan. Enabling jamming-resistant communications in wireless MIMO networks. In *Prof. of IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2017.

[36] Qiben Yan, Huacheng Zeng, Tingting Jiang, Ming Li, Wenjing Lou, and Y Thomas Hou. Jamming resilient communication using MIMO interference cancellation. *IEEE Transactions on Information Forensics and Security*, 11(7):1486–1499, Jul 2016.

[37] Qiben Yan, Huacheng Zeng, Tingting Jiang, Ming Li, Wenjing Lou, and Y Thomas Hou. MIMO-based jamming resilient communication in wireless networks. In *IEEE INFOCOM*, pages 2697–2706, April 2014.

[38] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. Argos: Practical many-antenna base stations. In *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 53–64, 2012.

[39] Ettus Research. USRP N210. *www.ettus.com/product/details/UN210-KIT [Online; accessed 30-Jul-2019]*, 2007.

[40] A free and open-source toolkit for software radio. *http://gnuradio.org [Online; Accessed 23-May-2019]*.

[41] Cisco Visual Networking Index. Global mobile data traffic forecast update, 2016–2021 white paper. *Cisco, San Jose, CA, USA*, 2017.

[42] Emna Charfi, Lamia Chaari, and Lotfi Kamoun. PHY/MAC enhancements and QoS mechanisms for very high throughput WLANs: A survey. *IEEE Communications Surveys & Tutorials*, 15(4):1714–1735, 2013.

[43] Junmei Yao, Jun Xu, Sheng Luo, Lu Wang, Chao Yang, Kaishun Wu, and Wei Lou. Comprehensive study on MIMO-related interference management in WLANs. *IEEE Communications Surveys & Tutorials*, 21(3):2087–2110, 2019.

[44] Hariharan Shankar Rahul, Swarun Kumar, and Dina Katabi. JMB: Scaling wireless capacity with user demands. In *Proc. of ACM SIGCOMM*, pages 235–246, 2012.

[45] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. AirSync: Enabling distributed multiuser MIMO with full spatial multiplexing. *IEEE/ACM Transactions on Networking (ToN)*, 21(6):1681–1695, 2013.

[46] Ezzeldin Hamed, Hariharan Rahul, and Bahar Partov. Chorus: Truly distributed distributed-MIMO. In *Proc. of ACM SIGCOMM*, pages 461–475, 2018.

[47] Xinyu Zhang, Karthikeyan Sundaresan, Mohammad A Amir Khojastepour, Sampath Rangarajan, and Kang G Shin. NEMOx: Scalable network MIMO for wireless networks. In

Proceedings of the 19th annual international conference on Mobile computing & networking, pages 453–464, 2013.

[48] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.

[49] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. Sourcesync: A distributed wireless architecture for exploiting sender diversity. *ACM SIGCOMM Computer Communication Review*, 41(4):171–182, 2011.

[50] Maria Antonieta Alvarez and Umberto Spagnolini. Distributed time and carrier frequency synchronization for dense wireless networks. *IEEE Transactions on Signal and Information Processing over Networks*, 4(4):683–696, 2018.

[51] Yao-Win Hong and Anna Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099, 2005.

[52] Wen-Qin Wang. Carrier frequency synchronization in distributed wireless sensor networks. *IEEE Systems Journal*, 9(3):703–713, 2014.

[53] Ali A Nasir, Hani Mehrpouyan, Steven D Blostein, Salman Durrani, and Rodney A Kennedy. Timing and carrier synchronization with channel estimation in multi-relay cooperative networks. *IEEE Transactions on Signal Processing*, 60(2):793–811, 2011.

[54] Malik Muhammad Usman Gul, Xiaoli Ma, and Sungeun Lee. Timing and frequency synchronization for OFDM downlink transmissions using Zadoff-Chu sequences. *IEEE Transactions on Wireless Communications*, 14(3):1716–1729, 2014.

[55] Anfu Zhou, Teng Wei, Xinyu Zhang, Min Liu, and Zhongcheng Li. Signpost: Scalable MU-MIMO signaling with zero CSI feedback. In *Proc. of ACM MobiHoc*, pages 327–336, 2015.

