ON EFFICIENCY AND INTELLIGENCE OF NEXT-GENERATION WIRELESS NETWORKS

By

Pedram Kheirkhah Sangdeh

A DISSERTATION

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

Computer Science – Doctor of Philosophy

2023

ABSTRACT

The ever-increasing demand for data-hungry wireless services and rapid proliferation of wireless devices in sub-6 GHz band have pushed the current wireless technologies to a breaking point, necessitating efficient and intelligent strategies to utilize scarce communication resources. This thesis aims at leveraging novel communication frameworks, artificial intelligence techniques, and synergies between them in bringing efficiency and intelligence to the next generation of wireless networks.

In the first chapter of this thesis, we propose a novel spectrum sharing scheme to address spectrum shortage, a fundamental issue in current and future wireless networks. Our proposed scheme enables transparent spectrum utilization for a small cognitive radio network by leveraging two interference management techniques that are not reliant on inter-network coordination, fine-grained synchronization, and knowledge about other occupants of the spectrum. We further extend this idea in the second chapter of this thesis and enable concurrent device-to-device and cellular communications in cellular networks where the base station and wireless devices exploit interference management techniques to avoid causing interference to each other, making concurrent spectrum utilization possible for both cellular and device-to-device communications. In the third chapter, to enhance spectral efficiency, connectivity, and throughput of Wireless Local Area Networks (WLAN), we propose a non-orthogonal multiplexing scheme (NOMA). In our proposed scheme, the access point (AP) is equipped with a novel precoder design and user grouping which are tailored based on the requirements of power-domain NOMA. Also, a novel successive interference cancellation technique is designed for users which does not require channel estimation to decode the signals and is more resilient to interference compared to the existing techniques.

The second part of this thesis focuses on taking advantage of artificial intelligence for solving communication and networking challenges and also taking advantage of novel communication

frameworks to let future wireless networks indulge intelligence-oriented networking and resource management. In the fourth chapter, we propose a new solution to solve a long-standing issue ahead of multi-user multiple-input multiple-output (MU-MIMO) communications in WLANs, which is the large sounding overhead for acquiring the channel state information (CSI). Our learning-based solution includes an automated mechanism that enables access points to collect, clear, and balance dataset, and also deep neural networks to compress CSI and reduce the airtime overhead for channel acquisition. However, with provisioning concurrent MU-MIMO and orthogonal frequency division multiple access (OFDMA) in the new generation of WLANs, not only the sounding overhead problem becomes more acute, but it also marries with a complex resource allocation problem which makes designing a practical enabler of MU-MIMO-OFDMA transmissions necessary for WLANs. In the fifth chapter of this thesis, we propose DeepMux, which comprises a deep-learning-based channel sounding and a deep-learning-based resource allocation both of which reside in access points and impose no computational/communication burden on users, enabling efficient downlink MU-MIMO-OFDMA transmissions in WLANs. We finally design a communication framework for accelerating federated learning in future intelligent transportation systems, where heterogeneous capabilities and mobility of users along with limited available bandwidth for communications are huge obstacles toward making the network intelligent in a distributed manner. With the aid of a deadline-driven scheduler and asynchronous uplink multi-user MIMO, our proposed solution reduces data loss at vehicles in a dynamic vehicular environment, making a concrete step toward the practical adoption of federated learning in future transportation systems.

To my parents, Jafar and Manijeh, and to my wife, Fariba

ACKNOWLEDGMENTS

This dissertation would not be possible without the contribution and support of many people. Their efforts cannot be completely credited for with one page of a document. First and foremost, my deepest gratitude goes out to my advisor, Dr. Huacheng Zeng. I am thankful for all the advice and encouragement he gave me, for the hours of discussion we shared, and for all that I learned while working under his guidance. I am very grateful for the additional time he took to prepare me for the key moments of the Ph.D., from day one to writing up this dissertation. It is indeed a great privilege and honor to study under his supervision. I would like to acknowledge Prof. Matt Mutka and Dr. Mi Zhang. Their guidance and support have transformed this research work into a foundation of career, for which I am more than grateful. I am also in debt to Dr. Qiben Yan. His valuable feedback and countless advice were the keystones of two interesting research efforts I had the pleasure to work on.

I am grateful to my fellow colleagues Hossein Pirayesh, Adnan Quadri, and Shichen Zhang for their support of my lab work and countless brainstorming moments during the last five years. My special thanks go to my friends in Louisville, Mahyar, Mehdi, and Masoud, for their moral support, numerous nights out, flat parties, dinners, and holidays together. Thanks also to my life-long friend Shayan for the endless hours we spent around lakes and rivers without catching a single fish. I am glad our adventure continues in San Diego. Thanks to my friends from Tehran and Parehsar, Moein, Ashkan, Omid, Farnoush, Pegah, Behzad, Shahrokh, Saeid, Mojtaba, Kaveh, Mohammad, Majid, Ali, and Vahid, for keeping in touch all the years and growing even closer to my heart whilst I am miles away.

My sincere gratitude goes to my wife, Fariba, who stood by me during the ups and downs of Ph.D. life. Not only for the moral support but also for all the fruitful discussions about my work. What an adventure, I could not love you more. Thanks to my dog, Kenji, for his constant

interruptions during important meetings and interviews. His persistence in chewing my papers and supplies has been always inspiring. He taught me the ability to focus during presentations and transformed my reading style from in-print to on-screen. Finally, my most special thanks to my parents, Jafar and Manijeh, for their love and support, for all the comments they gave me on my work, for all their advice and encouragement, and for absolutely everything that they did.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

WLAN	Wireless Local Area Network
IoT	Internet of Things
AP	Access Point
BS	Base Station
CRN	Cognitive Radio Network
D2D	Device-to-Device
NOMA	Non-Orthogonal Multiple Acess
AI	Artificial Intelligence
FL	Federated Learning
BBF	Blind BeamForming
BIC	Blind Interference Cancellation
DoF	Degrees of Freedom
CSI	Channel State Information
SIC	Successive Interference Cancellation
OMA	Orthogonal Multiple Access
DL	Deep Learning
MU-MIMO	Multi-User Multiple-Input Multiple-Output
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
ITS	Intelligent Transportation Systems
DNN	Deep Neural Network
DLCS	DL-based Channel Sounding
DLRA	DL-based Resource Allocation
ML	Machine Learning
DDVS	Deadline-Driven Vehicle Scheduler
CVPS	Concurrent Vehicle Polling Scheme

IC	Interference Cancellation
TDD	Time-Division Duplex
CDMA	Code-Division Multiple Access
ZF	Zero Forcing
MMSE	Minimum Mean Square Error
SDR	Software-Defined Radio
SIR	Signal to Interference Ratio
EVM	Error Vector Magnitude
EBF	Explicit Beamforming
NDP	Null Data Packet
IBF	Implicit Beamforming
UEs	User Equipment
PRB	Physical Resource Block
NR	New Radio
TDMA	Time Division Multiple Access
SNR	Signal to Noise Ratio
MSE	Mean Square Error
SVD	Singular Value Decomposition
MCS	Modulation and Coding Scheme
FFT	Fast Fourier Transform
FDMA	Frequency Division Multiple Access
RAT	Radio Access Technologies
MISO	Multi-Input Single-Output
SISO	Single-Input Single-Output
MM	Minorization-Majorization
MIMO	Multi-Input Multi-Output
SINR	Signal-to-Interference-and-Noise Ratio

ISNR	Interference-to-Signal-and-Noise Ratio
NDPA	Null Data Packet Announcement
L-STF	Legacy Short Training Field
L-LTF	Legacy Long Training Field
L-SIG	Legacy Signal
DNN-AE	Deep Neural Network Auto-Encoder
GR	Givens Rotation
PSE	Power spectral Entropy
ReLU	Rectified Linear Unit
AFC	Adaptive Feedback Compression
RU	Resource Unit
MINLP	Mixed-Integer Non-Linear Programming
LP	Linear Programming
BR	Beamforming Report
TBRP	Trigger Beamforming Report Poll
MILP	Mixed-Integer Linear Programming
NMSE	Normalized Mean Squared Error
EPS	Extended Polynomial Scheduler
FPD	Fictitious Polynomial Deadline
FSC	Fictitious Scheduler Construction
CNN	Convolutional Neural Network
RND	Random Scheduler
RR	Round-Robin Scheduler
EDF	Earliest Deadline First Scheduler
SP	Sequential Polling

Chapter 1

Introduction

The burgeoning demand for data-hungry wireless services and the proliferation of mobile devices have pushed the current wireless technologies to a breaking point whereby the traditional ways of deploying, operating, managing, and troubleshooting wireless networks are not efficient anymore. Only Wireless Local Area Networks (WLANs) have connected more than 22.2 billion devices around the globe [155], let alone tens of billions Internet of Things (IoT) devices [123] and smartphones [35]. The scarcity of spectrum and over-congestion of sub-6GHz band necessitate the advent of more efficient and intelligent wireless networking and communication paradigms for the next generations of wireless networks.

Efficient utilization of limited available resources and preservation of optimal performance at scale are two indispensable requirements in next-generation wireless networks. However, these two requirements are not easy to fulfill. First, if set to employ shared communication resources, current wireless technologies compete for exclusive utilization and only consider their individual performance [2, 67, 69]. Second, the performance of existing networking solutions may not scale well or even may start setting back, as the number of users grows. In fact, excessive computational complexity, adverse effect of interference, large communication overhead, and limited power budget at Access Points (APs)/Base Stations (BSs) stations impede the scalability of networks.

In this thesis, we study how efficient communication frameworks, intelligent networking solutions, and synergies between them help to fulfill the two aforementioned requirements. In what

follows, we discuss our research scope in bringing efficiency and intelligence to wireless networks and present the overview of this thesis.

1.1 Research Scope and Contribution

Our research scope is broadly categorized into two parts, efficiency and intelligence, both exploring the opportunities for improving the performance of future wireless networks. As shown in Figure 1.1, the first part of the thesis particularly focuses on the efficient utilization of the scarce and over-crowded spectrum in next-generation wireless networks. It specifically aims at investigating three problems and offers corresponding solutions as follows.

- Underlay Spectrum Sharing for Cognitive Radio Networks (CRNs)
- Transparent Device-to-Device (D2D) communications in cellular networks
- Non-Orthogonal Multiple Acess (NOMA) for WLANs

The second part of the thesis focuses on the synergy between Artificial Intelligence (AI) and wireless communications. Our research efforts in this part follow two trajectories: i) leveraging AI techniques for improving the functionality of wireless networks, and ii) leveraging wireless communication and networking solutions to pave the way for making intelligence native to the next generations of wireless networks. The problems we study in this part are as follows.

- Learning-based channel sounding for IEEE 802.11ac
- Learning-based channel sounding and resource allocation for IEEE 802.11ax
- A communication framework for Federated Learning (FL) in transportation systems

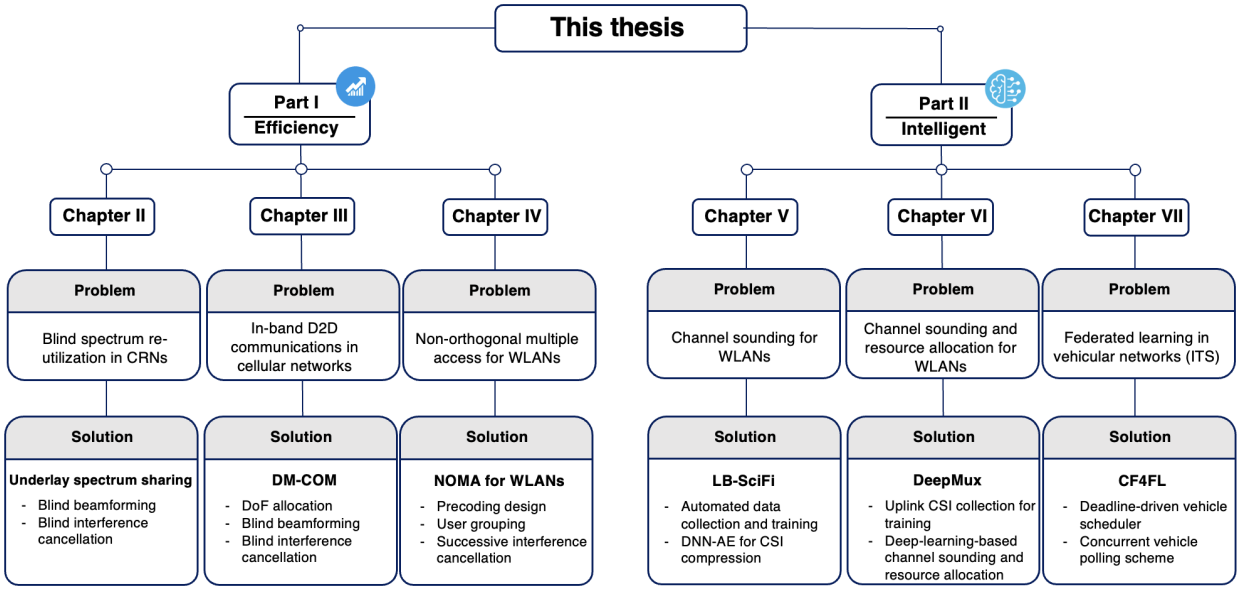


Figure 1.1: Overview of this thesis.

In the rest of this chapter, we explain our problems of interest and the shortcoming of existing solutions to each in more detail. We also briefly describe our proposed schemes and their novelty compared to state-of-the-art schemes.

1.1.1 Efficiency

The first part of the thesis intends to answer one important question; how to re-utilize the precious spectrum for establishing new communications without adversely affecting the existing ones. This question is explored in three different networking scenarios. In Chapter 2, we considered a very heterogeneous environment where the spectrum is potentially being used by a variety of technologies that are likely unknown to the new spectrum occupant we intend to introduce. While Chapter 2 focuses on a very generic scenario, Chapter 3 targets cellular networks and tailors a specific solution enabling spectrum re-utilization in the context of D2D communications. It particularly leverages the capabilities of the BS as the central coordinator of the network and lays the foundation of an interference management framework enabling D2D communications in the

presence of regular cellular devices. Finally, Chapter 4 focuses on WLANs and investigates the possibility of re-utilizing spectrum for a massive number of WLANs transmissions from a single AP to many devices without causing co-channel interference. In what follows, we further elaborate on the problems we intend to solve and the contributions of our proposed schemes.

1.1.1.1 Underlay Spectrum Sharing for Cognitive Radio Networks

Problem Statement and Existing Solutions. Sub-6 GHz frequency spectrum is very crowded while being the main carrier for the data traffic in commercial wireless systems. The scarcity of resources and rapid proliferation of devices are pushing the spectrum shortage issue to a breaking point, necessitating the enhancement in the utilization efficiency of sub-6 GHz spectrum. One promising solution is spectrum sharing in the context of CRNs. This cost-effective and immediate solution to solve the spectrum shortage issue has been a long-standing subject of study and several enablers have been proposed. The common concept among all the proposed enablers is co-channel interference management among spectrum utilizers through various signal processing techniques, such as spread spectrum [59], power control [91, 107, 175], and beamforming [49, 133]. Spread spectrum handles interference in the code domain, and power control tames interference in the power domain. Beamforming exploits the spatial Degrees of Freedom (DoF) provided by multiple antennas to steer the secondary signals to some particular directions, thereby avoiding interference for primary users. Compared to the other two techniques, beamforming is more appealing in practice as it is effective in interference management. However, the existing beamforming solutions are reliant on global network information and cross-network channel knowledge or reliant on cooperation with other occupants of the spectrum.

Proposed Scheme. We propose a practical scheme to enable transparent spectrum sharing [144] through two complementary modules, Blind BeamForming (BBF) and Blind Interference

Cancellation (BIC). These two techniques enable a new occupant of the spectrum to mitigate cross-network interference in the absence of inter-network coordination, fine-grained synchronization, and mutual knowledge from other occupants. Unlike the existing solutions, our scheme is not reliant on the momentary Channel State Information (CSI) nor seeking any coordination and cooperation from the other occupants of the spectrum for interference management and adjusting its transmission policy. We have built a prototype of our scheme on a wireless testbed and demonstrated its compatibility with commercial Wi-Fi devices. Experimental results show that our scheme is able to improve the spectral efficiency of CRNs by 1.1 bit/s/Hz in real-world wireless environments.

1.1.1.2 Transparent Device-to-Device Communications in Cellular Networks

Problem Statement and Existing Solutions. D2D communication is a promising technology for cellular networks [204] to enhance their spectral efficiency. It allows direct communication between two proximity-based mobile users without traversing the BS or core network. The advantages of D2D communications go beyond spectral efficiency. Saving the airtime at the core network, D2D offers more airtime to the BS that can be leveraged to serve a massive number of low-rate devices such as IoT sensors. It also can potentially reduce packet transmission delay, enhance user fairness, offload traffic for BSs, and alleviate congestion for core networks, especially in networks congested by IoT devices [177]. Despite its potential benefits, a D2D system needs to control co-channel interference and manage resources for competing users. In order for accomplishing these tasks, the enablers of D2D communications include beamforming [115, 165, 176], spectral resource management [30, 84, 130, 160], power control [9, 10, 61, 62, 98, 160, 174], and mode selection [15, 27, 60]. Most of existing works consider spectrum re-utilization in either uplink (see, e.g., [10, 61, 98]) or downlink (see, e.g., [115, 165, 176]) of cellular networks, but not

both. Moreover, most of the existing works require perfect global channel knowledge as well as network-wide synchronization.

Proposed Scheme. We propose DM-COM [80], a practical scheme for enabling D2D communications in cellular networks. The enabler of DM-COM is a new approach for managing the mutual interference between the D2D and cellular devices, which does not require CSI and is, therefore, amenable to practical implementation. It is also not restricted to only uplink or downlink and is compatible with both modes of transmission in cellular networks. We have built a prototype of DM-COM on a wireless testbed. Our experimental results show that using DM-COM in a small cellular network, D2D users achieve 1.9 bit/s/Hz spectral efficiency, while MU-MIMO users have less than 8% throughput degradation compared to the case without D2D users. Overall, DM-COM improves the throughput of the network by 82% compared to the case whole traffic is traversed to the BS.

1.1.1.3 Non-Orthogonal Multiple Access for WLANs

Problem Statement and Existing Solutions. NOMA allows multiple users to utilize the same spectrum band for signal transmissions at the same time and, therefore, offers many advantages such as improving spectral efficiency, enhancing resource allocation flexibility, reducing scheduling latency, increasing cell-edge throughput, and enabling massive connectivity. Although a considerable amount of research efforts have been made on the study of NOMA, most of them are limited to theoretical exploration and performance analysis in cellular networks. Very limited progress has been made so far in the development of practical NOMA schemes and experimental validation of NOMA in WLANs. This stagnation reflects the challenges in the design of practical NOMA schemes and the engineering issues related to their implementations, such as channel acquisition and precoding on the transmitter side and Successive Interference Cancellation (SIC)

realization on the receiver side. Specifically, there is no research effort focusing on the pre-coding design, user grouping, and SIC for WLANs. Moreover, there is no experimental validation of NOMA for these networks.

Proposed Scheme. We propose a practical downlink NOMA scheme for WLANs [81] and evaluate its performance in real-world wireless environments. Our NOMA scheme has three key components: precoder design, user grouping, and a new SIC scheme. On the transmitter side, we first formulate the precoding design problem as an optimization problem and then devise an efficient algorithm to construct precoders for downlink NOMA transmissions. We further propose a lightweight user grouping algorithm to ensure the success of SIC at the receivers. On the receiver side, we propose a new SIC method to decode the desired signal in the presence of strong interference. In contrast to existing SIC methods, our SIC method does not require channel estimation to decode the signals, thereby improving its resilience to interference. We have also built a prototype of the proposed NOMA scheme on a wireless testbed. This is the first experimental validation of NOMA on a real WLAN testbed. Experimental results show that, compared to Orthogonal Multiple Access (OMA), the proposed NOMA scheme considerably improves WLAN's weighted sum rate (36% on average).

1.1.2 Intelligence

The second part of the thesis focuses on both leveraging and domesticating intelligence in the next generations of wireless networks. It first turns its focus on using AI techniques for solving intricate networking and communication problems in WLANs. Second, as intelligent applications are indispensable parts of future wireless networks, it endeavors to provide a communication framework to make the intelligence native to these networks considering the application requirements and networking challenges. Particularly, Chapter 5 exploits the recent advances in Deep Learning (DL)

to relax the excessive overhead of channel sounding in the current IEEE 802.11ac [67] WLANs. If overhead is reduced, more airtime can be assigned for data transmissions, enhancing the overall throughput of WLANs in Multi-User Multiple-Input Multiple-Output (MU-MIMO) mode. While our proposed solution, LB-SciFi [79], effectively works for IEEE 802.11ac [67], it falls short in IEEE 802.11ax [70] where MU-MIMO can be mixed with Orthogonal Frequency Division Multiple Access (OFDMA). This marriage brings two challenges. It significantly scales up the channel sounding. Also, to fully exploit the gain of MU-MIMO-OFDMA mixed mode, a complicated resource allocation problem needs to be solved in real-time. To address these challenges, a novel technique beyond LB-SciFI is needed. We introduce DeepMux [82] in Chapter 6 to tackle these issues.

Finally, Chapter 7 lies within the latter trajectory of the second part of the thesis, which is the domestication of intelligence in wireless networks. We focus on Intelligent Transportation Systems (ITS) and design an elaborate communication framework to facilitate the establishment of Federated Learning (FL) in such a dynamic and heterogeneous wireless environment [78].

1.1.2.1 Learning-based Channel Feedback for MU-MIMO in WLANs

Problem Statement and Existing Solutions. To support downlink MU-MIMO communications in WLANs, an AP needs to access short-term CSI for the construction of beamforming filters. To acquire CSI, IEEE 802.11 standards [67, 70] adopted explicit CSI feedback. However, due to its reliance on over-the-air CSI feedback, it suffers from large airtime overhead. The large overhead of this method can be attributed to a large number of subcarriers in WLANs' Orthogonal Frequency-Division Multiplexing (OFDM) modulation, each of which has a channel matrix to be reported. To reduce the overhead, existing 802.11 protocols may group subcarriers and provide one CSI report per group. Apparently, such a naive scheme will lead to an inferior beamforming performance

and drastically compromises the throughput gain of MU-MIMO. Given the severity of this issue, research efforts have been devoted to studying the effect of channel acquisition parameters on network throughput or completely altering the channel acquisition paradigm to enhance network throughput [17, 33, 53, 87, 110, 113, 117, 128, 134, 139, 197, 200]. None of these works focused on reducing the overhead of explicit channel sounding. Although there are several learning-based solutions for cellular networks [54, 97, 111, 172, 179, 193], none of them can be directly applied to WLANs. CSI in WLANs are essentially real-valued spatial angles which are different from CSI in cellular networks which are complex-valued channel gains.