[56] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. Achieving high data rates in a distributed MIMO system. In *Proc. of ACM MobiCom*, pages 41–52, 2012.

[57] Aneeq Mahmood, Reinhard Exel, and Thomas Bigler. On clock synchronization over wireless LAN using timing advertisement mechanism and TSF timers. In *Proc. of IEEE ISPCS*, pages 42–46, 2014.

[58] Qualcomm Atheros. AR9271 Highly integrated single-chip USB with 802.11n support. *www.ath-drivers.eu/qualcomm-atheros-datasheets-for-AR9271.html [Online; accessed 28-Jul-2019]*.

[59] Qualcomm Atheros. open-ath9k-htc-firmware. *https://github.com/vanhoefm/modwifi-ath9k-htc [Online; accessed 28-Jul-2019]*.

[60] Ettus Research. Octoclock CDA. *https://www.ettus.com/all-products/octoclock-g/ [Online; accessed 28-Jul-2019]*.

[61] Jothi Prasanna Shanmuga Sundaram, Wan Du, and Zhiwei Zhao. A survey on LoRa networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials*, 22(1):371–388, 2019.

[62] Maria Stoyanova, Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Pallis, and Evangelos K Markakis. A survey on the Internet of Things (IoT) forensics: challenges, approaches, and open issues. *IEEE Communications Surveys & Tutorials*, 22(2):1191–1221, 2020.

[63] Yinghui Li, Jing Yang, and Jiliang Wang. DyLoRa: Towards energy efficient dynamic LoRa transmission control. In *IEEE Conference on Computer Communications*, pages 2312–2320, 2020.

[64] Sezana Fahmida, Venkata P Modekurthy, Mahbubur Rahman, Abusayeed Saifullah, and Marco Brocanelli. Long-lived LoRa: Prolonging the lifetime of a LoRa network. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2020.

[65] Andrew J Wixted, Peter Kinnaird, Hadi Larijani, Alan Tait, Ali Ahmadinia, and Niall Strachan. Evaluation of LoRa and LoRaWAN for wireless sensor networks. In *2016 IEEE SENSORS*, pages 1–3. IEEE, 2016.

[66] Fusang Zhang, Zhaoxin Chang, Kai Niu, Jie Xiong, Beihong Jin, Qin Lv, and Daqing Zhang. Exploring LoRa for long-range through-wall sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–27, 2020.

[67] Shuai Tong, Zilin Shen, Yunhao Liu, and Jiliang Wang. Combating link dynamics for reliable LoRa connection in urban settings. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 642–655, 2021.

[68] Yuguang Yao, Zijun Ma, and Zhichao Cao. Losee: Long-range shared bike communication system based on lorawan protocol. In *EWSN*, pages 407–412, 2019.

[69] Chenning Li and Zhichao Cao. LoRa networking techniques for large-scale and long-term IoT: A down-to-top survey. *ACM Computing Surveys (CSUR)*, 55(3):1–36, 2022.

[70] Konstantin Mikhaylov, Juha Petäjäjärvi, and Janne Janhunen. On LoRaWAN scalability: Empirical evaluation of susceptibility to inter-network interference. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–6. IEEE, 2017.

[71] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Jamming of LoRa PHY and countermeasure. In *IEEE Conference on Computer Communications*, pages 1–10, 2021.

[72] Xiong Wang, Linghe Kong, Zucheng Wu, Long Cheng, Chenren Xu, and Guihai Chen. SLoRa: towards secure LoRa communications with fine-grained physical layer features. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 258–270, 2020.

[73] Paul Marcelis, Nikolaos Kouvelas, Vijay S Rao, and Venkatesha Prasad. DaRe: Data recovery through application layer coding for LoRaWAN. *IEEE Transactions on Mobile Computing*, 2020.

[74] Qian Chen and Jiliang Wang. Alignтrack: Push the limit of LoRa collision decoding. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2021.

[75] Xianjin Xia, Yuanqing Zheng, and Tao Gu. FTrack: Parallel decoding for LoRa transmissions. *IEEE/ACM Transactions on Networking*, 28(6):2573–2586, 2020.

[76] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. CoLoRa: Enabling multi-packet reception in LoRa. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 2303–2311, 2020.

[77] Zhe Wang, Linghe Kong, Kangjie Xu, Liang He, Kaishun Wu, and Guihai Chen. Online concurrent transmissions at LoRa gateway. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 2331–2340, 2020.

[78] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. Concurrent interference cancellation: decoding multi-packet collisions in LoRa. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 503–515, 2021.

[79] Bin Hu, Zhimeng Yin, Shuai Wang, Zhuqing Xu, and Tian He. SCLoRa: leveraging multi-dimensionality in decoding collided LoRa transmissions. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2020.

[80] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. Pyramid: Real-time LoRa collision decoding with peak tracking. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2021.

[81] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321, 2017.

[82] Shuai Tong, Jiliang Wang, and Yunhao Liu. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 234–246, 2020.

[83] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. MLoRa: A multi-packet reception protocol in LoRa networks. In *Proceedings of IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–11, 2019.

[84] Qing Yang, Xiaoxiao Li, Hongyi Yao, Ji Fang, Kun Tan, Wenjun Hu, Jiansong Zhang, and Yongguang Zhang. BigStation: Enabling scalable real-time signal processingin large MU-MIMO systems. *ACM SIGCOMM Computer Communication Review*, 43(4):399–410, 2013.

[85] Zhe Chen, Xu Zhang, Sulei Wang, Yuedong Xu, Jie Xiong, and Xin Wang. BUSH: Empowering large-scale MU-MIMO in WLANs with hybrid beamforming. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[86] Yunze Zeng, Ioannis Pefkianakis, Kyu-Han Kim, and Prasant Mohapatra. MU-MIMO-aware AP selection for 802.11 ac networks. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '17, page 19, New York, NY, USA, 2017. ACM, ACM.

[87] Huacheng Zeng, Hongxiang Li, and Qiben Yan. Uplink MU-MIMO in asynchronous wireless LANs. In *Proc. of ACM MobiHoc*, pages 21–30, 2018.

[88] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. PCube: scaling LoRa concurrent transmissions with reception diversities. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 670–683, 2021.

[89] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, and Kyle Jamieson. CurvingLoRa to boost LoRa network capacity via concurrent transmission. *arXiv preprint arXiv:2201.05179*, 2022.

[90] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. NELoRa: Towards ultra-low SNR LoRa communication with neural-enhanced demodulation. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 56–68, 2021.

[91] Fusang Zhang, Zhaoxin Chang, Jie Xiong, and Daqing Zhang. Exploring LoRa for sensing. *GetMobile: Mobile Computing and Communications*, 25(2):33–37, 2021.

[92] Binbin Xie, Yuqing Yin, and Jie Xiong. Pushing the limits of long range wireless sensing with LoRa. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–21, 2021.

[93] Fusang Zhang, Zhaoxin Chang, Jie Xiong, Rong Zheng, Junqi Ma, Kai Niu, Beihong Jin, and Daqing Zhang. Unlocking the beamforming potential of LoRa for long-range multi-target respiration sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–25, 2021.

[94] Binbin Xie and Jie Xiong. Combating interference for long range LoRa sensing. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 69–81, 2020.

[95] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Kai Chen, Dianhe Han, Dingyi Fang, Zhanyong Tang, and Zheng Wang. WideSee: Towards wide-area contactless wireless sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 258–270, 2019.

[96] LoRa Alliance Technical Committee et al. LoRaWAN 1.1 Specification. *LoRa Alliance, Standard*, 1(1), 2017.

[97] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. LMAC: Efficient carrier-sense multiple access for LoRa. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

[98] Cheng-Xiang Wang, Ji Bian, Jian Sun, Wensheng Zhang, and Minggao Zhang. A survey of 5G channel measurements and models. *IEEE Communications Surveys & Tutorials*, 20(4):3142–3168, 2018.