Proposed Scheme. We present LB-SciFi [79], a learning-based channel feedback framework for MU-MIMO in WLANs. LB-SciFi takes advantage of deep neural network autoencoder (DNN-AE) to compress CSI in 802.11 protocols, thereby conserving airtime and improving spectral efficiency. The key component of LB-SciFi is an online DNN-AE training scheme, which allows an AP to train DNN-AEs by leveraging the side information of existing 802.11 protocols. With this training scheme, DNN-AEs are capable of significantly lowering the airtime overhead for MU-MIMO while preserving its backward compatibility with incumbent Wi-Fi client devices. To the best of our knowledge, LB-SciFi is the first learning-based channel feedback framework designed for WLANs. We have implemented LB-SciFi on a wireless testbed and evaluated its performance in indoor wireless environments. Experimental results show that LB-SciFi offers an average of 73% airtime overhead reduction and increases network throughput by 69% on average compared to 802.11 feedback protocols.

1.1.2.2 Learning-based Channel Sounding and Resource Allocation for IEEE 802.11ax

Problem Statement and Existing Solutions. Wi-Fi networks are evolving from 802.11n/ac to 802.11ax so that a Wi-Fi AP is capable of utilizing the spectrum more efficiently. To do so,

802.11ax features a new transmission mode in the downlink, i.e., MU-MIMO-OFDMA mixed mode. However, this new mode comes with two major challenges. The channel sounding overhead drastically scales up as the number of potential users grows. Second, the marriage of MU-MIMO and OFDMA largely expands the optimization space of resource allocation at an 802.11ax AP, making it infeasible to pursue an optimal resource allocation solution in real time due to the limited computational power of APs. Therefore, a low-complexity, yet efficient, algorithm is needed for an AP to solve the resource allocation problem. The existing efforts related to channel sounding overhead in WLANs either focus on re-using outdated CSI or bypassing the problem through leveraging implicit channel sounding. The most related prior works are [187] and [79] which reduce the channel sounding overhead by compressing CSI in the frequency domain. However, these two efforts require coordination from Wi-Fi clients to fully or partially compress CSI. Such a luxury is not readily available in practical WLANs. On the other hand, the study on resource allocation for MU-MIMO-OFDMA in WLANs is scarce. [169] and [170] are the only works considering downlink MU-MIMO-OFDMA in WLANs. However, these two works employ greedy iterative algorithms to compute a feasible solution; making them not appealing for real-time resource allocation.

Proposed Scheme. We present DeepMux [82], a deep-learning-based MU-MIMO-OFDMA transmission scheme for 802.11ax networks. DeepMux mainly comprises two components: DL-based Channel Sounding (DLCS) and DL-based Resource Allocation (DLRA), both of which reside in APs and impose no computational/communication burden on Wi-Fi clients (unlike LB-SciFi). DLCS reduces the airtime overhead of 802.11 protocols by leveraging DNNs. It uses uplink channels to train the DNNs for downlink channels, making the training process much faster than LB-SciFi. DLRA employs a DNN to solve the mixed-integer resource allocation problem, enabling an AP to obtain a near-optimal solution in polynomial time. We have built a wireless

testbed to measure the performance of DeepMux in real-world environments. Results show that DeepMux reduces the sounding overhead by $62.0\% \sim 90.5\%$ and increases the network throughput by $26.3\% \sim 43.6\%$.

1.1.2.3 A Communication Framework for Federated Learning in Transportation Systems

Problem Statement and Existing Solutions. Machine Learning (ML) techniques have been extensively studied to extract useful knowledge from massive data collected by vehicles so as to enhance the safety and efficiency of ITS. However, the sheer amount of data collected by vehicles and privacy concerns around the collected data make it impractical to transfer raw data to a server and use of conventional centralized training scheme. While FL can be regarded as a privacy-preserving and communication-efficient alternative training paradigm for vehicular networks, the limited communication capacity of these networks along with the heterogeneous sensing, storage, and processing capabilities of individual vehicles, bring severe challenges ahead of practical implementation of FL in ITS. Recently, pioneering works [28, 29, 40, 152, 171, 189, 195] have been conducted to incorporate FL into ITS. To the best of our knowledge, existing works mainly employ cross-layer optimization techniques to enhance learning efficiency. They assume that global CSI is available on the server. They also assume that CSI remains valid for the time period of an FL iteration. Given the small channel coherence time caused by the high mobility of vehicles, these two assumptions may not be valid in practical vehicular networks. Also, the existing works either rely on inter-vehicle synchronization or separate transmissions across frequency resources.

Proposed Scheme. We present a communication framework for FL (CF4FL) in transportation systems [78]. CF4FL aims to accelerate the convergence of FL training process through the innovation of two complementary networking components, Deadline-Driven Vehicle Scheduler (DDVS) and Concurrent Vehicle Polling Scheme (CVPS). DDVS identifies a subset of vehicles for local

model training in each iteration of FL, with the aim of minimizing data loss while respecting the deadline constraints derived from vehicles' storage, computation, and energy budgets. CVPS takes advantage of multiple antennas on an edge server to enable concurrent local model transmissions in dynamic vehicular networks, thereby reducing the airtime overhead of each FL iteration. CF4FL needs neither inter-vehicle synchronization nor instantaneous CSI for asynchronous concurrent vehicle transmissions. Our simulation results show that CF4FL reduces the convergence time of FL training by more than 39% compared to the existing solution.

1.2 Organization

The rest of the thesis is organized as follows: Chapter 2 presents a blind spectrum sharing scheme for CRNs. Chapter 3 presents a communication framework enabling concurrent MU-MIMO and D2D communications in cellular networks, yielding high spectral efficiency and throughput. Chapter 4 presents a downlink power-domain NOMA scheme for WLANs in detail. Chapter 5 describes LB-SciFi, a learning-based compression approach for reducing excessive airtime overhead in downlink MU-MIMO of WLANs. Chapter 6 presents DeepMux and its underlying modules for low-overhead channel sounding and fast resource allocation in downlink MU-MIMO-OFDMA mode of WLANs. Chapter 7 deals with designing a communication framework to accelerate FL in ITS and challenges ahead of its successful deployment. Finally, Chapter 8 enumerates possible research directions for our future research endeavors.

Chapter 2

Underlay Spectrum Sharing for CRNs

2.1 Introduction

The rapid proliferation of wireless devices and the burgeoning demands for wireless services have pushed the spectrum shortage issue to a breaking point. Although it is expected that much spectrum in the millimeter band (30 GHz to 300 GHz) will be allocated for communication purposes, most of this spectrum might be limited to short-range communications due to its severe path loss. Moreover, millimeter band is highly vulnerable to blockage and thus mainly considered for complementary use in next-generation wireless systems. As envisioned, sub-6 GHz frequency spectrum, which is already very crowded, will still be the main carrier for the data traffic in commercial wireless systems. Therefore, it is very necessary to maximize the utilization efficiency of sub-6 GHz spectrum.

To improve spectrum utilization efficiency, spectrum sharing in the context of CRNs has been widely regarded as a promising and cost-effective solution. In the past two decades, CRNs have received a large amount of research efforts and produced many cognitive radio schemes. Depending on the spectrum access strategy at secondary users, the existing cognitive radio schemes can be classified to three paradigms: interweave, overlay, and underlay [51]. In the *interweave* paradigm, secondary users exploit spectrum white holes and intend to access the spectrum opportunistically when primary users are idle. In the *overlay* paradigm, secondary users are allowed to access spec-

trum simultaneously with primary users, provided that the primary users share the knowledge of their signal codebooks and messages with the secondary users. Compared to these two paradigms, the *underlay* paradigm is more appealing as it allows secondary users to concurrently utilize the spectrum with primary users while requiring neither coordination nor knowledge from the primary users.

Although there is a large body of work on underlay CRNs in the literature, most of existing work is either focused on theoretical exploration or reliant on unrealistic assumptions such as cross-network channel knowledge and inter-network coordination (see, e.g., [34,89,91,107,122,138,146, 166, 175]). Thus far, very limited progress has been made in the design of practical underlay spectrum sharing schemes. To the best of our knowledge, there is no underlay spectrum sharing scheme that has been implemented in real-world wireless environments. This stagnation underscores the challenge in such a design, which is reflected in the following tasks: i) at a secondary transmitter, how to pre-cancel its generated interference for the primary receivers in its close proximity; and ii) at a secondary receiver, how to decode its desired signals in the presence of unknown interference from primary transmitters. These two tasks become even more challenging when secondary users have no knowledge (e.g., signal waveform and frame structure) about primary users.

In this chapter, we consider an underlay CRN that comprises a pair of primary users and a pair of secondary users. We assume that the secondary users are equipped with more antennas than the primary users. By leveraging their multiple antennas, the secondary users take the full responsibility for cross-network Interference Cancellation (IC). For such a CRN, we propose a practical spectrum sharing scheme that allows the secondary users to access the spectrum while remaining transparent to the primary users. The key components of our scheme are two interference management techniques: BBF and BIC.

The proposed BBF technique is used at the secondary transmitter to pre-cancel its generated

interference for the primary receiver. In contrast to existing beamforming techniques, which require channel knowledge for the construction of beamforming filters, our BBF technique does not require channel knowledge. Instead, it constructs the beamforming filters by exploiting the statistical characteristics of the overheard interfering signals from the primary users. The proposed BIC technique is used at the secondary receiver to decode its desired signals in the presence of unknown interference from the primary transmitter. Unlike existing IC techniques, which require CSI and inter-network synchronization, our BIC technique requires neither cross-network channel knowledge nor inter-network synchronization for signal detection. Rather, it leverages the reference symbols (preamble) embedded in the data frame of secondary users to construct the decoding filters for signal detection in the face of unknown interference. With these two IC techniques, the secondary users can effectively mitigate the cross-network interference in the absence of coordination from the primary users.

We have built a prototype of our scheme on a wireless testbed to evaluate its performance in real-world wireless environments. Particularly, we have demonstrated that our prototyped secondary devices share 2.4 GHz spectrum with commercial Wi-Fi devices (primary users) while not affecting Wi-Fi devices' throughput. A demo video of our scheme is presented in [131]. We further conduct experiments to evaluate the performance of our secondary network in coexistence with LTE-like and CDMA-like primary networks in the following two cases: i) the primary users are equipped with one antenna and the secondary users equipped with two antennas; and ii) the primary users are equipped with two antennas and the secondary users equipped with three antennas. Experimental results measured in an office environment show that the secondary network can achieve an average of 1.1 bits/s/Hz spectrum utilization without visibly degrading primary network throughput. Moreover, the proposed BBF and BIC techniques achieve an average of 25 dB and 33 dB IC capabilities, respectively.

The contributions of this work are summarized as follows:

- We have designed a new BIC technique for a wireless receiver, which is capable of decoding its data packets in the presence of unknown interference. Our prototype of such a wireless receiver can achieve 33 dB IC capability for unknown interference in real-world tests.
- We have designed a new BBF technique for a wireless transmitter, which is capable of pre-canceling its generated interference for an unintended receiver without the need of channel knowledge. Our prototype of such a wireless transmitter can achieve 25 dB IC capability for the unintended receiver.
- To the best of our knowledge, our work is the first one that demonstrates real-time concurrent spectrum utilization of two wireless systems in the absence of inter-network coordination and fine-grained synchronization.

2.2 Related Work

We focus our literature survey on spectrum sharing in underlay CRNs and the related interference management techniques.

Spectrum Sharing in Underlay CRNs. Underlay CRNs allow concurrent spectrum utilization for primary and secondary networks as long as the interference at primary users remains at an acceptable level. Different signal processing techniques have been studied for interference management in underlay CRNs, such as spread spectrum [59], power control [91, 107, 175], and beamforming [3, 4, 6, 8, 24, 31, 46, 49, 56, 57, 64, 118, 119, 125, 133, 190, 191, 202, 203]. Spread spectrum handles interference in the code domain, and power control tames interference in the power domain. Beamforming exploits the spatial DoF provided by multiple antennas to steer the

secondary signals to some particular directions, thereby avoiding interference for primary users. Compared to the other two techniques, beamforming is more appealing in practice as it is effective in interference management.

Given its potential, beamforming has been studied in underlay CRNs to pursue various objectives, such as improving energy efficiency of secondary transmissions [49, 64, 133, 191], maximizing data rate of secondary users [3, 4], maximizing sum rate of both primary and secondary users [8, 46, 56, 57], and enhancing the security against eavesdroppers [118, 119, 202]. However, most of these beamforming solutions are reliant on global network information and cross-network channel knowledge. Our work differs from these efforts as it requires neither cross-network channel knowledge nor inter-network cooperation.

BBF in Underlay CRNs. There are some pioneering works that studied BBF to eliminate the requirement of cross-network channel knowledge for the design of beamforming filters [6, 24, 31, 125, 190, 203]. In [203] and [31], an eigen-value-decomposition-based approach was proposed to construct beamforming filters at a secondary transmitter using its received interfering signals from a primary device. When the secondary device transmitting, the constructed beamforming filters would steer its radio signals to the null subspace of the cross-network channel, thereby avoiding interference for the primary device. Our BBF technique follows similar idea, but differs in the network setting and design objective. Specifically, [203] and [31] were focused on theoretical analysis to optimize the data rate of secondary users under certain interference temperature, while the BBF technique in our work is designed to guarantee its practicality and optimize its IC capability in real-world OFDM-based networks.

In [6] and [24], the beamforming design is formulated as a part of a network optimization problem, and some constraints are developed based on statistical channel knowledge to relax the requirement of cross-network channel knowledge. This approach is of high complexity, and it

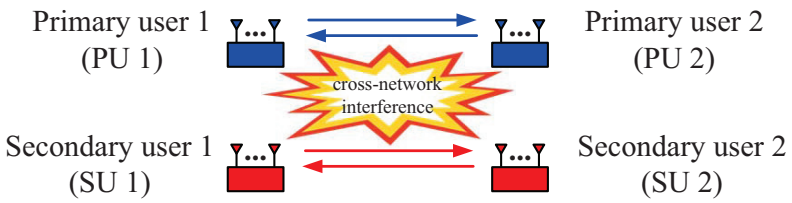


Figure 2.1: A CRN consisting of two active primary users and two active secondary users.

seems not amenable to practical implementation. on the In [125] and [190], spatial learning methods were proposed to iteratively adjust beamforming filters at the secondary devices based on the power level of primary transmission, with the objective of reducing cross-network interference for primary users. However, these learning-based methods are cumbersome and not amenable to practical use.

MIMO-based BIC. While there are many results on interference cancellation in cooperative wireless networks, the results of MIMO-based BIC in non-cooperative networks remain limited. In [142], Rousseaux et al. proposed a MIMO-based BIC technique to handle interference from one source. In [182], Winters proposed a spatial filter design for signal detection at multi-antenna wireless receivers to combat unknown interference. In [52], Gollakota et al. proposed a MIMO-based solution to mitigate narrow-band interference from home devices such as microwave. BIC was further studied in the context of radio jamming in wireless communications (see, e.g., [149, 192]). Compared to the existing BIC techniques, our BIC technique has a lower complexity and offers much better performance (33 dB IC capability in our experiments).

2.3 Problem Statement

We consider an underlay CRN as shown in Fig. 2.1, which consists of two active primary users and two secondary users. The primary users establish bidirectional communications in Time-Division Duplex (TDD) mode. The traffic flow in the primary network is persistent and consistent in both

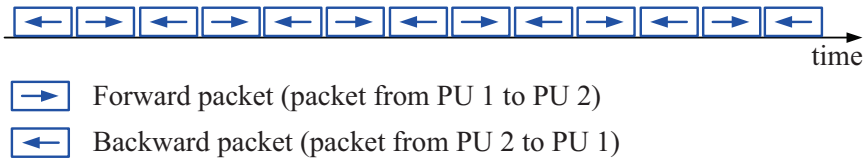


Figure 2.2: Consistent and persistent traffic in the primary network.

directions, as shown in Fig. 2.2. The secondary users want to utilize the same spectrum for their own communications. To do so, the secondary transmitter employs beamforming to pre-cancel its generated interference for the primary receiver; and the secondary receiver performs IC for its signal detection. Simply put, the secondary users take full burden of cross-network interference cancellation, and their data transmissions are transparent to the primary users.

In this CRN, there is no coordination between the primary and secondary users. The secondary users have no knowledge about cross-network interference characteristics. The primary users have one or multiple antennas, and the number of their antennas is denoted by M_{p}. The secondary users have multiple antennas, and the number of their antennas is denoted by M_{s}. We assume that the number of antennas on a secondary user is greater than that on a primary user, i.e., $M_{\mathrm{s}} > M_{\mathrm{p}}$. This assumption ensures that each secondary user has sufficient spatial DoF to tame cross-network interference.

Our Objective. In such a CRN, our objective is four-fold: i) design a BBF technique for the secondary transmitter to pre-cancel its generated interference for the primary receiver; ii) design a BIC technique for the secondary receiver to decode its desired signals in the presence of interference from the primary transmitter; iii) design a spectrum sharing scheme by integrating these two IC techniques; and iv) evaluate the IC techniques and the spectrum sharing scheme via experimentation in real wireless environments.

Two Justifications: First, in this work, we study a CRN that comprises one pair of primary users and one pair of secondary users. Although it has a small network size, such a CRN serves as a

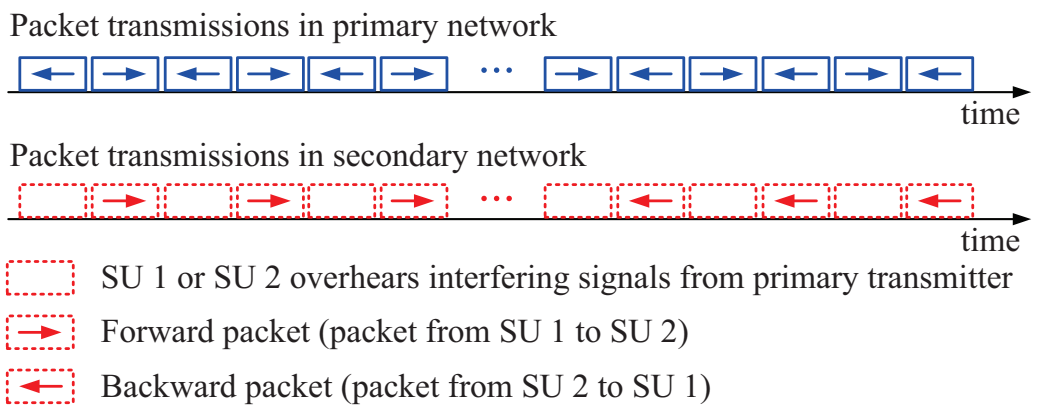


Figure 2.3: A MAC protocol for spectrum sharing in a CRN that has two primary users and two secondary users.

fundamental building block for a large-scale CRN that have many primary and secondary users. Therefore, understanding this small CRN is of theoretical and practical importance. Second, in our study, we assume that the secondary users have no knowledge about cross-network interference characteristics. Such a conservative assumption leads to a more robust spectrum sharing solution, which is suited for many application scenarios.

2.4 A Spectrum Sharing Scheme

In this section, we present a spectrum sharing scheme for the secondary network so that it can use the same spectrum for its communications while almost not affecting performance of the primary network. Our scheme consists of a lightweight MAC protocol and a new PHY design for the secondary users. In what follows, we first present the MAC protocol and then describe the new PHY design.

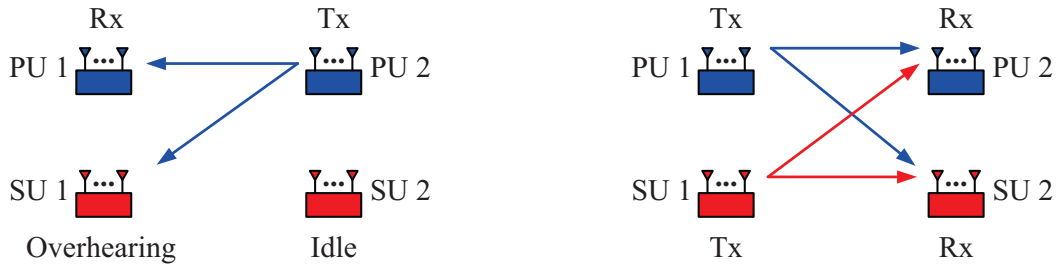
2.4.1 MAC Protocol for Secondary Network

Fig. 2.3 shows our MAC protocol in the time domain. It includes both forward communications (from SU 1 to SU 2) and backward communications (from SU 2 to SU 1) between the two secondary users. Since the two communications are symmetric, our presentation in the following will focus on the forward communications. The backward communications can be done in the same way.

The forward communications in the proposed MAC protocol comprise two phases: *overhearing (Phase I)* and *packet transmission (Phase II)*. In the time domain, Phase I aligns with the backward packet transmissions of the primary network, and Phase II aligns with the forward packet transmissions of the primary network, as illustrated in Fig. 2.3. We elaborate the operations in the two phases as follows:

- **Phase I:** SU 1 overhears the interfering signals from PU 2, and SU 2 remains idle, as shown in Fig. 2.4(a).
- **Phase II:** SU 1 first constructs beamforming filters using the overheard interfering signals in Phase I and then transmits signals to SU 2 using the constructed beamforming filters. Meanwhile, SU 2 decodes the signals from SU 1 in the presence of interference from PU 1. Fig. 2.4(b) shows the packet transmission in this phase.

When the primary network has consistent and persistent bidirectional traffic, it is easy for secondary devices to learn primary transmission direction and duration by leveraging wireless signals' spatial signature (e.g., signal angle-of-arrival). Based on the learned information, the secondary network can align its transmissions with the transmissions in the primary network, as illustrated in Fig. 2.3. It is noteworthy that the time alignment requirement of primary and secondary trans-



(a) Phase I: SU 1 overhears the interfering signals from PU 2.

(b) Phase II: SU 1 sends data to SU 2 using IC techniques.

Figure 2.4: Illustration of our proposed spectrum sharing scheme.

missions is loose, thanks to the capability of BBF and BIC at the PHY layer. To ensure that the secondary transmissions will not disrupt the primary transmissions, SU 1 sends its signals only after it detects the interfering signals from PU 2.

2.4.2 PHY Design for Secondary Users: An Overview

To support the proposed MAC protocol, we use IEEE 802.11 legacy PHY for the secondary network, including frame structure, OFDM modulation, and channel coding schemes. However, IEEE 802.11 legacy PHY is vulnerable to cross-network interference. Therefore, we need to modify the legacy PHY for the secondary users. The modified PHY should be resilient to cross-network interference on both transmitter and receiver sides. The design of such a PHY faces the following two challenges.

Challenge 1. Referring to Fig. 2.4(b), the main task of the secondary transmitter (SU 1) is to pre-cancel its generated interference for the primary receiver (PU 2). Note that we assume the secondary transmitter has no knowledge about the primary network, including the signal waveform, bandwidth, and frame structure. The primary network may use OFDM, Code-Division Multiple Access (CDMA) or other types of modulation for packet transmission. The lack of knowledge about the interference makes it challenging for SU 1 to cancel the interference.