[99] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A tutorial on IEEE 802.11ax high efficiency WLANs. *IEEE Communications Surveys & Tutorials*, 21(1):197–216, 2018.

[100] Pedram Kheirkhah Sangdeh and Huacheng Zeng. DeepMux: Deep-learning-based channel sounding and resource allocation for IEEE 802.11ax. *IEEE Journal on Selected Areas in Communications*, 39(8):2333–2346, 2021.

[101] Eldad Perahia and Robert Stacey. *Next generation wireless LANs: 802.11n and 802.11ac*. Cambridge university press, 2013.

[102] Erik Dahlman, Stefan Parkvall, Johan Skold, and Per Beming. *3G evolution: HSPA and LTE for mobile broadband*. Academic press, 2010.

[103] Nicolas Sornin, Miguel Luis, Thomas Eirich, Thorsten Kramp, and Olivier Hersent. Lorawan specification. *LoRa alliance*, 2015.

[104] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Aryan Mobiny, and Huacheng Zeng. LB-SciFi: Online learning-based channel feedback for MU-MIMO in wireless LANs. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2020.

[105] World Health Organization et al. Association for safe international road travel. *Global Status Report on Road Safety*, 2018.

[106] Umar Farooq, Jennifer L Hardy, Lei Gao, and Muhammad Abrar Siddiqui. Economic impact/forecast model of intelligent transportation systems in Michigan: An input output analysis. *Journal of Intelligent Transportation Systems*, 12(2):86–95, 2008.

[107] Matthew Barth and Kanok Boriboonsomsin. Environmentally beneficial intelligent transportation systems. *IFAC Proceedings Volumes*, 42(15):342–345, 2009.

[108] Syed Adeel Ali Shah, Ejaz Ahmed, Muhammad Imran, and Sherali Zeadally. 5G for vehicular communications. *IEEE Communications Magazine*, 56(1):111–117, 2018.

[109] US FCC. In the matter of use of the 5.850-5.925 GHz band. *Notice of Proposed Rulemaking, ET Docket*, (19-138):19–129, 2019.

[110] Oscar Punal, Carlos Pereira, Ana Aguiar, and James Gross. Experimental characterization and modeling of RF jamming attacks on VANETs. *IEEE transactions on vehicular technology*, 64(2):524–540, 2014.

[111] Zeinab El-Rewini, Karthikeyan Sadatsharan, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. Cybersecurity challenges in vehicular communications. *Vehicular Communications*, 23:100214, 2020.

[112] Muhammad Arif, Guojun Wang, Md Zakirul Alam Bhuiyan, Tian Wang, and Jianer Chen. A survey on security attacks in VANETs: Communication, applications and challenges. *Vehicular Communications*, 19:100179, 2019.

[113] Oscar Puñal, Ana Aguiar, and James Gross. In VANETs we trust? characterizing RF jamming in vehicular networks. In *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*, pages 83–92, 2012.

[114] Xiaozhen Lu, Dongjin Xu, Liang Xiao, Lei Wang, and Weihua Zhuang. Anti-jamming communication game for UAV-aided VANETs. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2017.

[115] Liang Xiao, Xiaozhen Lu, Dongjin Xu, Yuliang Tang, Lei Wang, and Weihua Zhuang. UAV relay in VANETs against smart jamming with reinforcement learning. *IEEE Transactions on Vehicular Technology*, 67(5):4087–4097, 2018.

[116] Anh Tuan Nguyen, Lynda Mokdad, and Jalel Ben Othman. Solution of detecting jamming attacks in vehicle ad hoc networks. In *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*, pages 405–410, 2013.

[117] Lynda Mokdad, Jalel Ben-Othman, and Anh Tuan Nguyen. DJAVAN: Detecting jamming attacks in vehicle ad hoc networks. *Performance Evaluation*, 87:47–59, 2015.

[118] Abderrahim Benslimane and Huong Nguyen-Minh. Jamming attack model and detection method for beacons under multichannel operation in vehicular networks. *IEEE Transactions on Vehicular Technology*, 66(7):6475–6488, 2016.

[119] Nikita Lyamin, Denis Kleyko, Quentin Delooz, and Alexey Vinel. Real-time jamming DoS detection in safety-critical V2V C-ITS using data mining. *IEEE Communications Letters*, 23(3):442–445, 2019.

[120] Dimitrios Karagiannis and Antonios Argyriou. Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning. *Vehicular Communications*, 13:56–63, 2018.

[121] Sunil Kumar, Karan Singh, Sushil Kumar, Omprakash Kaiwartya, Yue Cao, and Huan Zhou. Delimitated anti jammer scheme for Internet of vehicle: Machine learning based security approach. *IEEE Access*, 7:113311–113323, 2019.

[122] Tan Tai Do, Emil Björnson, Erik G Larsson, and S Mohammad Razavizadeh. Jamming-resistant receivers for the massive MIMO uplink. *IEEE Transactions on Information Forensics and Security*, 13(1):210–223, 2018.

[123] Hossein Akhlaghpasand, Emil Björnson, and S Mohammad Razavizadeh. Jamming suppression in massive MIMO systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(1):182–186, 2019.

[124] IEEE 802 Working Group. IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (LR-WPANs). *IEEE Std*, 802:4–2011, 2011.

[125] Michael Spuhler, Domenico Giustiniano, Vincent Lenders, Matthias Wilhelm, and Jens B Schmitt. Detection of reactive jamming in DSSS-based wireless communications. *IEEE Transactions on Wireless Communications*, 13(3):1593–1603, 2014.

[126] J Rewienski, Mateusz Groth, L Kulas, and Krzysztof Nyka. Investigation of continuous wave jamming in an IEEE 802.15.4 network. In *Proceedings of International Microwave and Radar Conference (MIKON)*, pages 242–246, 2018.

[127] Matthias Wilhelm, Ivan Martinovic, Jens B Schmitt, and Vincent Lenders. Short paper: Reactive jamming in wireless networks: how realistic is the threat? In *Proceedings of the fourth ACM conference on Wireless network security*, pages 47–52, 2011.

[128] Panlong Yang, Yubo Yan, Xiang-Yang Li, Yafei Zhang, Yue Tao, and Lizhao You. Taming cross-technology interference for Wi-Fi and ZigBee coexistence networks. *IEEE Transactions on Mobile Computing*, 15(4):1009–1021, 2015.

[129] Yubo Yan, Panlong Yang, Xiang-Yang Li, Yafei Zhang, Jianjiang Lu, Lizhao You, Jiliang Wang, Jinsong Han, and Yan Xiong. Wizbee: Wise ZigBee coexistence via interference cancellation with single antenna. *IEEE Transactions on Mobile Computing*, 14(12):2590–2603, 2014.

[130] James Hou, Benjamin Chang, Dae-Ki Cho, and Mario Gerla. Minimizing 802.11 interference on ZigBee medical sensors. In *Proceedings of the Fourth International Conference on Body Area Networks*, pages 1–8, 2009.

[131] Konstantinos Pelechrinis, Ioannis Broustis, Srikanth V Krishnamurthy, and Christos Gkantsidis. A measurement-driven anti-jamming system for 802.11 networks. *IEEE/ACM Transactions on Networking*, 19(4):1208–1222, 2011.

[132] Farhan M Aziz, Jeff S Shamma, and Gordon L Stüber. Resilience of LTE networks against smart jamming attacks. In *IEEE Global Communications Conference*, pages 734–739. IEEE, 2014.

[133] Marc Lichtman, Roger Piqueras Jover, Mina Labib, Raghunandan Rao, Vuk Marojevic, and Jeffrey H Reed. LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation. *IEEE Communications Magazine*, 54(4):54–61, 2016.

[134] Tan Tai Do, Emil Björnson, Erik G Larsson, and S Mohammad Razavizadeh. Jamming-resistant receivers for the massive MIMO uplink. *IEEE Transactions on Information Forensics and Security*, 13(1):210–223, 2017.