To address this challenge, we design a BBF technique for the secondary transmitter (SU 1) to pre-cancel its interference at the primary receiver. Our beamforming technique takes advantage of the overheard interfering signals in Phase I to construct precoding vectors for beamforming. Our BBF technique can completely pre-cancel the interference at the primary receiver if noise is zero and the reciprocity of forward/backward channels is maintained. Details of this beamforming technique are presented in Section 2.5.

Challenge 2. Referring to Fig. 2.4(b) again, the main task of the secondary receiver (SU 2) is to decode its desired signals in the presence of unknown cross-network interference. Note that the secondary receiver has no knowledge about the interference characteristics, and the primary and secondary networks may use different waveforms and frame formats for their transmissions. The lack of inter-network coordination, cross-network knowledge and fine-grained synchronization makes it challenging to tame interference for signal detection.

To address this challenge, we design a MIMO-based BIC technique for the secondary receiver. The core component of our BIC technique is a spatial filter, which mitigates unknown cross-network interference from the primary transmitter and recovers the desired signals. Details of this BIC technique are presented in Section 2.6.

2.5 Blind Beamforming

In this section, we study the beamforming technique at SU 1 in Fig. 2.4. In Phase I, SU 1 first overhears the interfering signals from the primary transmitter and then uses the overheard interfering signals to construct spatial filters. Based on channel reciprocity, the constructed spatial filters are used as beamforming filters in Phase II to avoid interference at the primary receiver. These operations are performed on each subcarrier in the OFDM modulation. In what follows, we first

present the derivation of beamforming filters and then offer performance analysis of the proposed beamforming technique.

Mathematical Formulation. Consider SU 1 in Fig. 2.4(a). It overhears interfering signals from PU 2. The overheard interfering signals are converted to the frequency domain through FFT operation.[1] We assume that the channel from PU 2 to SU 1 is a block-fading channel in the time domain. That is, all the OFDM symbols in the backward transmissions experience the same channel. Denote $\mathbf{Y}(l,k)$ as the lth sample of the overheard interfering signal on subcarrier k in Phase I. Then, we have[2]

$$\mathbf{Y}(l,k) = \mathbf{H}_{\mathrm{sp}}^{[1]}(k)\mathbf{X}_{\mathrm{p}}^{[1]}(l,k) + \mathbf{W}(l,k), \tag{2.1}$$

where $\mathbf{H}_{\mathrm{sp}}^{[1]}(k) \in \mathbb{C}^{M_{\mathrm{s}} \times M_{\mathrm{p}}}$ is the matrix representation of the block-fading channel from PU 2 to SU 1 on subcarrier k, $\mathbf{X}_{\mathrm{p}}^{[1]}(l,k) \in \mathbb{C}^{M_{\mathrm{p}} \times 1}$ is the interfering signal transmitted by PU 2 on subcarrier k, and $\mathbf{W}(l,k) \in \mathbb{C}^{M_{\mathrm{s}} \times 1}$ is the noise vector at SU 1. It is noteworthy that SU 1 knows $\mathbf{Y}(l,k)$ but does not know $\mathbf{H}_{\mathrm{sp}}^{[1]}(k)$, $\mathbf{X}_{\mathrm{p}}^{[1]}(l,k)$, and $\mathbf{W}(l,k)$.

At SU 1, we seek a spatial filter that can combine the overheard interfering signals in a destructive manner. Denote $\mathbf{P}(k)$ as the spatial filter on subcarrier k. Then, the problem of designing $\mathbf{P}(k)$ can be expressed as:

$$\min \ \mathbb{E}[\mathbf{P}(k)^*\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\mathbf{P}(k)], \quad \text{s.t. } \mathbf{P}(k)^*\mathbf{P}(k) = 1, \tag{2.2}$$

where $(\cdot)^*$ represents conjugate transpose operator.

Construction of Spatial Filters. To solve the optimization problem in (2.2), we use Lagrange

[1] The interfering signals are not necessarily OFDM signals.

[2] For the notation in this chapter, superscripts "[1]" and "[2]" mean Phase I and Phase II, respectively. Subscripts "s" and "p" mean the secondary and primary users, respectively.

multipliers method. We define the Lagrange function as:

$$\mathcal{L}(\mathbf{P}(k),\lambda)=\mathbb{E}\big[\mathbf{P}(k)^*\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\mathbf{P}(k)\big]-\lambda\big[\mathbf{P}(k)^*\mathbf{P}(k)-1\big], \tag{2.3}$$

where λ is Lagrange multiplier. By setting the partial derivatives of $\mathcal{L}(\mathbf{P}(k),\lambda)$ to zero, we have

$$\frac{\partial\mathcal{L}(\mathbf{P}(k),\lambda)}{\partial\mathbf{P}(k)}=\mathbf{P}(k)^*\Big(\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]-\lambda\mathbf{I}\Big)=0, \tag{2.4}$$

$$\frac{\partial\mathcal{L}(\mathbf{P}(k),\lambda)}{\partial\lambda}=\mathbf{P}(k)^*\mathbf{P}(k)-1=0. \tag{2.5}$$

Based on the definition of eigendecomposition, it is easy to see that the solutions to equations (2.4) and (2.5) are the eigenvectors of $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$ and the corresponding values of λ are the eigenvalues of $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$. Note that $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$ has M_{s} eigenvectors, each of which corresponds to a stationary point of the Lagrange function (extrema, local optima, and global optima). As λ is the penalty multiplier for the Lagrange function, the optimal spatial filter $\mathbf{P}(k)$ lies within the subspace spanned by the eigenvectors of $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$ that correspond to the minimum eigenvalue.

For Hermitian matrix $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$, it may have multiple eigenvectors that correspond to the minimum eigenvalue. Denote M_{e} as the number of eigenvectors that correspond to the minimum eigenvalue. Then, we can write them as:

$$[\mathbf{U}_1,\mathbf{U}_2,\cdots,\mathbf{U}_{M_{\mathrm{e}}}]=mineigvectors\Big(\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]\Big), \tag{2.6}$$

where $mineigvectors(\cdot)$ represents the eigenvectors that correspond to the minimum eigenvalue.

To estimate $\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*]$ in (2.6), we average the received interfering signal samples over

time. Denote $\mathbf{Y}(l,k)$ as the lth sample of the interfering signals on subcarrier k. Then, we have

$$[\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_{M_\mathrm{e}}] = mineigvectors\Big(\sum_{l=1}^{L_\mathrm{p}} \mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\Big), \tag{2.7}$$

where L_p is the number of overheard interfering signal samples (e.g., $L_\mathrm{p} = 20$). Also, the neighboring subcarriers can be bonded to improve accuracy. Based on (2.7), the optimal filter $\mathbf{P}(k)$ can be written as:

$$\mathbf{P}(k) = \sum_{m=1}^{M_\mathrm{e}} \alpha_m \mathbf{U}_m, \tag{2.8}$$

where α_m is a weight coefficient with $\sum_{m=1}^{M_\mathrm{e}} \alpha_m^2 = 1$.

Now, we summarize the BBF technique as follows. In Phase I, SU 1 overhears the interfering signal $\mathbf{Y}(l,k)$ from PU 2. Based on the overheard interfering signals, it constructs a spatial filter $\mathbf{P}(k)$ for subcarrier k using (2.7) and (2.8). In Phase II, we use $\overline{\mathbf{P}(k)}$ as the precoding vector for beamforming on subcarrier k, where $\overline{(\cdot)}$ is the element-wise conjugate operator.

For this beamforming technique, we have the following remarks: i) This beamforming technique does not require CSI. Rather, it uses the overheard interfering signals to construct the precoding vectors for beamforming. ii) This beamforming technique requires only one-time eigendecomposition on every subcarrier. It has a computational complexity similar to Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) precoding techniques. Therefore, it is amenable to practical implementation.

IC Capability of BBF. For the performance of the proposed beamforming technique, we have the following lemma:

Lemma 1. The proposed beamforming technique completely pre-cancels interference at the primary receiver if (i) forward and backward channels are reciprocal; and (ii) noise is zero.

Proof. We first consider the signal transmission in Phase I and then consider that in the Phase II. In Phase I, if the noise is zero, we have $\mathbf{Y}(l,k) = \mathbf{H}_{\mathrm{sp}}^{[1]}(k)\mathbf{X}_{\mathrm{p}}^{[1]}(l,k)$. Then, we have

$$\sum_{l=1}^{L_{\mathrm{p}}} \mathbf{Y}(l,k)\mathbf{Y}(l,k)^* \overset{(a)}{=} L_{\mathrm{p}}\mathbb{E}[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*] \overset{(b)}{=} L_{\mathrm{p}}\mathbf{H}_{\mathrm{sp}}^{[1]}(k)\mathbf{R}_{\mathrm{x}}(k)\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^*, \tag{2.9}$$

where (a) follows from that $\mathbf{Y}(l,k)$ is a stationary random process, which is true in practice; and (b) follows from the definition of $\mathbf{R}_{\mathrm{x}}(k) = \mathbb{E}[\mathbf{X}_{\mathrm{p}}^{[1]}(l,k)\mathbf{X}_{\mathrm{p}}^{[1]}(l,k)^*]$.

Based on (3.12), we have

$$Rank\Big(\sum_{l=1}^{L_{\mathrm{p}}} \mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\Big) = Rank\Big(L_{\mathrm{p}}\mathbf{H}_{\mathrm{sp}}^{[1]}(k)\mathbf{R}_{\mathrm{x}}(k)\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^*\Big) \leq Rank\Big(\mathbf{R}_{\mathrm{x}}(k)\Big) \leq M_{\mathrm{p}}. \tag{2.10}$$

Inequation (2.10) indicates that $\sum_{l=1}^{L_{\mathrm{p}}} \mathbf{Y}(l,k)\mathbf{Y}(l,k)^*$ has at least $M_{\mathrm{s}} - M_{\mathrm{p}}$ eigenvectors that correspond to zero eigenvalues. This further indicates that $[\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_{M_{\mathrm{e}}}]$ in (2.7) are corresponding to zero eigenvalues. Therefore, we have

$$\Big(\sum_{l=1}^{L_{\mathrm{p}}} \mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\Big)\mathbf{U}_m = \mathbf{0}, \ \text{ for } 1 \leq m \leq M_{\mathrm{e}}. \tag{2.11}$$

Based on (3.12) and (2.11), we have

$$\Big(L_{\mathrm{p}}\mathbf{H}_{\mathrm{sp}}^{[1]}(k)\mathbf{R}_{\mathrm{x}}(k)\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^*\Big)\mathbf{U}_m = \mathbf{0}, \ \text{ for } 1 \leq m \leq M_{\mathrm{e}}. \tag{2.12}$$

In real wireless environments, we have $Rank\big(\mathbf{H}_{\mathrm{sp}}^{[1]}(k)\big) = M_{\mathrm{p}}$ and $Rank\big(\mathbf{R}_{\mathrm{x}}(k)\big) = M_{\mathrm{p}}$. Therefore, the following equation can be deducted from (2.12).

$$\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^*\mathbf{U}_m = \mathbf{0}, \ \text{ for } 1 \leq m \leq M_{\mathrm{e}}. \tag{2.13}$$

Based on (2.8) and (2.13), we have

$$\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^{*}\mathbf{P}(k) = \sum_{m=1}^{M_{\mathrm{e}}} \alpha_m \mathbf{H}_{\mathrm{sp}}^{[1]}(k)^{*}\mathbf{U}_m = \mathbf{0}. \tag{2.14}$$

We now consider signal transmission in Phase II (see Fig. 2.4(b)). Denote $\mathbf{H}_{\mathrm{ps}}^{[2]}$ as the matrix representation of the channel from SU 1 to PU 2 on subcarrier k in Phase II. Given that the forward and backward channels in the two phases are reciprocal, we have $\mathbf{H}_{\mathrm{ps}}^{[2]} = \left(\mathbf{H}_{\mathrm{sp}}^{[1]}\right)^{T}$. Then, we have

$$\mathbf{H}_{\mathrm{ps}}^{[2]}(k)\overline{\mathbf{P}(k)} = \left(\mathbf{H}_{\mathrm{sp}}^{[1]}\right)^{T}\overline{\mathbf{P}(k)} = \overline{\mathbf{H}_{\mathrm{sp}}^{[1]}(k)^{*}\mathbf{P}(k)} = \mathbf{0}. \tag{2.15}$$

It means that the precoding vector $\overline{\mathbf{P}(k)}$ is orthogonal to the interference channel $\mathbf{H}_{\mathrm{ps}}^{[2]}(k)$. Therefore, we conclude that the proposed beamforming scheme can completely pre-cancel the interference from the secondary transmitter at the primary receiver in Phase II. ■

The proof of Lemma is based on reciprocity of channels in backward and forward transmissions. To maintain the reciprocity of forward and backward channels in practical wireless systems, we can employ the relative calibration method in [150]. This relative calibration method is an internal and standalone method that can be done with assistance from one device. In our experiments, we have implemented this calibration method to preserve the channels reciprocity.

2.6 Blind Interference Cancellation

In this section, we focus on SU 2 in Phase II as shown in Fig. 2.4(b). We design a BIC technique for the secondary receiver (SU 2) to decode its desired signals in the presence of interference from the primary transmitter (PU 1).

Mathematical Formulation. Recall that we use IEEE 802.11 legacy PHY for data transmissions in the secondary network. Specifically, SU 1 sends packet-based signals to SU 2, which comprise a bulk of OFDM symbols. In each packet, the first four OFDM symbols carry preambles (pre-defined reference signals) and the remaining OFDM symbols carry payloads.

Consider the signal transmission in Fig. 2.4(b). Denote $X_{\mathrm{s}}^{[2]}(l,k)$ as the signal that SU 1 transmits on subcarrier k in OFDM symbol l. Denote $\mathbf{X}_{\mathrm{p}}^{[2]}(l,k)$ as the signal that PU 1 transmits on subcarrier k in OFDM symbol l.[3] Denote $\mathbf{Y}(l,k)$ as the received signal vector at SU 2 on subcarrier k in OFDM symbol l. Then, we have

$$\mathbf{Y}(l,k) = \mathbf{H}_{\mathrm{ss}}^{[2]}(k)\overline{\mathbf{P}(k)}X_{\mathrm{s}}^{[2]}(l,k) + \mathbf{H}_{\mathrm{sp}}^{[2]}(k)\mathbf{X}_{\mathrm{p}}^{[2]}(l,k) + \mathbf{W}(l,k), \tag{2.16}$$

where $\mathbf{H}_{\mathrm{ss}}^{[2]}(k)$ is the block-fading channel between SU 2 and SU 1 on subcarrier k, $\mathbf{H}_{\mathrm{sp}}^{[2]}(k)$ is the block-fading channel between SU 2 and PU 1 on subcarrier k, and $\mathbf{W}(l,k)$ is noise on subcarrier k in OFDM symbol l.

At SU 2, in order to decode the intended signal in the presence of cross-network interference, we use a linear spatial filter $\mathbf{G}(k)$ for all OFDM symbols on subcarrier k. Then, the decoded signal can be written as:

$$\hat{X}_{\mathrm{s}}^{[2]}(l,k) = \mathbf{G}(k)^{*}\mathbf{Y}(l,k). \tag{2.17}$$

While there exist many criteria for the design of $\mathbf{G}(k)$, our objective is to minimize the mean square error (MSE) between the decoded and original signals. Thus, the signal detection problem can be formulated as:

$$\min \;\; \mathbb{E}\Big[\,\Big|\hat{X}_{\mathrm{s}}^{[2]}(l,k) - X_{\mathrm{s}}^{[2]}(l,k)\Big|^{2}\,\Big]. \tag{2.18}$$

[3] PU 1 does not necessarily send OFDM signals. But at SU 2, the interfering signals from PU 1 can always be converted to the frequency domain using FFT operation.

Construction of Spatial Filters. To solve the optimization problem in (2.18), we use Lagrange multipliers method again. We define the Lagrange function as:

$$\mathcal{L}(\mathbf{G}(k)) = \mathbb{E}\Big[\Big| \hat{X}_{\mathrm{s}}^{[2]}(l,k) - X_{\mathrm{s}}^{[2]}(l,k) \Big|^2 \Big]. \tag{2.19}$$

Based on (2.17), (2.19) can be rewritten as:

$$\mathcal{L}(\mathbf{G}(k)) = \mathbb{E}\Big[\Big| \mathbf{G}(k)^* \mathbf{Y}(l,k) - X_{\mathrm{s}}^{[2]}(l,k) \Big|^2 \Big]. \tag{2.20}$$

Equation (2.20) is a quadratic function of $\mathbf{G}(k)$. To minimize MSE, we can take the gradient with respect to $\mathbf{G}(k)$. The optimal filter $\mathbf{G}(k)$ can be obtained by setting the gradient to zero, which we show as follows:

$$\mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\big]\mathbf{G}(k) - \mathbb{E}\big[\mathbf{Y}(l,k)X_{\mathrm{s}}^{[2]}(l,k)^*\big] = 0. \tag{2.21}$$

Based on (2.21), we obtain the optimal filter

$$\mathbf{G}(k) = \mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\big]^{+}\mathbb{E}\big[\mathbf{Y}(l,k)X_{\mathrm{s}}^{[2]}(l,k)^*\big] \ , \tag{2.22}$$

where $(\cdot)^{+}$ denotes pseudo inverse operation. Equation (2.22) is the optimal design of $\mathbf{G}(k)$ in the sense of minimizing MSE. To calculate $\mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\big]$ and $\mathbb{E}\big[\mathbf{Y}(l,k)X_{\mathrm{p}}^{[2]}(l,k)^*\big]$ in (2.22), we can take advantage of the pilot (reference) symbols in wireless systems (e.g., the preamble in IEEE 802.11 legacy frame). Denote $\mathcal{Q}_k$ as the set of pilot symbols in a frame that can be used for the design of interference mitigation filter $\mathbf{G}(k)$. Then, we can approach the statistical expectations

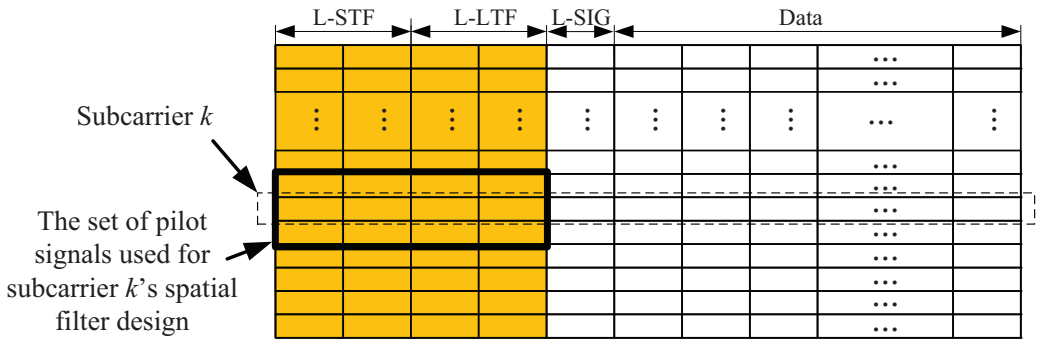


Figure 2.5: An example of $\mathcal{Q}(k)$ in IEEE 802.11 legacy frame.

in (2.22) using the averaging operations as follows:

$$\mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\big] \approx \frac{1}{|\mathcal{Q}_k|}\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')\mathbf{Y}(l,k')^*\,, \tag{2.23}$$

$$\mathbb{E}\big[\mathbf{Y}(l,k)X_{\mathrm{p}}^{[2]}(l,k)^*\big] \approx \frac{1}{|\mathcal{Q}_k|}\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')X_{\mathrm{p}}^{[2]}(l,k')^*, \tag{2.24}$$

where an example of $\mathcal{Q}_k$ is illustrated in Fig. 2.5.

Note that, with a bit abuse of notation, we replace the approximation sign in (2.23) and (2.24) with an equation sign for simplicity. Then, the spatial filter $\mathbf{G}(k)$ can be written as:

$$\mathbf{G}(k)=\Big[\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')\mathbf{Y}(l,k')^*\Big]^{+}\Big[\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')X_{\mathrm{p}}^{[2]}(l,k')^*\Big]. \tag{2.25}$$

We now summarize our BIC technique as follows. In Phase II, SU 2 needs to decode its desired signal in the presence of interference from PU 1. To do so, SU 2 first constructs a spatial filter for each of its subcarriers using (2.25), and then decodes its desired signal using (2.17).

For this BIC technique, several remarks are in order: i) The spatial filter in (2.25) not only cancels the interference but also equalizes the channel distortion for signal detection. ii) As shown in (2.17) and (2.25), our BIC technique does not require knowledge about the interference character-

istics, including waveform and bandwidth. iii) our BIC technique does not require CSI. Rather, it only requires pilot signals at the secondary transmitter. In contrast to conventional signal detection techniques (e.g., ZF and MMSE detectors), our BIC technique technique does not require channel estimation. iv) As shown in (2.17) and (2.25), the computational complexity of our BIC technique is similar to that of the ZF detector, which is widely being used in real-world wireless systems.

IC Capability of BIC. For the performance of the proposed BIC technique, we have the following lemma:

Lemma 2. If the pilot signals are sufficient and noise is zero, the BIC technique can perfectly recover the desired signals in the presence of cross-network interference (i.e., $\hat{X}_{\mathrm{s}}^{[2]}(k,l) = X_{\mathrm{s}}^{[2]}(k,l)$, $\forall k, l$).

Proof. For notational simplicity, we denote $\mathbf{H}(k)$ as the compound channel between the SU 2 and the two transmitters (SU 1 and PU 1), i.e., $\mathbf{H}(k) = \left[\mathbf{H}_{\mathrm{ss}}^{[2]}(k)\overline{\mathbf{P}(k)} \;\; \mathbf{H}_{\mathrm{sp}}^{[2]}(k)\right]$; we also denote $\mathbf{X}(l,k)$ as the compound transmit signals at the two transmitters, i.e., $\mathbf{X}(l,k) = \left[X_{\mathrm{s}}^{[2]}(l,k) \;\; \mathbf{X}_{\mathrm{p}}^{[2]}(l,k)\right]^T$. Then, in noise-negligible scenarios, (2.16) can be rewritten as $\mathbf{Y}(l,k) = \mathbf{H}(k)\mathbf{X}(l,k)$.

By defining $\mathbf{R}_{\mathrm{X}}$ as the autocorrelation matrix of the compound transmit signals, we have

$$\mathbf{R}_{\mathrm{X}} = \mathbb{E}(\mathbf{X}\mathbf{X}^H) \overset{(a)}{=} \begin{bmatrix} R_{\mathrm{xs}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathrm{xp}} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathrm{xp}} \end{bmatrix}, \tag{2.26}$$

where R_{xs} is the autocorrelation of SU 1's transmit signal and $\mathbf{R}_{\mathrm{xp}}$ is the autocorrelation matrix of PU 1's transmit signals. (a) follows from our assumption that the transmit signal from SU 1 is independent of the transmit signals from PU 1. Note that $\mathbf{R}_{\mathrm{xp}}$ is not necessarily an identity matrix since the signals from PU 1's different antennas might be correlated.

Based on (2.25), (3.4), and (3.5), we have

$$\begin{aligned}\mathbf{G}(k) &= \Big[\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')\mathbf{Y}(l,k')^H\Big]^+\Big[\sum_{(l,k')\in\mathcal{Q}_k}\mathbf{Y}(l,k')X_{\mathrm{S}}^{[2]}(l,k')^*\Big] \\ &\overset{(a)}{=} \mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^*\big]^+\mathbb{E}\big[\mathbf{Y}(l,k)X_{\mathrm{S}}^{[2]}(l,k)^*\big] \\ &\overset{(b)}{=} \big[\mathbf{H}(k)\mathbf{R}_{\mathrm{X}}\mathbf{H}(k)^*\big]^+\big[\mathbf{H}(k)\mathbf{I}_1\big], \end{aligned} \tag{2.27}$$

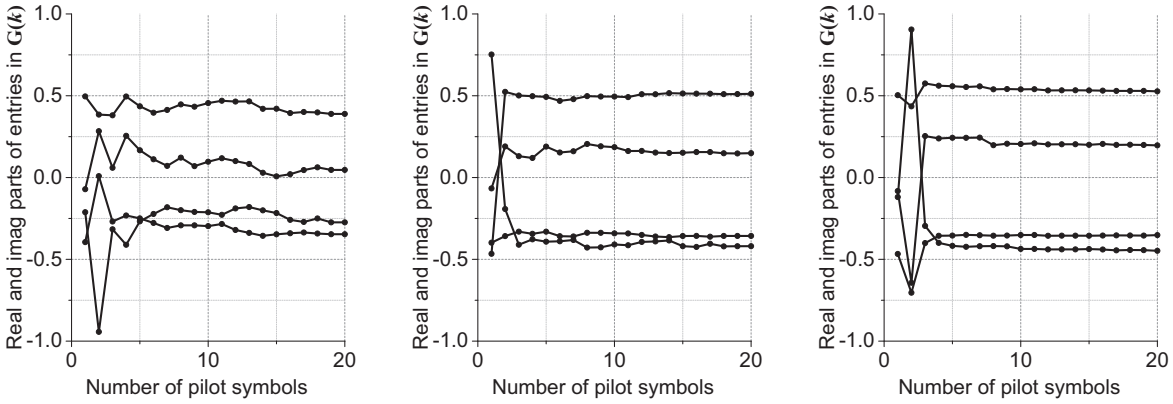
where (a) follows from our assumption that the amount of reference signals is sufficient to achieve convergence of $\mathbf{G}(k)$; (b) follows from the definition that $\mathbf{I}_1$ is a vector where its first entry is 1 and all other entries are 0. Based on (2.17) and (3.6), we have

$$\begin{aligned}\hat{X}_{\mathrm{S}}^{[2]}(l,k) &= \mathbf{G}(k)^*\mathbf{Y}(l,k) \\ &= \Big\{\big[\mathbf{H}(k)\mathbf{R}_{\mathrm{X}}\mathbf{H}(k)^*\big]^+\big[\mathbf{H}(k)\mathbf{I}_1\big]\Big\}^*\mathbf{H}(k)\mathbf{X}(l,k) \\ &= X_{\mathrm{S}}^{[2]}(l,k), \quad \forall l,k. \end{aligned} \tag{2.28}$$

■

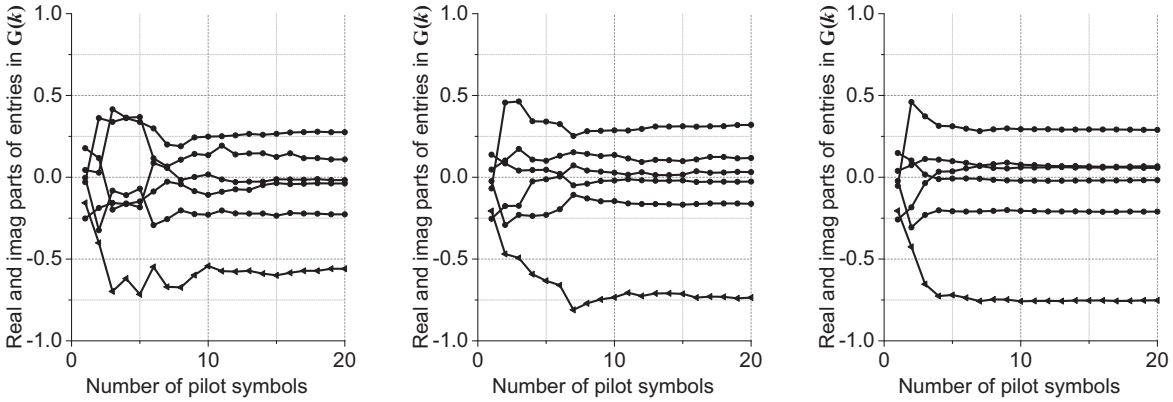
Pilot Signals for Spatial Filter Construction. Lemma 2 shows the superior performance of our BIC technique when the pilot signals are sufficient. A natural question to ask is how many pilot signals are considered to be sufficient. To answer this question, we first present our simulation results to study the convergence speed of the spatial filter over the number of pilot signals, and then propose a method to increase the number of pilot signals for the spatial filter construction.

As an instance, we simulated the convergence speed of the spatial filter over the number of pilot symbols for SU 2 in Fig. 2.4. Fig. 2.6 and Fig. 2.7 present our simulation results in two network settings: $(M_{\mathrm{p}} = 1, M_{\mathrm{S}} = 2)$ and $(M_{\mathrm{p}} = 2, M_{\mathrm{S}} = 3)$. From the simulation results, we can see



(a) SNR=5dB case. (b) SNR=15dB case. (c) SNR=25dB case.

Figure 2.6: Convergence speed of spatial filter over the number of pilot symbols in $(M_{\mathrm{p}} = 1, M_{\mathrm{s}} = 2)$ network.

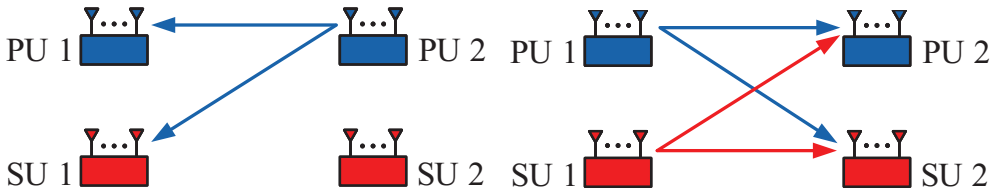


(a) SNR=5dB case. (b) SNR=15dB case. (c) SNR=25dB case.

Figure 2.7: Convergence speed of spatial filter over the number of pilot symbols in $(M_{\mathrm{p}} = 2, M_{\mathrm{s}} = 3)$ network.

that the spatial filter converges at a pretty fast speed in these two network settings. Specifically, the spatial filter can achieve a good convergence within about 10 pilot symbols.

Recall that the secondary network uses IEEE 802.11 legacy frame for transmissions from SU 1 to SU 2, which only has four pilot symbols on each subcarrier (i.e., two L-STF OFDM symbols and two L-LTF OFDM symbols). So, the construction of spatial filter is in shortage of pilot symbols. To address this issue, for each subcarrier, we not only use the pilot symbols on that subcarrier but also the pilot symbols on its neighboring subcarriers, as illustrated in Fig. 2.5. The rationale behind this operation lies in the fact that channel coefficients on neighboring subcarriers are highly



(a) Transmission in Phase I. (b) Transmission in Phase II.

Figure 2.8: Experimental setup for an underlay CRN with two network settings: $(M_p{=}1, M_s{=}2)$ and $(M_p{=}2, M_s{=}3)$.

correlated in real-world wireless environments. By leveraging the pilot symbols on two neighboring subcarriers, we have 12 pilot symbols for the construction of the spatial filter, which appears to be sufficient based on our simulation results in Fig. 2.6 and Fig. 2.7. We note that analytically studying the performance of BIC with respect to the number and format of pilot signals is beyond the scope of this work. Instead, we resort to experiments to study its performance in real-world network settings.

2.7 Performance Evaluation

In this section, we consider an underlay CRN in two time slots as shown in Fig. 2.8. We have built a prototype of the proposed underlay spectrum sharing scheme in this network on a Software-Defined Radio (SDR) testbed and evaluated its performance in real-world wireless environments.

2.7.1 Implementation

PHY Implementation. We consider three different primary networks: a commercial Wi-Fi primary network, a LTE-like primary network, and a CDMA-like primary network. The commercial Wi-Fi network comprises Alfa AWUS036NHA 802.11n Adapters, each of which has one antenna for radio signal transmissions and receptions. The LTE-like and CDMA-like primary networks as well as the secondary network are built using USRP N210 devices and general-purpose com-

Table 2.1: The implementation parameters of primary and secondary networks.

	Primary network 1	Primary network 2	Primary network 3	Secondary network
System type	Commercial	Custom-built	Custom-built	Custom-built
Standard	Wi-Fi	LTE-like	CDMA-like	Wi-Fi-like
Waveform	OFDM	OFDM	CDMA	OFDM
FFT-Point	64	1024	-	64
Valid subcarriers	52	600	-	52
Sample rate	20 MSps	10 MSps	5 MSps	5, 25 MSps
Signal bandwidth	$\sim$16 MHz	$\sim$5.8 MHz	$\sim$5 MHz	$\sim$4.06, 20.31 MHz
Carrier frequency	2.48 GHz	2.48 GHz	2.48 GHz	2.48 GHz
Max tx power	$\sim$20 dBm	$\sim$15 dBm	$\sim$15 dBm	$\sim$15 dBm
Antenna number	1	1, 2	1	2, 3

puters. The USRP devices are used for radio signal transmission/reception while the computers are used for baseband signal processing and MAC protocol implementation. The implementation parameters are listed in Table 2.1.

MAC Implementation. We implement the MAC protocol in Fig. 2.3 for the primary and secondary networks. The packet transmissions in the two networks are loosely aligned in time, as shown in Fig. 2.3. Since the bidirectional communications in the secondary network are symmetric, we only consider the forward communications (from SU 1 to SU 2). We implement BBF on SU 1 to pre-cancel interference for the primary receiver. We also implement BIC on SU 2 to decode its desired signals in the presence of interference from PU 1. Moreover, we implement the RF chain calibration method [150] on SU 1 in Fig. 2.8 to maintain relative channel reciprocity. Note that the calibration needs to be done at a low frequency (0.1 Hz in our experiments) and therefore would not consume much airtime resource.

2.7.2 Experimental Setup and Performance Metrics

Experimental Setup. Consider the primary and secondary networks in Fig. 2.8. We place the devices on a floor plan as shown in Fig. 2.9(a). The two primary users are always placed at the

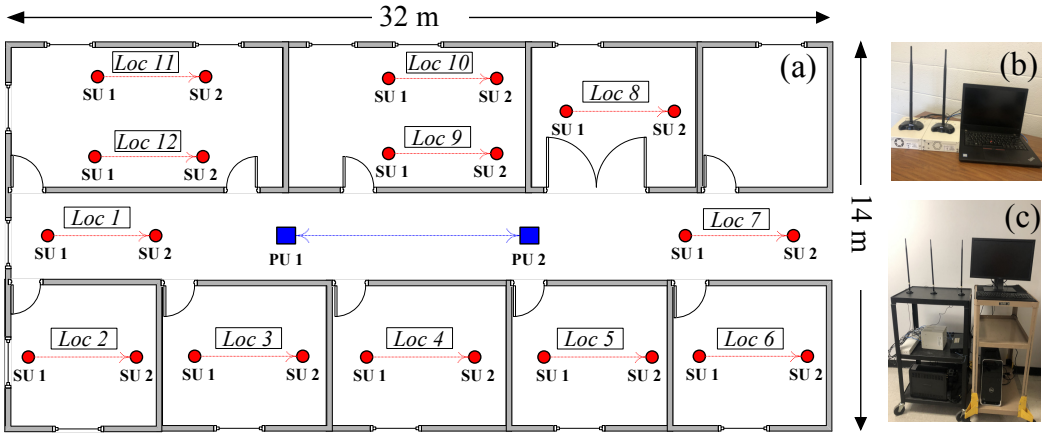


Figure 2.9: Experimental setting: (a) floor plan of primary and secondary users' locations; (b) a primary transceiver; and (c) a secondary transceiver.

spots marked "PU 1" and "PU 2." The two secondary users are placed at one of the 12 different locations. The distance between PU 1 and PU 2 is 10 m and the distance between SU 1 and SU 2 is 6 m. Fig. 2.9(b-c) show the prototyped secondary and primary transceivers on our wireless testbed. The transmit power of primary users is fixed to the maximum level specified in Table 2.1, while the transmit power of secondary users is properly adjusted to ensure that its generated interference to the primary receiver (after BBF) is below noise level.

Performance Metrics. We evaluate the performance of the proposed spectrum sharing scheme using the following four metrics: *i) Tx-side IC capability at SU 1:* This IC capability is from SU 1's BBF. It is defined as $\beta_{\text{tx}} = 10\log_{10}(P_1/P_2)$, where P_1 is the received interference power at PU 2 when SU 1 uses $[\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}}]$ or $[\frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}}]$ as the precoder, and P_2 is the received interference power at PU 2 when SU 1 uses our BBF precoder. *ii) Rx-side IC capability at SU 2:* This IC capability is from SU 2's BIC. It is defined as $\beta_{\text{rx}} = |\text{EVM}| - \max\{SIR_m\}$, where SIR_m is Signal to Interference Ratio (SIR) on SU 2's mth antenna and Error Vector Magnitude (EVM) will

Table 2.2: EVM specification in IEEE 802.11ac [67].

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

Table 2.3: EVM specification for LTE-like PHY [43, 77].

EVM (dB)	[-6.3 -9.1)	[-9.1 -11.8)	[-11.8 -14.2)	[-14.2 -16.8)	[-16.8 -19.1)
CQI	6	7	8	9	10
Modulation	QPSK	16QAM	16QAM	16QAM	64QAM
Coding rate $\times$1024	602	378	490	616	466
γ(EVM)	1.1758	1.4766	1.9141	2.4063	2.7305
EVM (dB)	[-19.1 -21.0)	[-21.0 -23.3)	[-23.3 -25.7)	[-25.7 -28.2)	[-28.2 -∞)
CQI	11	12	13	14	15
Modulation	64QAM	64QAM	64QAM	64QAM	64QAM
Coding rate $\times$1024	567	666	772	873	948
γ(EVM)	3.3223	3.9023	4.5234	5.1152	5.5547

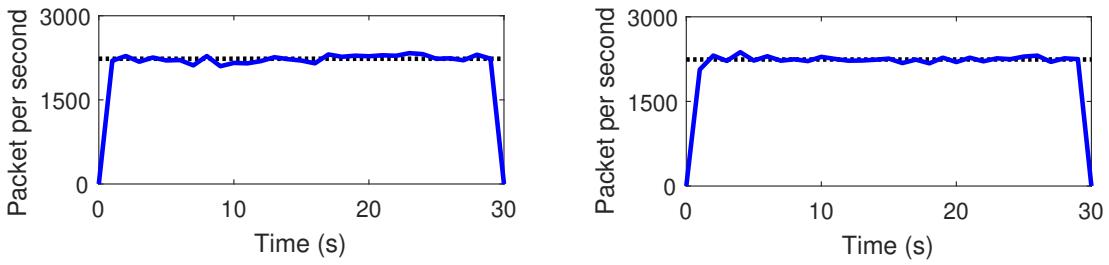
be defined in the following. *iii) EVM of the decoded signals at SU 2:* It is defined as follows:

$$\mathrm{EVM} = 10\log_{10}\left(\frac{\mathbb{E}\left[\left|\hat{X}_{\mathrm{s}}^{[2]}(l,k) - X_{\mathrm{s}}^{[2]}(l,k)\right|^2\right]}{\mathbb{E}\left[\left|X_{\mathrm{s}}^{[2]}(l,k)\right|^2\right]}\right). \tag{2.29}$$

iv) Throughput of secondary and primary networks: The throughput of the primary and secondary networks are extrapolated based on the measured EVM at SU 2 and PU 2, respectively. To calculate throughput, we use

$$r = \frac{N_{\mathrm{sc}}}{N_{\mathrm{fft}} + N_{\mathrm{cp}}} \cdot b \cdot \eta_t \cdot \gamma\left(\mathrm{EVM}\right), \tag{2.30}$$

where N_{sc}, N_{fft}, and N_{cp} denote number of used subcarriers, FFT points, and the length of cyclic prefix, respectively. b is the sampling rate in MSps. η_t is the portion of available airtime being used for signal transmissions. $\gamma(\mathrm{EVM})$ is the average number of bits carried by one subcarrier. This parameter is specified in Table 2.2 and Table 2.3 for WiFi-like PHY and LTE-like PHY, respectively.



(a) Interference-free scenario.

(b) Spectrum sharing scenario.

Figure 2.10: Packet delivery rate of the primary network in interference-free and spectrum sharing scenarios.

2.7.3 Coexistence with Commercial Wi-Fi Devices

We first consider primary network 1 in Table 2.1. The two Wi-Fi devices (Alfa 802.11n dongles with Atheros Chipset) in this primary network are connected in the ad-hoc mode, and they send data packets to each other as shown in Fig. 2.3. These two devices are placed at the spots marked by blue squares in Fig. 2.9. The secondary network is also specified in Table 2.1. Each secondary device is equipped with two antennas. We place the two secondary devices at location 1 in Fig. 2.9(a).

Primary Network. We first study the performance of the primary devices with and without spectrum sharing. Fig. 2.10(a) shows the measured packet delivery rate between the two primary devices in the absence of secondary devices (i.e., the secondary devices are turned off). Fig. 2.10(b) shows the measured result when the secondary devices conduct their transmissions in Phase II (see Fig. 2.8(b)). It can be seen that, in both cases, the primary network achieves almost the same packet delivery rate. This indicates that the primary network is almost not affected by the secondary network.

How is the interference from the secondary transmitter handled? Is it because of the BBF on the secondary transmitter? To answer these questions, we conduct another experiment. When both

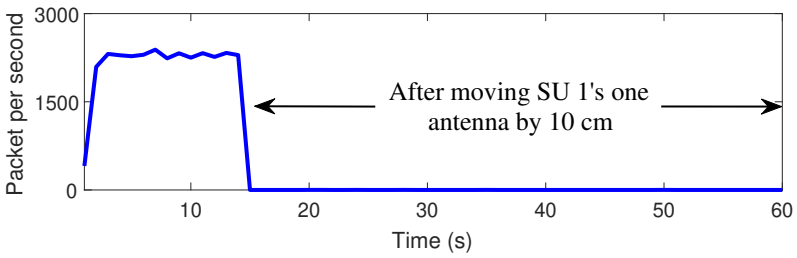
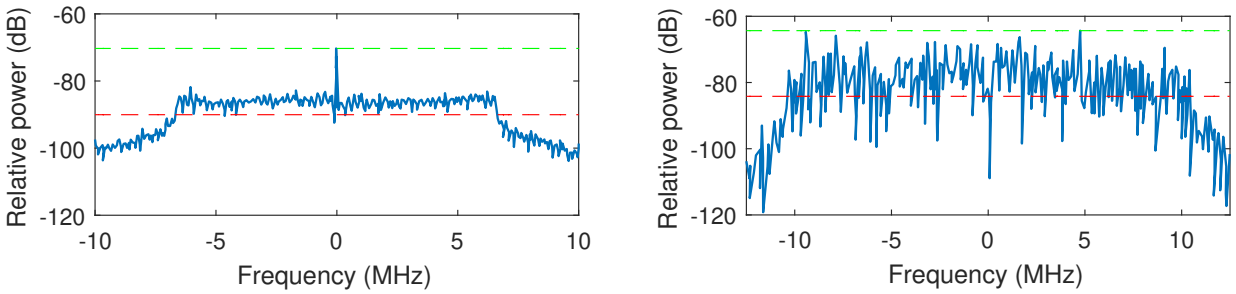


Figure 2.11: Packet delivery rate of the primary network before and after moving SU 1's one antenna by 10 cm.



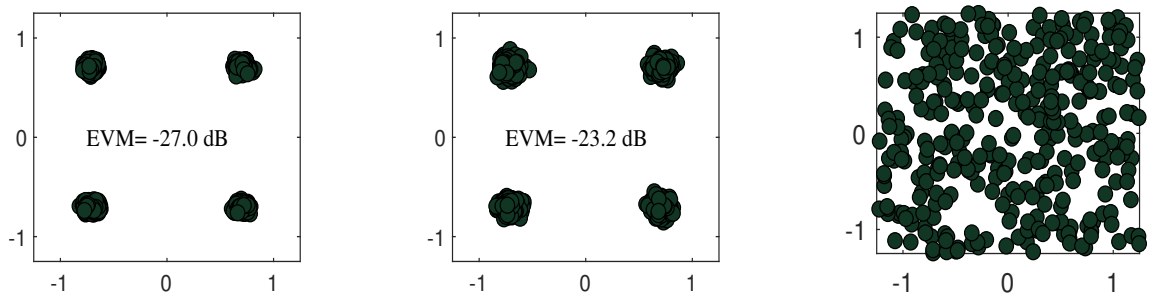
(a) Received interference from PU 1.

(b) Received signal from SU 1.

Figure 2.12: Relative power spectral density of the received signal and interference at the secondary receiver's first antenna.

primary and secondary networks are transmitting, we move one of the secondary transmitter's antennas about 10 cm. Fig. 2.11 shows the packet delivery rate of the primary network before and after the antenna movement. We can see that the movement of SU 1's one antenna results in a steep drop of primary network's packet delivery rate. This indicates that it is SU 1's BBF that mitigates the interference for PU 2.

Secondary Network. We now shift our focus to the secondary network. We first check the strength of signal and interference at the secondary receiver. Fig. 2.12 shows the measured results on one of SU 2's antennas. We can see that the signal and interference at the secondary receiver are at the similar level. This observation also holds for the another antenna. We then check the performance of the secondary receiver in the presence of interference from the primary transmitter. To do so, we conduct three experiments: i) interference-free transmission of the secondary network (secondary devices only, no primary devices); ii) spectrum-sharing transmission with SU 2 using



(a) Interference-free. (b) BIC-spectrum sharing. (c) ZF- spectrum sharing.

Figure 2.13: Constellation diagram of the decoded signals at the secondary receiver (SU 2) in three different experiments.

our proposed BIC; and iii) spectrum-sharing transmission with SU 2 using ZF signal detection. The measured results are presented in Fig. 2.13. It is clear to see that, with the aid of BIC, the secondary receiver can successfully decode its desired signals. Compared to the interference-free scenario, the EVM degradation is about 3.8 dB. The conventional ZF signal detection method cannot decode the signal in the presence of interference. This shows the effectiveness of our proposed BIC technique. A demo video of our real-time spectrum sharing scheme can be found in [131].