[135] Simon Begashaw, Danh H Nguyen, and Kapil R Dandekar. Enhancing blind interference alignment with reinforcement learning. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2016.

[136] Andrea Toma, Tassadaq Nawaz, Yue Gao, Lucio Marcenaro, and Carlo S Regazzoni. Interference mitigation in wideband radios using spectrum correlation and neural network. *IET Communications*, 13(10):1336–1347, 2019.

[137] Ying He, Zheng Zhang, F Richard Yu, Nan Zhao, Hongxi Yin, Victor CM Leung, and Yanhua Zhang. Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks. *IEEE Transactions on Vehicular Technology*, 66(11):10433–10445, 2017.

[138] Lixin Li, Yang Xu, Zihe Zhang, Jiaying Yin, Wei Chen, and Zhu Han. A prediction-based charging policy and interference mitigation approach in the wireless powered Internet of Things. *IEEE Journal on Selected Areas in Communications*, 37(2):439–451, 2018.

[139] Shahin Farahani. *ZigBee wireless networks and transceivers*. Newnes, Amsterdam, The Netherlands, 2011.

[140] Eric Blossom. GNU Radio: tools for exploring the radio frequency spectrum. *Linux journal*, 2004(122):4, 2004.

[141] Thomas Schmid. GNU Radio 802.15. 4 en-and decoding. UCLA NESL, Los Angeles, CA, Tech. Rep., 2006.

[142] Impinj. Impinj reader specifications table, 2021. Available at: www.shorturl.at/bhzSV [accessed 2021-10-01].

APPENDIX

Based on our assumption of zero-noise, (7.1) can be written as $\mathbf{y} = \mathbf{h}_1 x(n) + \mathbf{h}_2 z(n)$. We assume that $\mathbb{E}[|x(n)|^2] = 1$ and $\mathbb{E}[|z(n)|^2] = 1$. The ZigBee and jamming signal power is expressed by their channels ($\mathbf{h}_1$ and $\mathbf{h}_2$). Let $\mathbf{p} = \mathbf{u}_1^{\mathsf{H}}$, where $\mathbf{u}_1$ is a column of $\mathbf{u}$ and $[\mathbf{u} \;\; \mathbf{d} \;\; \mathbf{v}] = svd\big(\sum_{n=1}^{N_{\text{sync}}} \mathbf{y}(n)\mathbf{y}(n)^{\mathsf{H}}\big)$. Without loss generality, we assume $|v_{11}| \geq |v_{21}|$ for $\mathbf{v} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$. At the ZigBee receiver, we use $\mathbf{p}$ as the projection filter for signal projection. The projected signal can be written as:

$$
\begin{aligned}
\mathbf{py} &= \mathbf{u}_1^{\mathsf{H}}\mathbf{udv}^{\mathsf{H}}[x(n) \;\; z(n)]^{\mathsf{T}} \\
&= \mathbf{u}_1^{\mathsf{H}}[\mathbf{u}_1 \;\; \mathbf{u}_2] \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^{\mathsf{H}} [x(n) \;\; z(n)]^{\mathsf{T}} \\
&= [1 \;\; 0] \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^{\mathsf{H}} [x(n) \;\; z(n)]^{\mathsf{T}} \\
&= [d_1 \;\; 0] \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^{\mathsf{H}} [x(n) \;\; z(n)]^{\mathsf{T}} \\
&= d_1[v_{11}^* \;\; v_{21}^*][x(n) \;\; z(n)]^{\mathsf{T}} \\
&= d_1[v_{11}^* x(n) + v_{21}^* z(n)]. \qquad (1)
\end{aligned}
$$

Based on (1), the SJR of the sigal after projection can be written as $\frac{|v_{11}^* x(n)|^2}{|v_{21}^* z(n)|^2}$. Given that $|v_{11}| \geq |v_{21}|$ and $\mathbb{E}[x(n)]$ and $\mathbb{E}[z(n)]$, we have the SJR of $\mathbf{py}$ is greater than or equal to 1. This completes the proof.