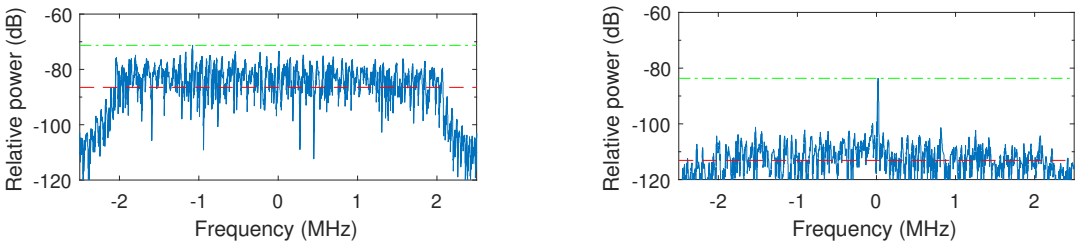
2.7.4 Network Setting: $(M_p = 1, M_s = 2)$

We now consider the CRN in Fig. 2.8 when the primary devices have one antenna ($M_p = 1$) and the secondary devices have two antennas ($M_s = 2$). Primary networks 2 and 3 specified in Table 2.1 are used in our experiments.

2.7.4.1 A Case Study

As a case study, we use primary network 3 (CDMA-like) in Table 2.1 and place the secondary devices at location 1 to examine the proposed spectrum sharing scheme.

Tx-Side IC Capability. We first want to quantify the tx-side IC capability at the secondary



(a) SU 1 uses $[\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}}]$ as precoder.

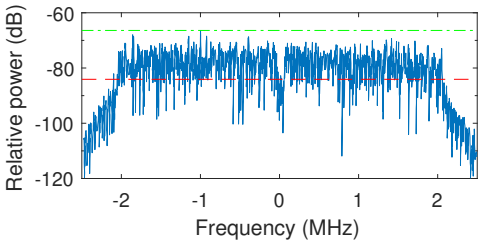
(b) SU 1 uses our BBF technique.

Figure 2.14: Relative power spectral density of PU 2's received interference from two-antenna SU 1 in two cases.

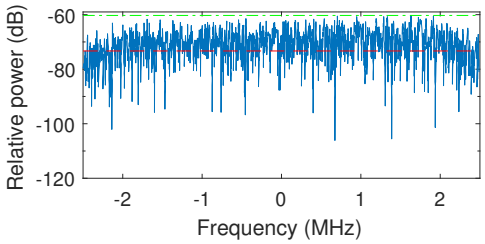
transmitter (SU 1) from its BBF. To do so, we conduct the following experiments. We turn off the primary transmitter (PU 1) and measure the received interference at the primary receiver (PU 2) in two cases: (i) using $[\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}}]$ as the precoder; and (ii) using our proposed beamforming precoder in (2.7) and (2.8) with $\alpha_1 = 1$. Fig. 2.14 presents our experimental results. We can see that, in the first case, the relative power spectral density of PU 2's received interference is about -87 dB. In the second case, the relative power spectral density of PU 2's received interference is about -113 dB. Comparing these two cases, we can see that the tx-side IC capability from BBF is about $113 - 87 = 26$ dB. We note that, based on our observations, the relative power spectral density of the noise at PU 2 is in the range of -120 dB to -110 dB. Therefore, thanks to BBF, the interference from the secondary transmitter to the primary receiver is at the noise level.

Rx-Side IC Capability, EVM, and Data Rate. We now study the performance of the secondary receiver (SU 2). First, we measure SIR at SU 2. Fig. 2.15 shows our measured results on SU 2's first antenna. We can see that the relative power spectral density of its received signal and interference is -83 dB and -73 dB, respectively. This indicates that the SIR on SU 2's first antenna is -10 dB (assuming that noise is negligible). Using the same method, we measured that the SIR on SU 2's second antenna is -12 dB.

We measure the EVM of SU 2's decoded signals in the presence of interference. Fig. 2.16(a–b) present the constellation of the decoded signals at SU 2. It is evident that SU 2 can decode both

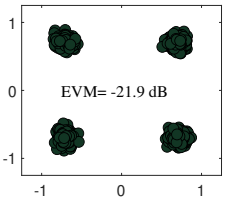


(a) SU 2's received signal on its first antenna.

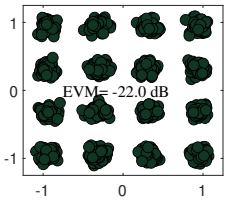


(b) SU 2's received interference on its first antenna.

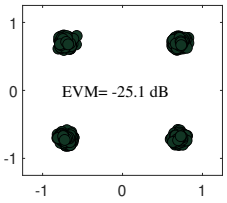
Figure 2.15: Relative power spectral density of SU 2's received signal and interference on its first antenna.



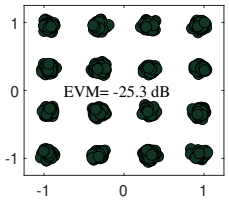
(a) QPSK signals-spectrum sharing.



(b) 16QAM signal-spectrum sharing.



(c) QPSK signal-interference-free.

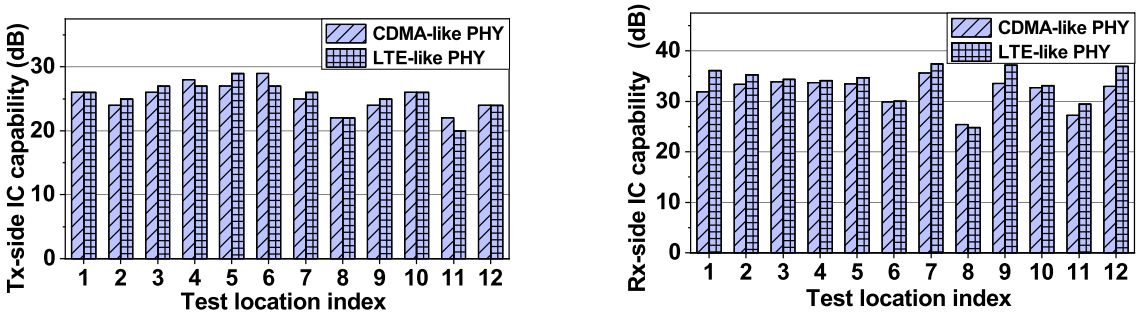


(d) 16QAM signal-interference-free.

Figure 2.16: Constellation diagram of decoded signals at SU 2: our spectrum sharing scheme versus interference-free scenario.

QPSK and 16QAM signals from SU 1 in the presence of interference from PU 1. The EVM is -21.9 dB when QPSK is used for the secondary network and -22 dB when 16QAM is used for the secondary network. As a benchmark, Fig. 2.16(c–d) present the experimental results when there is no interference from PU 1. Comparing Fig. 2.16(a–b) with Fig. 2.16(c–d), we can see that SU 2 can effectively cancel the interference from PU 1.

Finally, we calculate SU 2's IC capability and throughput. Based on the SIR on SU 2's antennas and the EVM of its decoded signals, SU 2's IC capability is $10 + 21.9 = 31.9$ dB in this case. Based on (2.30) and the measured EVM, the throughput (data rate) of secondary network is extrapolated to be 4.5 Mbps.



(a) Tx-side IC capability from the secondary transmitter's BBF.

(b) Rx-side IC capability from the secondary receiver's BIC.

Figure 2.17: Tx-side and rx-side IC capabilities of the secondary network for $\left(M_p = 1, M_s = 2\right)$ setting.

2.7.4.2 Experimental Results at all Locations

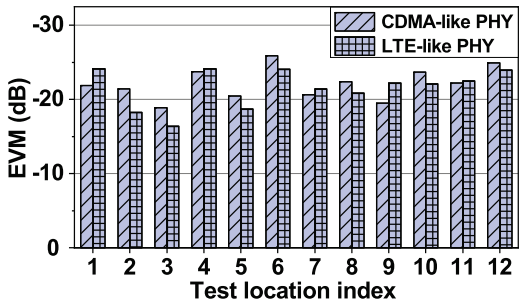
We now extend our experiments from one location to all the 12 locations and present the measured results as follows.

Tx-Side IC Capability. Fig. 2.17(a) presents the tx-side IC capability of the two-antenna secondary transmitter (SU 1). We can see that the secondary transmitter achieves a minimum of 20.0 dB and an average of 25.3 dB IC capability across all the 12 locations.

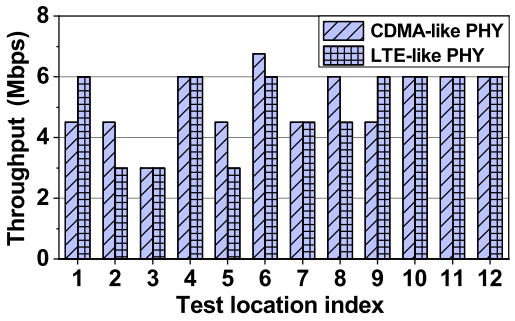
Rx-Side IC Capability. Fig. 2.17(b) presents the rx-side IC capability of the two-antenna secondary receiver. We can see that the secondary receiver achieves a minimum of 25.0 dB, a maximum of 38.0 dB, and an average of 32.8 dB IC capability across all the 12 locations, regardless of the PHY used for the primary network.

Rx-Side EVM. Fig. 2.18(a) presents the EVM of the decoded signals at the two-antenna secondary receiver in the presence of interference from the primary transmitter. We can see that in all the locations, although the EVM varies, the EVM achieves an average of -21.8 dB, regardless of the PHY used for the primary network.

Throughput of Secondary Network. Based on the measured EVM at the secondary receiver, we extrapolate the achievable data rate in the secondary network using (2.30). Fig. 2.18(b) presents



(a) EVM of the decoded signals at the secondary receiver.



(b) Throughput of the secondary network.

Figure 2.18: Performance of the secondary network in the proposed spectrum sharing scheme for $\big(M_p = 1, M_s = 2\big)$ setting.

the results. As we can see, the secondary network achieves a minimum of 3.0 Mbps data rate, a maximum of 6.7 Mbps, and an average of 5.1 Mbps across all the 12 locations. Note that this data rate is achieved by the secondary network in 5 MHz bandwidth, and the secondary transmitter's power is controlled so that its interference at the primary receiver (after BBF) remains at the noise level.

2.7.4.3 BBF versus Other Beamforming Techniques

As BBF is the core component of our spectrum sharing scheme, we would like to further examine its performance by comparing it against the following two beamforming techniques.

- *Explicit Beamforming (EBF)*: In this technique, the secondary transmitter (SU 1) has knowledge of *forward* channel between itself and the primary receiver (PU 2), i.e., $\mathbf{H}_{\text{sp}}^{[1]}(k)$. The forward channel knowledge is obtained through explicit channel feedback. Specifically, SU 1 sends a Null Data Packet (NDP) to PU 2, which estimates the channel and feed the estimated channel information back to SU 1. After obtaining the forward channel $\mathbf{H}_{\text{sp}}^{[1]}(k)$, SU 1 constructs the precoder by $\mathbf{P}(k) = mineigvectors(\mathbf{H}_{\text{sp}}^{[1]}(k))$, where k is subcarrier index.

- *Implicit Beamforming (IBF)*: In this technique, the secondary transmitter (SU 1) has knowl-

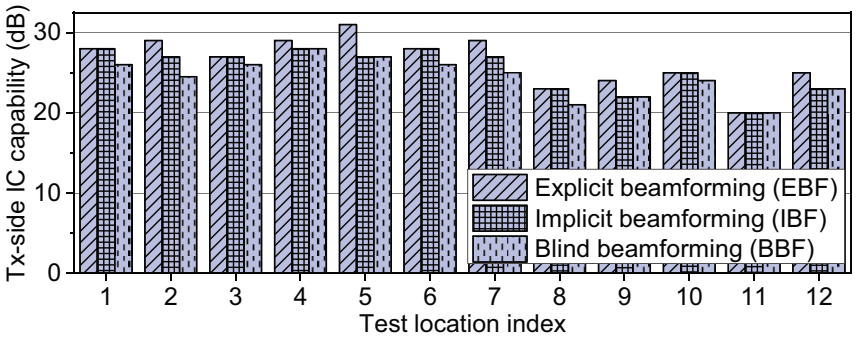


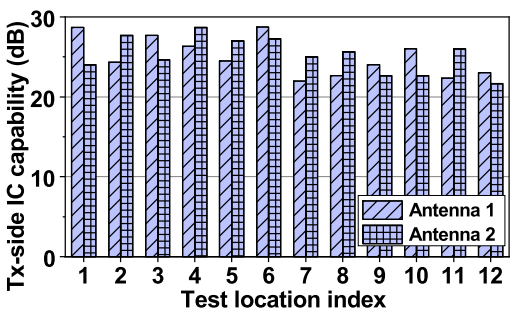
Figure 2.19: Tx-side IC capability of the three beamforming techniques when the secondary device has three antennas.

edge of *backward* channel from the primary receiver (PU 2) to itself, i.e., $\mathbf{H}_{\text{ps}}^{[1]}(k)$. The backward channel knowledge is obtained through implicit channel feedback. Specifically, PU 2 sends an NDP to SU 1. SU 1 first estimates the backward channel $\mathbf{H}_{\text{ps}}^{[1]}(k)$. It then constructs the precoder by $\mathbf{P}(k) = mineigvectors(\mathbf{H}_{\text{ps}}^{[1]}(k))$, where k is subcarrier index. Channel calibration has been performed at SU 1 before signal transmission.

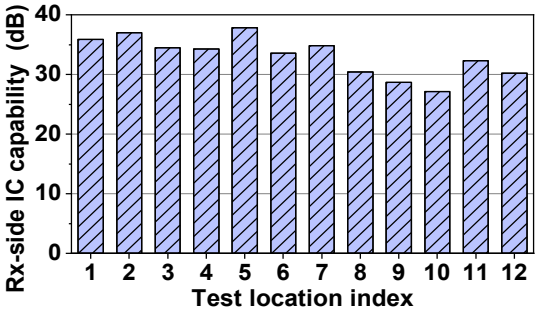
We conduct experiments to measure the tx-side IC capability of these three beamforming techniques. Fig. 2.19 depicts our results. We can see that, compared to EBF, our proposed BBF has a maximum of 4.5 dB and an average of 2.1 dB degradation. Compared to IBF, our proposed BBF has a maximum of 2.5 dB and an average of 1.0 dB degradation. The results show that the proposed BBF has competitive performance compared to EBF and IBF. We note that, although offering better performance, EBF and IBF cannot be used in underlay CRNs as they require knowledge and cooperation from the primary devices.

2.7.5 Network Setting: $(M_p = 2, M_s = 3)$

We now study the CRN in Fig. 2.8 when the primary devices have two antennas and the secondary devices have three antennas (i.e., $M_p = 2$ and $M_s = 3$). The primary devices use their two



(a) Tx-side IC capability from the secondary transmitter's BBF.



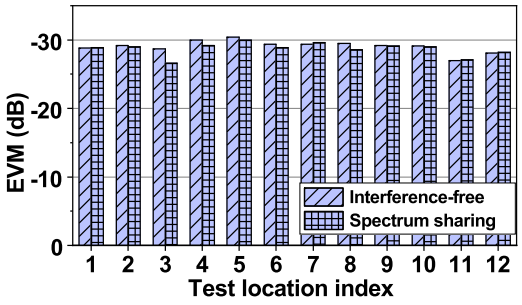
(b) Rx-side IC capability from the secondary receiver's BIC.

Figure 2.20: Tx-side and rx-side IC capabilities of the secondary network for $\left(M_p = 2, M_s = 3\right)$ setting.

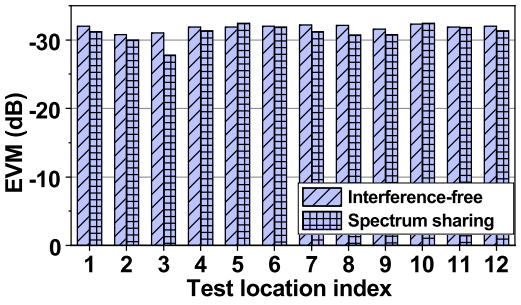
antennas for spatial multiplexing. That is, two independent data streams are transfered in the primary network. The secondary devices use their spatial DoF provided by their three antennas for both interference management and signal transmission. Indeed, one data stream is transfered in the secondary network. The primary network uses LTE-like PHY (see primary network 2 in Table 2.1) for data transmission. We study our spectrum sharing scheme in this CRN and report the measured results below.

Tx-Side IC Capability. In this CRN, since the primary receiver has two antennas, the secondary transmitter needs to cancel its generated interference for both antennas on the primary receiver. We measure the IC capability of our proposed BBF for the primary receiver's both antennas. Fig 2.20(a) exhibits our measured results. We can see that a three-antenna secondary transmitter can effectively cancel the interference on the primary receiver's both antennas. Specifically, the BBF on the secondary transmitter achieves a minimum of 21.7 dB, a maximum of 28.7 dB, and an average of 25.1 dB IC capability for the primary receiver's two antennas.

Rx-Side IC Capability. In this CRN, since the primary transmitter sends two independent data streams, the secondary receiver needs to decode its desired signals in the presence of two interference sources. We measure the rx-side IC capability of our proposed BIC at the three-antenna



(a) EVM of the decoded data stream 1 at the primary receiver.



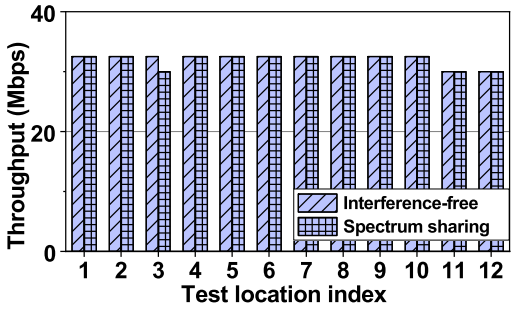
(b) EVM of the decoded data stream 2 at the primary receiver.

Figure 2.21: EVM of the two data streams in the primary network with and without the secondary network for $\left(M_p = 2, M_s = 3\right)$ setting.

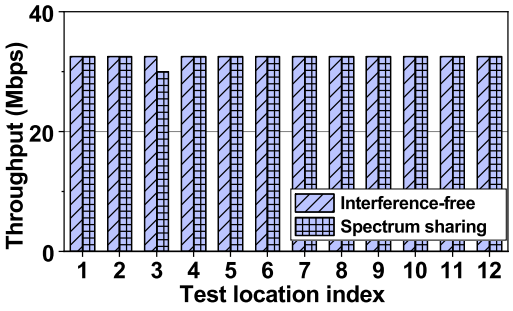
secondary receiver. Fig 2.20(b) exhibits our measured results. We can see that the proposed BIC on the secondary receiver achieves a minimum of 26.5 dB, a maximum of 38.1 dB, and an average of 33.0 dB IC capability over the 12 locations. This shows the effectiveness of the proposed BIC in handling unknown interference.

EVM at Primary Receiver. We now study the performance of the two data streams in the primary network. We want to see if the presence of secondary network harmfully affects the traffic in the primary network. To do so, we measure the EVM of the decoded two data streams at the primary receiver in two cases: i) in the presence of the secondary network, and ii) in the absence of the secondary network. Fig. 2.21 presents our measured results. It can be seen that the presence of the secondary network does not visibly affect the EVM performance of the primary network. This indicates that the BBF at the secondary network successfully mitigates the interference from the secondary transmitter to the primary receiver.

Throughput of Primary Network. Based on the measured EVM at the primary receiver, we extrapolate the achievable data rate on each data stream of the primary network using (2.30). The extrapolated throughput is presented in Fig. 2.22. Referring to Fig. 2.22(a), the primary network achieves an average of 32.1 Mbps throughput for its stream 1 in interference-free case and an average of 31.9 Mbps throughput in coexistence with the secondary network. As shown in

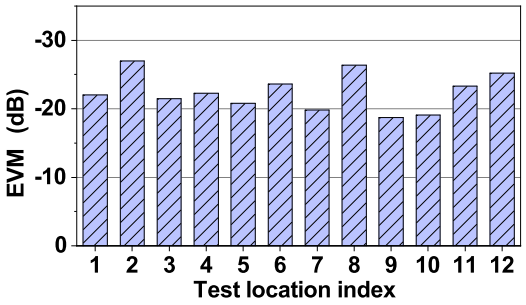


(a) Extrapolated throughput of decoded primary data stream 1.

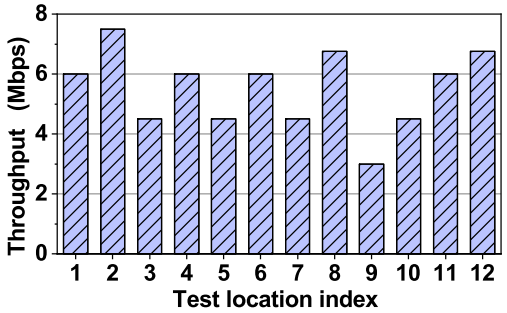


(b) Extrapolated throughput of decoded primary data stream 2.

Figure 2.22: Throughput of the two data streams in the primary network with and without the secondary network for $\left(M_p = 2, M_s = 3\right)$ setting.



(a) EVM of decoded signals at the secondary receiver.



(b) Throughput of the secondary network.

Figure 2.23: Performance of the secondary network in the proposed spectrum sharing scheme for $\left(M_p = 2, M_s = 3\right)$ setting.

Fig. 2.22(b), for its data stream 2, the primary network achieves 32.5 Mbps and 32.3 Mbps throughput on average in the interference-free and spectrum sharing scenarios, respectively. For both data streams, only 0.2 Mbps degradation is observed in the throughput of the primary network.

EVM at Secondary Receiver. Having confirmed that the spectrum utilization of secondary network does not degrade the performance of primary network, we now study the achievable performance of the secondary network. Recall that we transfer one data stream in the secondary network. We measure EVM of the decoded signal at the secondary receiver. Fig. 2.23(a) depicts the measured results. We can see that the EVM at the secondary receiver achieves a minimum of -27.7 dB, a maximum of -18.2 dB, and an average of -22.5 dB over the 12 locations.

Throughput of Secondary Network. Based on the measured EVM at the secondary receiver,

we extrapolate the achievable data rate of the secondary network using (2.30). The extrapolated data rate is presented in Fig. 2.23(b). We can see that the proposed spectrum sharing scheme achieves a minimum of 3.0 Mbps, a maximum of 7.5 Mbps, and an average of 5.5 Mbps over the 12 locations. Note that this data rate is achieved by the secondary network in 5 MHz and without harmfully affecting the primary network.

2.7.6 Summary of Observations

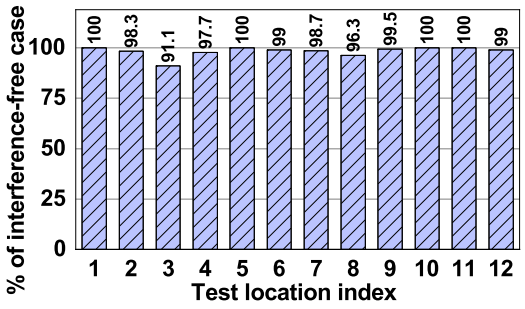
We now summarize the observations from our experimental results as follows:

- *BBF:* BBF demonstrates its capability of handling cross-network interference in CRNs where the secondary network has no knowledge about the primary network. In $\left(M_p=1, M_s=2\right)$ network setting, BBF achieves an average of 25.3 dB IC capability. In $\left(M_p=2, M_s=3\right)$ network setting, BBF achieves an average of 25.1 dB IC capability.

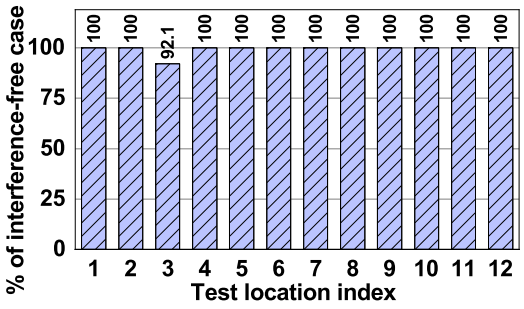
- *BIC:* BIC also demonstrates its capability of decoding the desired signals in the presence of unknown interference. In $\left(M_p=1, M_s=2\right)$ network setting, it achieves an average of 32.8 dB IC capability. In $\left(M_p=2, M_s=3\right)$ network setting, it achieves an average of 33.0 dB IC capability.

- *Primary Network:* The primary network has very small performance degradation when the secondary network shares the spectrum (compared to the case without secondary network). As shown in Fig. 2.24(a), the average EVM degradation at the primary receiver is 1.6% over the 12 locations. Also, as shown in Fig. 2.24(b), the average throughput degradation at the primary receiver is 0.7% over the 12 locations.

- *Secondary Network:* Using BBF at its transmitter and BIC at its receiver, the secondary net-



(a) Measured EVM of primary data streams.



(b) Extrapolated throughput of primary data streams.

Figure 2.24: Performance of the proposed spectrum sharing scheme w.r.t. interference-free case for $\left(M_p = 2, M_s = 3\right)$ setting.

work intends to establish communications by sharing the spectrum with the primary network. The secondary network achieves 1.0 bits/s/Hz in the CRN with $\left(M_p = 1, M_s = 2\right)$ network setting and 1.1 bits/s/Hz in the CRN with $\left(M_p = 2, M_s = 3\right)$ network setting.

2.8 Limitations and Discussions

While the proposed scheme demonstrates its potential in real-world networks, there are still some issues that remain open and need to be addressed prior to its real applications.

Primary Traffic Directions. In our spectrum sharing scheme, we assume that the primary communications are bidirectional and that the pattern of primary traffic is consistent. Under such assumptions, duration and direction of primary traffic are easy to learn for beamforming filter design. In real systems, the pattern of primary traffic might not be consistent. In such a case, a sophisticated learning algorithm is needed for the secondary devices to differentiate the forward and backward transmissions of the primary network.

Channel Coherence Time. In static networks (e.g., indoor Wi-Fi), the devices are stationary or moving at a low speed. Then, the channel coherence time is large enough to cover the entire

period of primary forward transmission. But in the dynamic networks with highly mobile devices, the channel coherence time may be smaller than the duration of primary forward transmission. In such a case, the secondary network cannot use entire airtime of primary forward transmission. Instead, it can only access the spectrum when its beamforming filters remain valid (i.e., within the channel coherence time).

Extension to Large-Scale Networks. In this work, we presented a spectrum sharing scheme for a small-size CRN consisting of one PU pair and one SU pair. This spectrum sharing scheme can be extended to a large-scale CRN that comprises multiple PU pairs and multiple SU pairs. This is because in most real-world wireless networks (e.g., Wi-Fi and cellular), only one user pair is active on a frequency band at a time. Therefore, our current design is a fundamental building block for spectrum sharing in a large-scale CRN. Nevertheless, extending our design to a large-scale CRN still faces several challenges. First, a secondary device should be capable of learning the active PU devices over time as well as their transmission direction and duration. For a secondary device, how to accurately obtain this information through a learning procedure is a challenging task. Second, primary devices may not be stationary (e.g., vehicular and unmanned aerial networks). How to design an adaptive and intelligent spectrum sharing MAC protocol for the secondary network is another challenging task.

2.9 Chapter Summary

In this chapter, we proposed a spectrum sharing scheme for an underlay CRN that comprises two primary users and two secondary users. The proposed scheme allows the secondary users to use the spectrum without affecting the throughput of the primary users. The key components of our scheme are two MIMO-based IC techniques: BBF and BIC. BBF enables the secondary transmitter to pre-

cancel its generated interference for the primary receiver. BIC enables the secondary receiver to decode its desired signals in the presence of unknown cross-network interference. These two IC techniques make it possible for the secondary users to access the spectrum while remaining transparent to the primary users. We have built a prototype of our spectrum sharing scheme on a wireless testbed. We demonstrated that our prototyped secondary devices can coexist with commercial Wi-Fi devices. Extensive experimental results show that, for a secondary user with two or three antennas, BBF and BIC achieve about 25 dB and 33 dB IC capabilities in real wireless environments, respectively.

Chapter 3

D2D Communications in Cellular Networks

3.1 Introduction

Cellular networks are key components of the telecommunications infrastructure in our society. Their roles of providing ubiquitous wireless Internet services become increasingly important with the proliferation of Internet-based applications such as smart cities, IoT, and autonomous driving. To increase the network capacity, provide massive connectivity, and meet the growing demands for wireless services, many advanced wireless technologies have been proposed for next-generation cellular networks. MU-MIMO, which allows a multi-antenna BS to simultaneously serve multiple User Equipment (UEs) on the same spectrum band, is one of the pivotal technologies for cellular networks [102]. As its benefits are well recognized, MU-MIMO has already been deployed in real cellular networks to harness its throughput gain in the presence of antenna configuration asymmetricity.

D2D communication is another promising technology for cellular networks [204]. Its basic idea is to allow direct communication between two proximity-based mobile users without traversing the BS or core network. As mobile users in today's cellular networks require high data rate services (e.g., video sharing, online gaming, proximity-aware networking) in which they could potentially be in a short range for direct communication, D2D communication can greatly increase the spectral efficiency of the network. Moreover, the advantages of D2D communication go be-

yond spectral efficiency. Saving the airtime at the core network, D2D offers more airtime to the BS that can be leveraged to serve massive number of low-rate devices such as IoT sensors. It also can potentially reduce packet transmission delay, enhance user fairness, offload traffic for BSs, and alleviate congestion for core networks, especially in networks congested by IoT devices [177].

Although there are many results of MU-MIMO and D2D communications, most of them are limited to their respective domains and there is a lack of practical design to harvest the benefits of both technologies in cellular networks. Such a stagnation underscores the critical need for bridging this gap. The main challenge in such a joint design is the interference management between MU-MIMO devices (BS and UEs) and D2D devices. As existing MU-MIMO schemes are vulnerable to interference (e.g., pilot contamination), the performance of MU-MIMO communication will be dramatically degraded by the interference from active D2D devices if the interference is not properly handled. At the same time, the interference from MU-MIMO devices will also disrupt the D2D communications. Therefore, the coexistence of D2D and MU-MIMO communications necessitates a systematic scheme to tame the mutual interference between the two subsystems.

In this chapter, we present DM-COM, a practical scheme for enabling the coexistence of D2D and MU-MIMO communications for cellular networks. We consider a single cell that comprises a BS, a set of cellular UEs (C-UEs), and a pair of D2D UEs (D-UEs) on each Physical Resource Block (PRB). The BS is equipped with several antennas; the C-UEs are equipped with one antenna; and the D-UEs are equipped with one or multiple antennas. MU-MIMO is used for communication between the BS and set of C-UEs. D2D technology is used for communication between the pair of D-UEs. We assume that MU-MIMO communication follows the principles of 5G New Radio (NR) standard (e.g., waveform and frame structure). We also assume that D-UEs know the network protocol and transmission pattern used by MU-MIMO as such information will be broadcast by BS over control channel. We further assume that the D2D applications are sensitive to communication

latency (e.g., virtual reality, online gaming, and health monitoring) and thus require low-delay bidirectional transmissions. In such a network, our objective is to enable the concurrent spectrum utilization of MU-MIMO and D2D communications.

Towards this objective, we employ a blend of two interference management techniques: interference cancellation and beamforming, which are used in the following way. In the uplink MU-MIMO, the BS receives both desired signals from C-UEs and interfering signals from D-UEs. To decode its desired signals, the BS leverages the spatial DoF provided by its multiple antennas and constructs a decoding matrix to cancel the interference and equalize the channel distortion. In the downlink MU-MIMO, the BS constructs a beamforming (precoding) matrix to send its intended signals to C-UEs while pre-cancelling the interference for the receiving D-UEs. The C-UEs do not participate in the interference management and, instead, they rely on other devices to handle their interference. A similar approach is adopted to manage the interference in the D2D subsystem.

While the idea of our interference management scheme is clear, many technical issues remain challenging. For uplink MU-MIMO transmission, how can the BS decode the signals from C-UEs in the presence of interference from D-UEs? For downlink MU-MIMO transmission, how can the BS perform beamforming in the downlink so it can mute its interference for the D-UEs? For these two questions, one possible solution is to design a dedicated channel acquisition protocol for the BS to obtain CSI for signal detection and beamforming. However, such a solution not only entails a large airtime overhead but also complicates the system operation. In light of this, we propose a new MU-MIMO scheme that is resilient to the interference from/to D-UEs. The key idea of our new scheme is that, instead of relying on CSI for signal detection and beamforming, we blindly use the received signals to extract spatial information required to train decoding and beamforming matrices. Surprisingly, such a scheme leads to a very good performance for signal detection in the face of interference, provided that the BS has sufficient antennas.

For D2D communication, we apply the same approach to managing interference. For a transmitting D-UE, it leverages the overheard interference from C-UEs to construct the precoding matrix for beamforming. For a receiving D-UE, it leverages the reference signals to construct the decoding matrix for signal detection in the presence of interference. By doing so, the D-UEs do not require CSI for signal detection and beamforming. Therefore, the need for notorious channel feedback is eliminated.

Based on the above interference management scheme, we have developed DM-COM to enable the coexistence of D2D and MU-MIMO communications in cellular networks. In a nutshell, DM-COM advances the state-of-the-art in the following aspects:

- At the cellular BS, we have designed an interference management technique that cancels interference from/to D2D users at uplink/downlink. This scheme does not need CSI nor synchronization with D2D users.
- At the D2D users, we have designed an interference management technique that cancels interference from/to cellular nodes. This scheme does not need CSI nor synchronization with cellular subsystem.
- We have proposed DM-COM, a holistic scheme to enable coexistence of D2D and MU-MIMO technologies without adversely affecting each other.
- We have built a prototype of DM-COM on a wireless testbed consisting of USRP N210 devices and shown DM-COM's efficacy in handling cross-subsystem interference in real-world wireless environment.

We evaluated the performance of DM-COM in a pico-cell network where a four-antenna BS serves two single-antenna C-UEs in accordance with 5G NR standard. In the network, there coexists a pair of D-UEs for direct communication. One D-UE has one antenna and the other has

three antennas. Our experimental results show that DM-COM reaches 1.9 bit/s/Hz spectral efficiency for D2D users. This is achieved at the cost of 8.0% throughput degradation for MU-MIMO users (compared to the case without D2D users). Moreover, compared to the conventional case where all the users (C-UEs and D-UEs) are served by the BS, DM-COM improves the average network throughput from 21.9 Mbps to 35.1 Mbps in 5 MHz bandwidth, i.e., 60.3% throughput gain for DM-COM is observed. Our experimental results show that DM-COM successfully re-uses the spectrum that is pre-occupied by C-UEs. DM-COM maintains the performance of incumbent C-UEs and increases the overall network throughput through establishing D2D communications.

3.2 Related Work

We briefly review D2D and MU-MIMO solutions in cellular networks.

D2D. To accommodate ever-increasing users in cellular networks and enhance the spectrum re-utilization, D2D users are allowed to communicate directly without involvement of the BS. Despite its potential benefits, a D2D sub-system needs to control co-channel interference, manage resources for competing users, and mitigate security threats [177]. In order for accomplishing these tasks, the enablers of D2D communications include beamforming [115,165,176], spectral resource management [30, 84, 130, 160], power control [9, 10, 61, 62, 98, 160, 174], and mode selection [15, 27, 60]. The existing research follows different objectives, such as achievable data rate [9, 10, 15, 27, 160], fairness [98], interference minimization [61], energy efficiency [62, 84, 130, 165, 174], and security of D2D systems [112, 148, 158]. From another perspective, most of existing works consider spectrum re-utilization in either uplink (see, e.g., [10, 61, 98]) or downlink (see, e.g., [115, 165, 176]) of cellular networks, but not both.

Moreover, most of the existing works require perfect global channel knowledge as well as

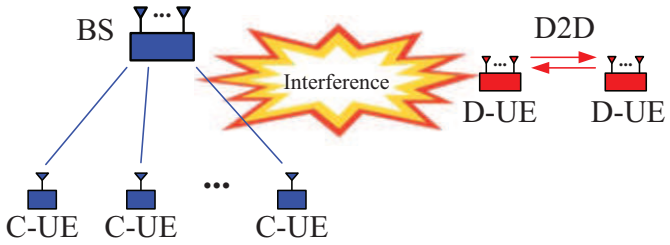


Figure 3.1: Coexisting MU-MIMO and D2D communications over one PRB in a cellular network.

network-wide synchronization. In contrast, DM-COM enables spectrum re-utilization in both uplink and downlink. It does not require channel feedback between the network devices, nor network-wide fine-grained synchronization.

MU-MIMO. MU-MIMO has widely been employed in current wireless systems. The main components of MU-MIMO are beamforming in the downlink and multi-user detection in the uplink. Most of beamforming methods are reliant on perfect CSI [34, 122, 163]. In the uplink, blind beamforming methods offer a solution to this challenge, but suffer from high computational complexity and long processing delays since they need to solve a complex optimization problem [6, 24, 173] or follow sophisticated procedures to learn spatial information [12, 125, 126]. In the downlink, existing signal detection methods consider benign environments where the network nodes are perfectly synchronized [22, 42, 183, 201]. DM-COM differs from existing methods as it eliminates the need for channel feedback and network-wide synchronization in both downlink and uplink.

3.3 Problem Description

Network Setting. We consider a cellular network as shown in Fig. 3.1. It comprises a BS, a set of C-UEs, and a pair of D-UEs on one PRB. The BS has multiple antennas, and each C-UE has

a single antenna.[1] Let M_{bs} denote the number of the BS's antennas. Let N denote the number of C-UEs. To fully utilize the BS's antennas and maximize the spectral efficiency, MU-MIMO is used for the communication between the BS and N C-UEs. The BS coordinates uplink and downlink transmissions with Time Division Multiple Access (TDMA) to serve cellular users. Within the cellular network, there coexists a pair of D-UEs intending to conduct bi-directional communication over a PRB without traversing the BS. without any loss of generality, in the remainder of this chapter, we focus on one pair of D-UEs over a PRB. All the arguments hold for multiple pair of D-UEs, each of which exclusively work over one or multiple PRBs. In the D2D pair under consideration, the two D-UEs may have different numbers of antennas. Let M_{d1} and M_{d2} denote the number of D-UE 1's and D-UE 2's antennas, respectively. Without loss of generality, we also assume that the number of D-UE 1's antennas is less than or equal to the number of D-UE 2's antennas, i.e., $M_{\mathrm{d1}} \leq M_{\mathrm{d2}}$. For such a network, we have the following assumptions and justifications:

- We assume that the user selection for MU-MIMO and D2D has taken place. User selection is not within the scope of this work. In real networks, there may exist multiple pairs of D-UEs. In that case, different pairs of D-UEs can be assigned to different PRBs based on some criteria. So, focusing on one pair is sufficient to study the coexistence problem which is indeed the main objective of this chapter.

- We assume that the BS has more antennas than C-UEs, i.e., $M_{\mathrm{bs}} > N$. This assumption can be fulfilled through user selection algorithms. Under this assumption, in addition to decoding the N desired data streams from C-UEs, the BS has remaining spatial DoF provided by its antennas to cancel interference from/to D-UEs.

[1] DM-COM can support the case where C-UE has multiple antennas by simply treating a multi-antenna C-UE as multiple single-antenna C-UEs.

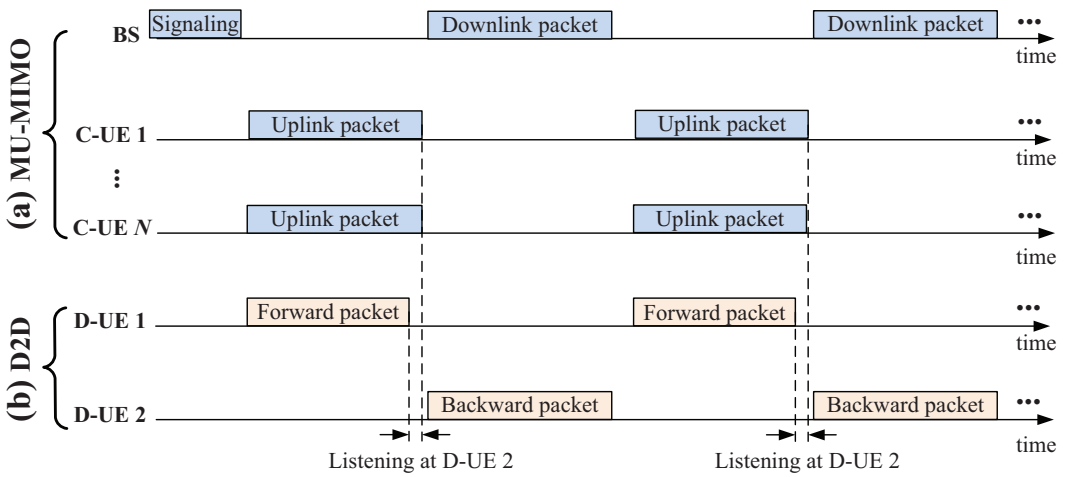


Figure 3.2: The proposed network protocol for coexisting MU-MIMO and D2D communications.

- We assume that the D-UEs know the data flow pattern of MU-MIMO communication indicated by slot format in NR. We also assume that C-UEs are oblivious to D-UEs. C-UEs will not contribute to the interference management.

- We assume that the channel coherence time is sufficient (e.g., 1 ms). The same assumption has been made by other beamforming-based MIMO systems [34, 122, 163].

Our Objective. We aim to develop DM-COM, a practical scheme to enable the coexistence (concurrent spectrum utilization) of D2D and MU-MIMO communications by taming their mutual interference. More specifically, we aim to maximize the throughput of the D2D communication while maintaining the performance of MU-MIMO subsystem.

3.4 DM-COM: An Overview

In this section, we first present a network protocol for the concurrent spectrum utilization of coexisting MU-MIMO and D2D subsystems, and then analyze the achievable data streams on the D2D link. Finally, we point out the underlying challenges in interference management at the physical layer and outline our solutions.

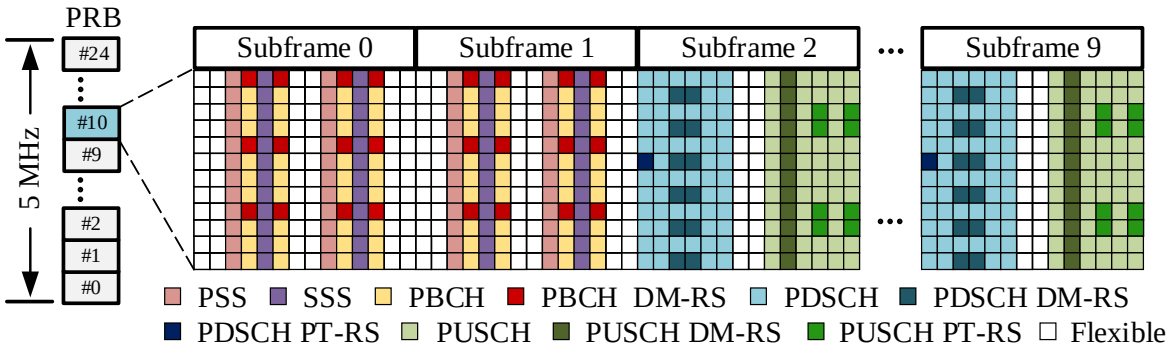


Figure 3.3: Frame structure for MU-MIMO communication.

3.4.1 Network Protocols

MU-MIMO Communication. In the context of cellular networks, Fig. 3.2(a) presents our proposed protocol for uplink and downlink MU-MIMO transmissions. The protocol works as follows. The BS first broadcasts an announcement about MU-MIMO transmission to the selected C-UEs. Then, the selected C-UEs send their packets to the BS in the uplink, which is followed by spatial multiplexing in downlink transmissions. The uplink and downlink transmissions repeat until the session of MU-MIMO communication terminates.

To support MU-MIMO communication, we consider NR-like frame format. Fig. 3.3 depicts the frame structure in one PRB within a frame. To be specific, this frame structure is adopted based on N38 frequency band and slot format 45 setting over 5 MHz [1]. As shown in the figure, the frame is composed of 10 subframes, each of which comprises 14 OFDM symbols according to numerology $\mu = 0$ in NR. Based on the bandwidth configuration, an OFDM symbol has 300 occupied subcarriers grouped into 25 PRBs.

Reference signals are embedded into frames for synchronization, signal demodulation, phase tracking, etc. Among the reference signals shown in Fig. 3.3, We will leverage PDSCH DM-RS of downlink packets and PUSCH DM-RS of uplink packets in our design. As shown in the figure, not every subcarrier has these reference signals. This is because, the subcarrier spacing is small

(15 kHz), and the channels of adjacent subcarriers are highly correlated. Therefore, if a subcarrier does not have reference signal, the reference signals on its adjacent subcarriers can be used for signal demodulation (detection). This feature will also be leveraged in the design of our signal detection method. TDD is considered for MU-MIMO to support its uplink and downlink transmissions. The ratio of uplink and downlink duration can be configured as desired based on the slot format. For ease of demonstration, we have considered slot format 45 with equal downlink/uplink duration, and we equally assigned flexible OFDM symbols to uplink and downlink transmissions.

D2D Communication. Fig. 3.2(b) shows the proposed transmission protocol for the D2D communication, with respect to the timeline of uplink/downlink transmission in MU-MIMO subsystem. In the uplink MU-MIMO, D2D conducts forward transmissions (from D-UE 1 to D-UE 2). In the downlink MU-MIMO, D2D conducts backward transmissions (from D-UE 2 to D-UE 1). To establish such a timing alignment, D2D subsystem needs neither fine-grained synchronization with MU-MIMO subsystem nor coordination from the cellular BS. The D2D subsystem can learn cellular traffic pattern by either listening the information over the control channel or tracking the spatial signatures of signals on multiple antennas on D-UEs. It then adjusts its transmission activities based on learned pattern. As illustrated in the figure, the time duration of D2D forward transmissions is slightly shorter than that of uplink MU-MIMO. In this time period, D-UE 2 overhears the interfering signals from C-UEs, which will be used for the calculation of its beamforming matrix.

For the D2D communication, two remarks are in order. First, the mutual interference between D2D and MU-MIMO communications will be properly handled at physical layer. Therefore, the D2D and MU-MIMO subsystems remain oblivious to each other from the viewpoint of MAC or upper layers. Second, as we shall see later, the interference management at the physical layer does not require PHY-layer cooperation between the D2D and MU-MIMO subsystems. Hence, the D2D

and MU-MIMO subsystems do not need to use the same frame structure and modulation. As we will show via experiments, D2D can employ IEEE 802.11 PHY for its transmissions.

3.4.2 Achievable Data Streams (DoF) on the D2D Link

For the protocol in Fig. 3.2, a natural question to ask is how many data streams can be transported on the D2D link. Apparently, it depends on the number of D-UE 2's antennas. If D-UE 2 has a large number of antennas, then many data streams can be transported on the D2D link, provided that D-UE 1 has enough DoF to support all incoming streams from D-UE 2. If D-UE 2 does not have sufficient antennas, then no data stream can be transported on the D2D link. We note that the number of data streams on an MIMO link, which is also known as DoF, is the first-order approximation of its Shannon capacity with respect to Signal to Noise Ratio (SNR). It also represents the multiplexing gain of the MIMO link in high-SNR regime. Therefore, studying the number of data streams is of great theoretical importance to analyze the achievable data rate of the D2D link (given that analyzing its Shannon capacity is out of our capability). In what follows, we derive the achievable data streams on the D2D link by analyzing the spatial DoF consumption in the uplink and downlink MU-MIMO using an existing DoF model [153].

Assume that the bi-directional transmissions on the D2D link are symmetric, i.e., the number of data streams from D-UE 1 to D-UE 2 is the same as that from D-UE 2 to D-UE 1. We let $d \in \mathbb{N}_0$ denote the number of data streams on the D2D link. To determine the maximum value of d, we first consider the uplink MU-MIMO as shown in Fig. 3.4(a). At the BS, it needs to decode N data streams from C-UEs and cancel d interfering streams from D-UE 1. Based on the DoF model in [153], we have $N + d \leq M_{\mathrm{bs}}$. At D-UE 1, it needs to transmit d data streams. We therefore have $d \leq M_{\mathrm{d1}}$. At D-UE 2, it needs to decode d data streams and cancel N data streams from C-UEs. We have $N + d \leq M_{\mathrm{d2}}$. Based on the above three constraints, the maximum number of achievable

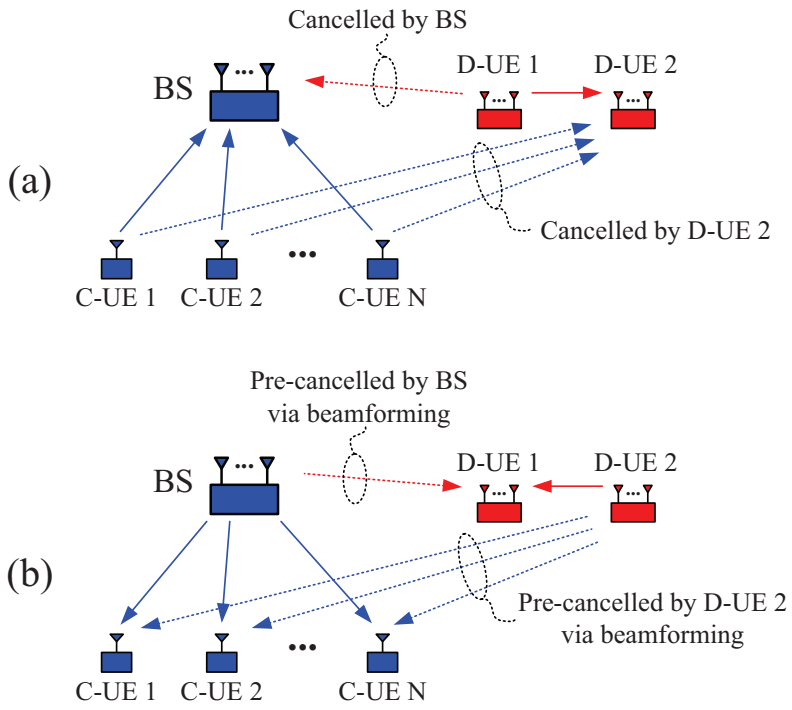


Figure 3.4: Illustrating the mutual interference between MU-MIMO and D2D subsystems: (a) uplink; (b) downlink.

data streams on the D2D link can be expressed by

$$d = \min(M_{d1}, M_{d2} - N, M_{\text{bs}} - N)_+, \tag{3.1}$$

where $(\cdot)_+$ returns a nonnegative number, i.e., $\max(\cdot, 0)$. By the same token, it is easy to verify that d in (3.1) is also the maximum number of achievable data streams on the D2D link in downlink MU-MIMO (see Fig. 3.4(b)).

3.4.3 Interference Management and Its Challenges

Now the question is how to handle the interference at the physical layer so that the D2D link can achieve d data streams while the MU-MIMO subsystem can maintain its N data streams between the BS and the N C-UEs. To answer this question, we consider uplink and downlink MU-MIMO

separately. For the uplink as shown in Fig. 3.4(a), we need to design a signal detection method for both D-UE 2 and the BS so that they can decode their respective signals in the presence of interference from unintended transmitters. For the downlink as shown in Fig. 3.4(b), we need to design a beamforming method for both D-UE 2 and the BS so that they can pre-cancel their generated interference for their unintended receivers.

While there are many results of signal detection and beamforming in the context of MIMO, most of them require global CSI and perfect synchronization. Such requirements entail a large amount of airtime overhead, thereby degrading the spectral efficiency and complicating system operation. In light of this challenge, we propose a new signal detection method and show its resilience to interference. In contrast to existing detection methods (e.g., ZF and MMSE detectors), our signal detection method does not require CSI but is capable of decoding signals in the face of interference. In the downlink, we propose two new beamforming methods for BS and D-UE 2, respectively. Again, the proposed beamforming methods do not require CSI for the design of precoding matrix, differentiating themselves from existing beamforming methods.

3.5 MU-MIMO Communication

In this section, we present new signal detection and beamforming methods for MU-MIMO to handle the interference between the BS and D-UE 1 (see Fig. 3.4). The interference between C-UEs and D-UE 2 will be handled by the D2D communication method presented in the next section.

3.5.1 Basic Idea

In the MU-MIMO subsystem, the BS handles its interference in both uplink and downlink transmissions by leveraging its spatial DoF offered by multiple antennas. Specifically, in the uplink

MU-MIMO as shown in Fig. 3.4(a), the BS performs interference cancellation and signal detection to recover its desired signals from the N C-UEs in the presence of interference from D-UE 1. At the BS, interference cancellation and signal detection will be done using spatial matrices to combine the received signals from its multiple antennas. In the downlink MU-MIMO as shown in Fig. 3.4(b), the BS applies beamforming to pre-cancel its generated interference at D-UE 1. Recall that we assume the BS has more antennas than C-UEs ($M_{\mathrm{bs}} > N$). This assumption ensures that the BS has sufficient spatial DoF to send N data streams towards the N C-UEs and, at the same time, it is able to nullify its generated interference at D-UE 1.

In contrast to the BS, the C-UEs do not participate in the interference management since they have a single antenna. They will rely on D-UE 2 to handle the interference in both uplink and downlink. As such, DM-COM preserves backward compatibility with incumbent C-UEs. In what follows, we focus on the baseband signal processing at the BS. We first present the signal detection method for the uplink and then present the beamforming method for the downlink.

3.5.2 Uplink Signal Detection at the BS

Mathematical Formulation. We consider the uplink MU-MIMO transmissions in the presence of interference from D-UE 1 as shown in Fig. 3.4. Let $\mathbf{s}_{\mathrm{c}} \in \mathbb{C}^{N\times 1}$ denote the vector of signals that are transmitted by the N C-UEs. Let $\mathbf{s}_{\mathrm{d}} \in \mathbb{C}^{d\times 1}$ denote the vector of signals that are transmitted by D-UE 1. Let $\mathbf{P}_{\mathrm{d}} \in \mathbb{C}^{M_{\mathrm{d1}}\times d}$ denote its precoding vector. Also, $\mathbf{H}_{\mathrm{c}} \in \mathbb{C}^{M_{\mathrm{bs}}\times N}$ denotes the compound channel between the BS and the N C-UEs, and $\mathbf{H}_{\mathrm{d}} \in \mathbb{C}^{M_{\mathrm{bs}}\times M_{\mathrm{d1}}}$ stands for the MIMO channel between the BS and D-UE 1. We further let $\mathbf{w} \in \mathbb{C}^{M_{\mathrm{bs}}\times 1}$ denote noise at the receiving BS. Then, the vector of received signals at the BS, which we denote as $\mathbf{y} \in \mathbb{C}^{M_{\mathrm{bs}}\times 1}$, can be written as:

$$\mathbf{y} = \mathbf{H}_{\mathrm{c}}\mathbf{s}_{\mathrm{c}} + \mathbf{H}_{\mathrm{d}}\mathbf{P}_{\mathrm{d}}\mathbf{s}_{\mathrm{d}} + \mathbf{w}. \tag{3.2}$$

At the BS, to recover its desired signal $\mathbf{s}_{\mathrm{c}}$ in the presence of interference $\mathbf{s}_{\mathrm{d}}$ and noise $\mathbf{w}$, one approach is using conventional detectors, such as ZF and MMSE detectors. These approaches, however, require channel knowledge about $\mathbf{H}_{\mathrm{c}}$ and $\mathbf{H}_{\mathrm{d}}$. While $\mathbf{H}_{\mathrm{c}}$ is easy to obtain, $\mathbf{H}_{\mathrm{d}}$ is not. If the BS intends to obtain $\mathbf{H}_{\mathrm{d}}$, it requires to cooperatively work with D-UE 1, and a dedicated protocol is needed for channel sounding as well. This increases the airtime overhead and complicates network operation remarkably. In light of this challenge, we propose an approximate-MMSE MIMO detector for the BS, which does not require channel knowledge about $\mathbf{H}_{\mathrm{c}}$ and $\mathbf{H}_{\mathrm{d}}$ for signal detection.

Detection Matrix Design. We consider linear detection at the BS. By letting $\mathbf{G} \in \mathbb{C}^{N \times M_{\mathrm{bs}}}$ denote the detection matrix, the estimated signal at the BS can be written as $\hat{\mathbf{s}}_{\mathrm{c}} = \mathbf{G}\mathbf{y}$, where $\hat{\mathbf{s}}_{\mathrm{c}}$ is the estimated version of signal $\mathbf{s}_{\mathrm{c}}$. Then, the Mean Square Error (MSE) between the original signal $\mathbf{s}_{\mathrm{c}}$ and estimated signal $\hat{\mathbf{s}}_{\mathrm{c}}$ can be written as: $\mathrm{MSE} = \mathbb{E}\big[|\mathbf{G}\mathbf{y} - \mathbf{s}_{\mathrm{c}}|^2\big]$, where $|\cdot|^2$ is ℓ^2-norm of a complex vector. By letting $\frac{\partial \mathrm{MSE}}{\partial \mathbf{G}} = \mathbf{0}$, we can obtain $\mathbf{G} = \mathbb{E}[\mathbf{s}_{\mathrm{c}}\mathbf{y}^{\mathsf{H}}]\mathbb{E}[\mathbf{y}\mathbf{y}^{\mathsf{H}}]^{+}$, where $[\cdot]^{+}$ is Moore-Penrose inverse. This is actually another form of MMSE MIMO detector.[2]

To calculate $\mathbf{G}$ in real systems, we need to compute $\mathbb{E}[\mathbf{s}_{\mathrm{c}}\mathbf{y}^{\mathsf{H}}]$ and $\mathbb{E}[\mathbf{y}\mathbf{y}^{\mathsf{H}}]$. To do so, we take advantage of the demodulation reference signals for uplink (PUSCH DM-RS) in the frame structure, as shown in Fig. 3.3. In the uplink frame, one OFDM symbol is used for PUSCH DM-RS within a PRB. We can use these reference signals to estimate $\mathbb{E}[\mathbf{y}\mathbf{y}^{\mathsf{H}}]$ and $\mathbb{E}[\mathbf{y}\mathbf{s}_{\mathrm{c}}^{\mathsf{H}}]$. Let us define that a PRB has 12 subcarriers and 14 OFDM symbols. Let $\mathcal{R}$ denote the set of PUSCH DM-RS elements in an uplink PRB as shown in Fig. 3.3. Let k and l denote the index of subcarriers and OFDM symbols, respectively. Then, we have $\mathbb{E}[\mathbf{y}\mathbf{y}^{\mathsf{H}}] \approx \frac{1}{|\mathcal{R}|}\sum_{(l,k)\in\mathcal{R}} \mathbf{y}(l,k)\mathbf{y}(l,k)^{\mathsf{H}}$ and

[2] By letting $\mathbf{H}$ denote the compound channel and assuming that the distribution of transmit signal is i.i.d., $\mathbf{G}$ can be transformed to its classical form: $\mathbf{G} = \mathbb{E}[\mathbf{s}_{\mathrm{c}}\mathbf{y}^{\mathsf{H}}]\mathbb{E}[\mathbf{y}\mathbf{y}^{\mathsf{H}}]^{+} = \mathbf{H}^{\mathsf{H}}(\mathbf{H}\mathbf{H}^{\mathsf{H}} + \sigma^2\mathbf{I})^{-1}$, where σ^2 is the normalized noise power.

$\mathbb{E}[\mathbf{s}_{\mathrm{c}}\mathbf{y}^{\mathsf{H}}] \approx \frac{1}{|\mathcal{R}|}\sum_{(l,k)\in\mathcal{R}} \mathbf{s}_{\mathrm{c}}(l,k)\mathbf{y}(l,k)^{\mathsf{H}}$. Consequently, $\mathbf{G}$ can be approximately expressed as:

$$\mathbf{G} = \Big[\sum_{(l,k)\in\mathcal{R}} \mathbf{s}_{\mathrm{c}}(l,k)\mathbf{y}(l,k)^{\mathsf{H}}\Big]\Big[\sum_{(l,k)\in\mathcal{R}} \mathbf{y}(l,k)\mathbf{y}(l,k)^{\mathsf{H}}\Big]^{+}, \tag{3.3}$$

where $\mathbf{s}_{\mathrm{c}}(l,k)$, $(l,k)\in\mathcal{R}$, is a PUSCH DM-RS element at the N C-UEs; and $\mathbf{y}(l,k)$, $(l,k)\in\mathcal{R}$, is the corresponding received signal at the BS, which includes both PUSCH DM-RS element from the C-UEs and interfering signals from D-UE 1. We note that in (3.3), we replace the approximation sign ($\approx$) with equation sign ($=$) for simplicity. We also note that, since $\mathbf{G}$ in (3.3) is an approximation of MMSE MIMO detector, we therefore term it approximate-MMSE MIMO detector.

Performance Analysis. The approximate-MMSE MIMO detector does not require CSI for the signal detection. Instead, it uses the transmitted and received reference signals to compute the detection matrix. For this reason, the approximate-MMSE MIMO detector can decode desired signals in the presence of unknown interference.

It is interesting to explore the performance of this approximate-MMSE MIMO detector in the cellular network. Let us assume that the signals in a PRB experience the same channel, i.e., channel coherence frequency is greater than 12 subcarriers (180 kHz) and channel coherence time is greater than 14 OFDM symbols (1 ms). Let us further assume that the noise is negligible (i.e., zero-noise). We have the following lemma:

Lemma 3. If th BS is equipped with sufficient number of antennas then the approximate-MMSE MIMO detector at BS can perfectly decode the signals from the N C-UEs, i.e., $\hat{\mathbf{s}}_{\mathrm{c}}(l,k) = \mathbf{s}_{\mathrm{c}}(l,k)$, $\forall l,k$.

Proof. We denote $\mathbf{H}(k)$ as the compound channel between the BS and all the transmitting UEs over subcarrier k, i.e., $\mathbf{H}(k)=\Big[\mathbf{H}_{\mathrm{c}}(k)\ \ \mathbf{H}_{\mathrm{d}_1}(k)\Big]$; we also denote $\mathbf{S}(l,k)$ as the compound transmit

signals at all C-UEs and D-UE 1, i.e., $\mathbf{S}(l,k) = \Big[\mathbf{s}_{\mathrm{c}}(l,k)\ \ \mathbf{s}_{\mathrm{d}_1}(l,k)\Big]^T$. Then, we can re-write (3.2) over subcarrier k and OFDM symbol l as:

$$\mathbf{Y}(l,k) = \mathbf{H}(k)\mathbf{S}(l,k). \tag{3.4}$$

As the auto-correlation matrix of the compound transmit signals, we have

$$\mathbf{R}_{\mathrm{S}} = \mathbb{E}(\mathbf{S}\mathbf{S}^{H}) \overset{(a)}{=} \begin{bmatrix} \mathbf{R}_{\mathrm{c}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathrm{d}} \end{bmatrix} \overset{(b)}{=} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathrm{d}} \end{bmatrix} \tag{3.5}$$

where $\mathbf{R}_{\mathrm{S}}$, $\mathbf{R}_{\mathrm{c}}$, and $\mathbf{R}_{\mathrm{d}}$ are the auto-correlation matrix of the compound transmit signals, auto-correlation matrix of C-UEs' transmit signals, and auto-correlation matrix of D-UE 1 transmit signals, respectively. Equality (a) follows from the fact that the transmit signal from C-UEs are independent of the transmit signals from D-UE 1. Also, (b) follows from the fact that transmit signals from C-UEs are independent too.

Based on (3.3), (3.4), and (3.5), we obtain the approximate-MMSE MIMO detector $\mathbf{G}(k)$ over subcarrier k as follows:

$$\begin{aligned} \mathbf{G}(k) &= \Big[\sum_{(l,k')\in\mathcal{R}} \mathbf{Y}(l,k')\mathbf{Y}(l,k')^{\mathsf{H}}\Big]^{+}\Big[\sum_{(l,k')\in\mathcal{R}} \mathbf{Y}(l,k')\mathbf{s}_{\mathrm{c}}(l,k')^{\mathsf{H}}\Big] \\ &= \mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{Y}(l,k)^{\mathsf{H}}\big]^{+}\mathbb{E}\big[\mathbf{Y}(l,k)\mathbf{s}_{\mathrm{c}}(l,k)^{\mathsf{H}}\big] \\ &= \big[\mathbf{H}(k)\mathbf{R}_{\mathrm{S}}\mathbf{H}(k)^{\mathsf{H}}\big]^{+}\big[\mathbf{H}(k)\mathbf{I}'\big], \end{aligned} \tag{3.6}$$

where $\mathbf{I}'$ is a matrix which its entries on the diameter are one and other entries are zero. Then, we

have

$$\begin{aligned}\hat{\mathbf{s}}_{\mathrm{c}}(l,k) &= \mathbf{G}(k)^{*}\mathbf{Y}(l,k) \\ &= \Big\{ \big[\mathbf{H}(k)\mathbf{R}_{\mathrm{S}}\mathbf{H}(k)^{\mathsf{H}}\big]^{+}\big[\mathbf{H}(k)\mathbf{I}'\,\big] \Big\}^{\mathsf{H}} \mathbf{H}(k)\mathbf{S}(l,k) \\ &= \mathbf{s}_{\mathrm{c}}(l,k), \quad \forall l,k. \end{aligned} \tag{3.7}$$

This means that the approximate-MMSE MIMO detector $\mathbf{G}(k)$ is capable of perfectly recovering the original signal over subcarrier k and OFDM symbol l in a noise-free environment. ■

This lemma shows the superior performance of approximate-MMSE MIMO detector in ideal scenarios (frequency-flat channel, sufficiently large channel coherence time, and zero-noise regime). For its performance in non-ideal scenarios, we resort to experimentation. Our experimental results will show that the approximate-MMSE MIMO detector yields a good performance in real network scenarios.

3.5.3 Downlink Beamforming at BS

Beamforming Matrix Design. We now consider the beamforming for downlink MU-MIMO as shown in Fig. 3.4(b). Based on the network information theory, if a network can send N data streams in the uplink, it can also send N data streams in the downlink. This principle inspires us in the design of beamforming matrix. Our beamforming method is simple – we use the detection matrix derived in the uplink as the beamforming matrix in the downlink. Let $\mathbf{z}(l,k) \in \mathbb{C}^{N\times 1}$ denote the vector of signals in OFDM symbol l on subcarrier k that the BS wants to send towards N C-UEs. Let $\mathbf{x}(l,k) \in \mathbb{C}^{M_{\mathrm{bs}}\times 1}$ denote the vector of precoded signals in OFDM symbol l on subcarrier k that the BS sends to its M_{bs} antenna ports. Then, the beamforming operation can be expressed as: $\mathbf{x}(l,k) = \alpha\mathbf{G}^{\mathsf{T}}\mathbf{z}(l,k)$, $\forall l,k$, where $\mathbf{G}$ is obtained in the uplink and $\alpha \in \mathbb{R}$ is a

scaling factor to meet the requirement of the BS's transmit power.

In Lemma 3, we showed that the $\mathbf{G}$ matrix can perfectly recover the desired signals at the BS in the uplink. If the uplink and downlink channels reciprocity is maintained, it is evident that the C-UEs can also perfectly recover their respective signals in the downlink. Moreover, the BS can perfectly pre-cancel the interference for D-UE 1, which is a receiver in this time period (see Fig. 3.4(b)). For the beamforming method in non-ideal scenarios, we leave its performance evaluation to our experimental results in Section 3.7.

Channel Calibration. The proposed beamforming method relies on the channel reciprocity. For its deployment in real systems, relative channel calibration at the BS can be implemented to maintain the channel reciprocity. In our experiments, the relative calibration method in [150] was implemented at the BS as a part of beamforming implementation.

3.5.4 Discussions on Its Limitations

Two remarks on this MU-MIMO method are in order. First, channel coherence time plays a critical role in the proposed MU-MIMO method. Suppose that both uplink and downlink occupy one sub-frame (1 ms). Then, the required channel coherence time should be longer than 1 ms. This is a mild requirement in real wireless environments. Second, the performance of the proposed MU-MIMO method is dependent on the number of reference signals in an uplink PRB. Per our experiments, when a device has N_{ant} antennas, $\mathcal{R}$ needs to be selected such that $|\mathcal{R}| \geq 2N_{\text{ant}}$. In this case, the average EVM gap between approximate-MMSE and ideal MMSE detectors is less than 3 dB. As such, D2D and MU-MIMO subsystems individually set an appropriate $\mathcal{R}$ and PUSCH DM-RS pattern to meet their own needs. For instance, $\mathcal{R}$ may embrace more than one PRB or PUSCH DM-RS pattern may entail dense distribution of reference signals to meet the requirements of D-UEs and the BS.

3.6 D2D Communication

In this section, we focus on the D2D communication. As the interference related to D-UE 1 has been tamed by the BS, we now focus on the interference related to D-UE 2. Specifically, we design a D2D communication scheme such that D-UE 2 can properly handle its related interference in both uplink and downlink. A proper D2D scheme has to address the following two questions: For the uplink shown in Figure 3.4(a), how can D-UE 2 decode its intended signals in the presence of interference from C-UEs? For the downlink in Figure 3.4(b), how can D-UE 2 send its signal to D-UE 1 while pre-canceling its generated interference for C-UEs? In what follows, we present our solutions to these questions.

3.6.1 Signal Detection at D-UE 2

Referring to D2D forward transmissions in Figure 3.4(a), we follow the same approach presented in Section 3.5.2 for D-UE 2 to decode its signals in the presence of interference from C-UEs. Specifically, D-UE 2 first calculates a detection matrix using (3.3) and then uses the calculated detection matrix to filter out the interference from C-UEs and equalize the channel distortion for signal recovery. The remaining question is what frame structure should be used for the D2D transmission. Actually, the frame structure for D2D transmission is flexible. As we will show in our experiments, the frame structure of D2D communication can be the same as the MU-MIMO frame structure as shown in Figure 3.3; it also can be IEEE 802.11 frame structure (consisting of preamble and data parts [67]).

3.6.2 Beamforming at D-UE 2

We now consider the D2D backward transmissions in Figure 3.4(b). In this time period, D-UE 2 needs to perform beamforming to pre-cancel its interference for C-UEs. Our beamforming method takes advantage of the overheard interfering signals in the previous time period, as illustrated in Figure 3.2. By leveraging the overheard signals, D-UE 2 constructs a beamforming matrix for signal transmission. In what follows, we detail the construction of beamforming matrix at D-UE 2.

Beamforming Matrix Design. Referring to Figure 3.2, in a short time period at the end of uplink MU-MIMO, D-UE 1 does not transmit signal and thus D-UE 2 receives only interfering signals from C-UEs. Let $\mathbf{Y}_{\mathrm{d}} \in \mathbb{C}^{M_{\mathrm{d}2} \times 1}$ denote the received signals at D-UE 2 in this time period. Then, we have

$$\mathbf{y}_{\mathrm{d}} = \mathbf{H}_{\mathrm{dc}}\mathbf{s}_{\mathrm{c}} + \mathbf{W}_{\mathrm{d}}, \tag{3.8}$$

where $\mathbf{H}_{\mathrm{dc}} \in \mathbb{C}^{M_{\mathrm{d}2} \times N}$ is the channel between C-UEs and D-UE 2; $\mathbf{s}_{\mathrm{c}} \in \mathbb{C}^{N \times 1}$ is the vector of transmit signals at the N C-UEs; and $\mathbf{w}_{\mathrm{d}} \in \mathbb{C}^{M_{\mathrm{d}2} \times 1}$ is the noise vector at D-UE 2.

Let $\mathbf{P}_{\mathrm{d}} \in \mathbb{C}^{M_{\mathrm{d}2} \times d}$ denote the precoding matrix at D-UE 2. Then, based on the received signal $\mathbf{Y}_{\mathrm{d}}$, we construct $\mathbf{P}_{\mathrm{d}}$ as:

$$\mathbf{P}_{\mathrm{d}} = \mathbf{U}(:, M_{\mathrm{d}2} - d + 1 : M_{\mathrm{d}2}), \tag{3.9}$$

where $\mathbf{U}(:, n \ : \ m)$ is a submatrix of $\mathbf{U}$, which is from $\mathbf{U}$'s nth column to mth column. $\mathbf{U}$ is computed by

$$[\mathbf{U}\ \mathbf{D}\ \mathbf{V}] = \mathrm{svd}(\mathbf{y}_{\mathrm{d}}\mathbf{y}_{\mathrm{d}}^{\mathsf{H}}), \tag{3.10}$$

where $\mathbf{D}$ and $\mathbf{V}$ are redundant outputs, and $\mathrm{svd}\,(\cdot)$ denotes singular value decomposition. Using (3.9) and (3.10), we compute a beamforming matrix $\mathbf{P}_{\mathrm{d}}$ for each subcarrier in the OFDM symbols.

Then, the $\mathbf{P}_{\mathrm{d}}$ is applied to the corresponding subcarrier for beamforming during D-UE 2's signal transmission. Since the matrix $\mathbf{P}_{\mathrm{d}}$ is computed using the uplink interfering signal, it necessitates channel reciprocity when using $\mathbf{P}_{\mathrm{d}}$ as the beamforming matrix in the downlink. Therefore, channel calibration has to been done at D-UE 2 in order to pre-cancel its interference for C-UEs. Again, RF calibration can be used at D-UE 2 in the baseband signal processing domain to preserve channel reciprocity.

Performance Analysis. We first study the performance of proposed beamforming scheme in an ideal network scenario. Let us assume that all the MIMO channels have full rank. Let us assume that the channel coherence time is sufficiently large (larger than the duration of downlink). Let us assume that the channel is perfectly calibrated at D-UE 2, i.e., the downlink and uplink channels are reciprocal. Let us further assume that the noise is negligible, and D-UE 2 has sufficient number of antennas, i.e., $d + N \leq M_{\mathrm{d2}}$. Then, we have the following lemma:

Lemma 4. The constructed beamforming matrix $\mathbf{P}_{\mathrm{d}}$ can completely pre-cancel the interference for the N C-UEs on every OFDM subcarrier.

Proof. Referring to Fig. 3.2, D-UE 1 first remains silent for a while, and D-UE 2 merely receives interfering signals from N C-UEs. Then, D-UE 2 uses the overheard interference to design precoding filter for pre-cancelling its generated interference at C-UEs in backward transmission. The received interference can be written as:

$$\mathbf{y}_{\mathrm{d}}(k) = \mathbf{H}_{\mathrm{dc}}(k)\mathbf{s}_{\mathrm{c}}(k), \tag{3.11}$$

where $\mathbf{H}_{\mathrm{dc}}(k)$ denotes the compound channel between D-UE 2 and all the C-UEs over subcarrier

k. Then, we have

$$\mathbb{E}[\mathbf{y}_{\mathrm{d}}(k)\mathbf{y}_{\mathrm{d}}(k)^{\mathsf{H}}] \overset{(a)}{=} \mathbf{H}_{\mathrm{dc}}^{[1]}(k)\mathbf{H}_{\mathrm{dc}}^{[1]}(k)^{\mathsf{H}}, \tag{3.12}$$

where (a) follows from the fact that $\mathbb{E}[\mathbf{s}_{\mathrm{c}}(k)\mathbf{s}_{\mathrm{c}}(k)^{\mathsf{H}}] = \mathbf{I}$ as the N C-UEs send N independent data streams. Recall that $N + d \leq M_{\mathrm{d2}}$ and consequently $d \leq M_{\mathrm{d2}} - N$. Based on the right hand side of (3.12), rank of $\mathbb{E}[\mathbf{y}_{\mathrm{d}}(l,k)\mathbf{y}_{\mathrm{d}}(l,k)^{\mathsf{H}}]$ is at most N. The rank reduces when channel is correlated and rank deficient. Therefore, $\mathrm{svd}(\mathbf{y}_{\mathrm{d}}(k)\mathbf{y}_{\mathrm{d}}(k)^{\mathsf{H}})$ has at least d zero singular vectors. If $\mathbf{u}_i$ denotes the ith left singular vector, based on (3.12), we have

$$\Big(\mathbf{H}_{\mathrm{dc}}(k)\mathbf{H}_{\mathrm{dc}}(k)^{\mathsf{H}}\Big)\mathbf{u}_i = \mathbf{0}, \;\; M_{\mathrm{d2}} - d + 1 \leq i \leq M_{\mathrm{d2}}. \tag{3.13}$$

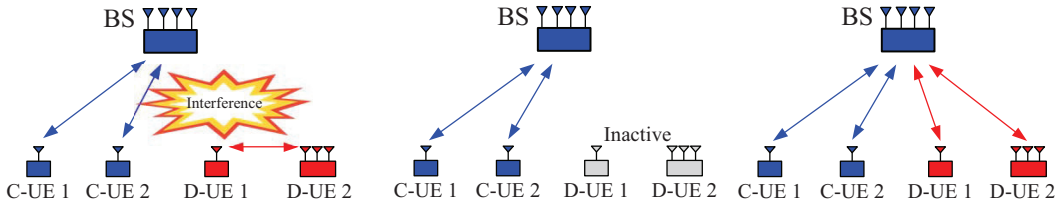
If channel reciprocity is maintained with the aid of a channel calibration method, $\mathbf{H}_{\mathrm{cd}}(k) = \big(\mathbf{H}_{\mathrm{dc}}(k)\big)^T$. Then, it is easy to show that $\mathbf{H}_{\mathrm{cd}}(k)\mathbf{P} = \mathbf{0}$.

■

Lemma 4 shows the superior performance of the proposed beamforming method. It is worth noting that, although the beamforming technique presented in Section 3.5 works for D-UE 2, we observed in experiments that the proposed Singular Value Decomposition (SVD)-based technique has superior performance in terms of interference leakage. In light ot this, the proposed technique is applied on D-UE 2 to preserve the performance of MU-MIMO subsystem.

3.7 Experimental Evaluation

In this section, we build a prototype of DM-COM and evaluate its performance in a small network.



(a) DM-COM (b) Cellular-MU-MIMO (C) Full-MU-MIMO

Figure 3.5: Experimental setup and comparison baselines.

3.7.1 Implementation and Experimental Setup

Implementation. We have built a wireless network testbed that consists of a BS, two C-UEs, and two D-UEs as shown in Fig. 3.5(a). The BS has four antennas. The C-UEs has one antenna. D-UE 1 has one antenna. D-UE 2 has three antennas. The BS, C-UEs, and D-UEs are built using USRP N210 devices as the radio transceivers and general-purpose computers as baseband signal processors.

We implement DM-COM on this testbed. The MU-MIMO subsystem is implemented using a custom-built 5G NR PHY, while the D2D subsystem is implemented using both NR-like and WiFi-like PHYs. The PHY parameters of DM-COM implementation are listed in Table 3.1. Based on these PHYs, we implement the MAC protocols for both MU-MIMO and D2D subsystems as shown in Fig. 3.2. For the MU-MIMO protocol, both uplink and downlink transmissions have the same duration. For the D2D protocol, the time duration of "listening at D-UE 2" is about 71.35 µs.

Experimental Setup. Fig. 3.6 depicts the floor plan of our experimentations. The BS and C-UEs are always placed on the spots marked by blue and red colors, respectively. The distance between BS and cellular users is about 7 m. D-UE 1 and D-UE 2 are deployed over 50 random locations in Fig. 3.6. In each location, the distance between D-UEs is about 3 m. We use the indoor environments for ease of experimentation. Moreover, many small cells will be deployed in the buildings as mobile hotspot in the near future.

Table 3.1: The parameters of experimental network.

	MU-MIMO subsystem	D2D subsystem 1	D2D subsystem 2
Standard	NR-like	NR-like	WiFi-like
Waveform	OFDM	OFDM	OFDM
FFT point	512	512	64
Valid subcarrier	300	300	52
Sample rate	5 Msps	5 Msps	5 Msps
Symbol duration	71.35 μs	71.35 μs	16 μs
Signal bandwidth	2.9 MHz	2.9 MHz	4 MHz
Carrier frequency	2.48 GHz	2.48 GHz	2.48 GHz
Transmit power	$\sim$ 18 dBm	$\sim$ 18 dBm	$\sim$ 18 dBm

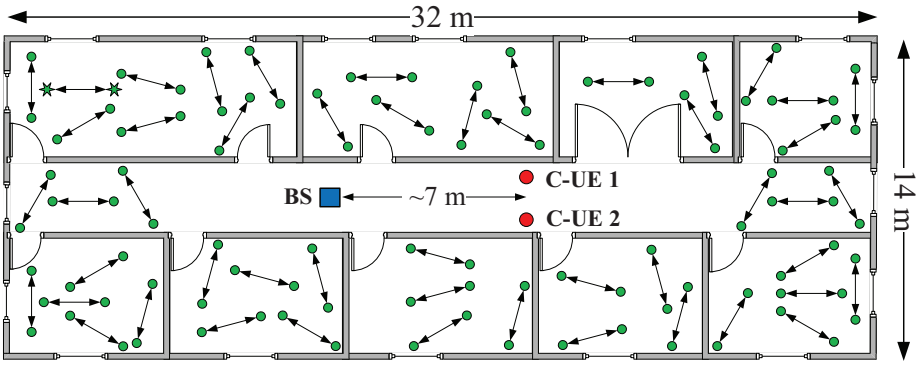
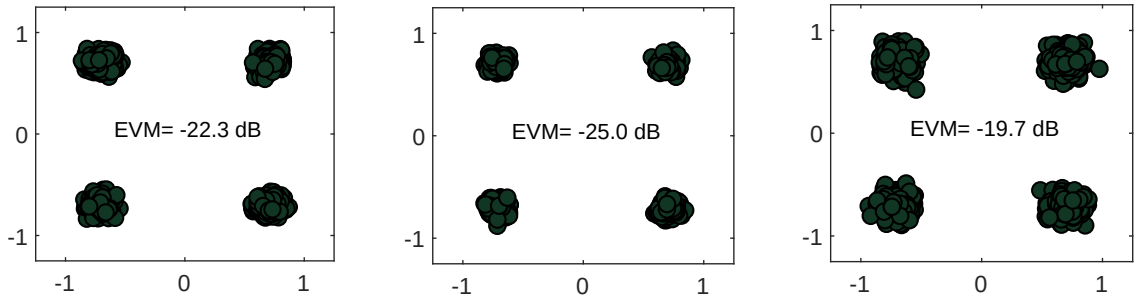


Figure 3.6: The floor plan of our experimentation.

3.7.2 Performance Metrics and Comparison Baselines

Performance Metrics. We use two metrics to evaluate DM-COM. The first one is EVM, which is defined as follows: $\text{EVM} = 10\log_{10}(\frac{\mathbb{E}[|\hat{S}(l,k)-S(l,k)|^2]}{\mathbb{E}[|S(l,k)|^2]})$, where $\hat{S}(l,k)$ and $S(l,k)$ are the estimated and original signals, respectively. EVM is widely used in both IEEE 802.11 standards [67] and 3GPP standards [1] to measure quality of decoded signals, define modulation Modulation and Coding Scheme (MCS), and estimate the achievable data rate as we see shortly.

The second metric is the achievable data rate. Based on the measured EVM, we extrapolate the achievable data rate using the MCS defined in the 3GPP standard and IEEE 802.11ac standard as follows: $r = \frac{1}{2} \cdot \frac{N_{\text{sc}}}{N_{\text{fft}}+N_{\text{cp}}} \cdot b \cdot \gamma\,(\text{EVM})$, where coefficient $1/2$ stems from halftime uplink

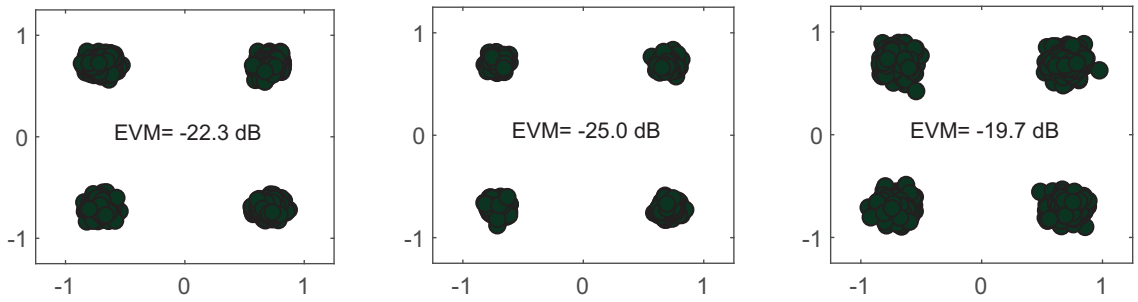


(a) Decoded signal from C-UE 1 at BS.

(b) Decoded signal from C-UE 2 at BS.

(c) Decoded signal fromD-UE 1 at D-UE 2.

Figure 3.7: Decoded (demodulated) signals in the MU-MIMO and D2D subsystems in the uplink/forward transmission.



(a) Decoded signal from BS at C-UE 1.

(b) Decoded signal from BS at C-UE 2.

(c) Decoded signal fromD-UE 2 at D-UE 1.

Figure 3.8: Decoded (demodulated) signals in the MU-MIMO and D2D subsystems in the downlink/backward transmission.

and halftime downlink transmissions in MU-MIMO. N_{sc}, N_{fft}, and N_{cp} denote number of used subcarriers, Fast Fourier Transform (FFT) points, and the length of cyclic prefix, respectively. b is the sampling rate in Msps. $\gamma(\text{EVM})$ is the spectral efficiency of transmission based on MCS selection defined in standards. Table 2.3 and Table 2.2 present $\gamma\left(\text{EVM}\right)$ for NR-like and WiFi-like PHYs, respectively.

Comparison Baselines. As shown in Fig. 3.5, we compare DM-COM with two existing schemes: Cellular-MU-MIMO and Full-MU-MIMO. In the Cellular-MU-MIMO, the BS serves the two C-UEs only, and the two D-UEs are deactivated. In the Full-MU-MIMO, the BS serves the two C-UEs while the two D-UEs communicate with each other with the aid of BS. Technically,

the BS simultaneously serves the four UEs in both uplink and downlink.

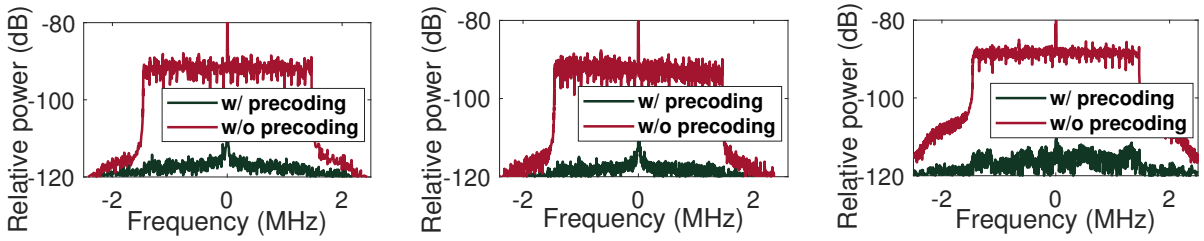
3.7.3 A Case Study of DM-COM

We first use a case study to scrutinize DM-COM and its interference cancellation capability. In this case study, we place the two D-UEs at two spots marked by stars in the upper-left room in Fig. 3.6, and the D2D subsystem uses NR-like PHY for communications. Recall that DM-COM comprises two phases: uplink and downlink, as shown in Fig. 3.4. In what follows, we first examine the decoded signals in the two phases and then study the interference cancellation capability.

Constellation, EVM, and Data Rate. Referring to Fig. 3.4(a), in the uplink, the BS demodulates the signals from the two C-UEs; at the same time, D-UE 2 demodulates the signal from D-UE 1. Fig. 3.7 exhibits the constellation of the demodulated signals at the BS and D-UE 2, as well as their EVMs. Based on the measured EVM, the uplink data rates of C-UE 1 and C-UE 2 are extrapolated to 6.2 Mbps and 7.6 Mbps, respectively. Meanwhile, the data rate of D-UE 2 is extrapolated to 4.5 Mbps.

Referring to Fig. 3.4(b), in the downlink, the BS sends the data to the two C-UEs; at the same time, D-UE 2 sends data to D-UE 1. Fig. 3.8 presents the constellation of the demodulated signals at the two C-UEs and D-UE 1, as well as their EVMs. Based on the measured EVM, the downlink data rates of C-UE 1 and C-UE 2 are extrapolated to 5.3 Mbps and 6.2 Mbps, respectively. Meanwhile, the data rate of D-UE 1 is extrapolated to 4.6 Mbps.

Beamforming Capability. Referring to Fig. 3.4(b), in the downlink, we examine the effectiveness of beamforming at the two transmitters (BS and D-UE 2) . To do so, we measure the interference at the receiving nodes (C-UE 1, C-UE 2, and D-UE 1) in two cases: with and without beamforming. For the case without beamforming, we use precoder $[1/2,\ 1/2,\ 1/2,\ 1/2]$ at the BS and $[1/\sqrt{3},\ 1/\sqrt{3},\ 1/\sqrt{3}]$ at D-UE 2. Fig. 3.9 presents the measured the interference at the receiv-



(a) Received interference at C-UE 1.

(b) Received interference at C-UE 2.

(c) Received interference at D-UE 1.

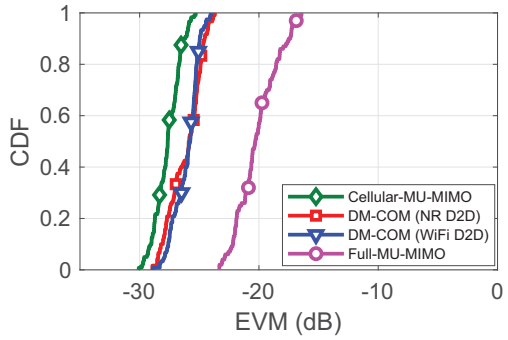
Figure 3.9: Received interference at the receiving nodes in the downlink with and without beamforming at the transmitters.

ing nodes. It is evident that the proposed beamforming methods can very effectively pre-cancel the interference. Specifically, both beamforming methods achieve at least 28.5 dB interference cancellation capability. Thanks to the effective beamforming methods, both MU-MIMO and D2D subsystems can achieve superior performance in the downlink, as shown in Fig. 3.8.

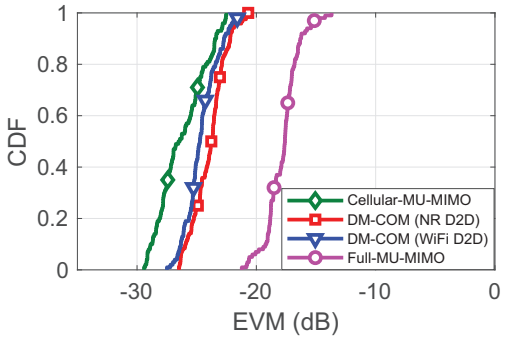
3.7.4 DM-COM vs. Cellular-MU-MIMO and Full-MU-MIMO

By the same token in the case study, we now study the performance of DM-COM by placing the two D-UEs at 50 different locations as shown in Fig. 3.6. In this study, we use Cellular-MU-MIMO and Full-MU-MIMO as the comparison baselines (see Fig. 3.5).

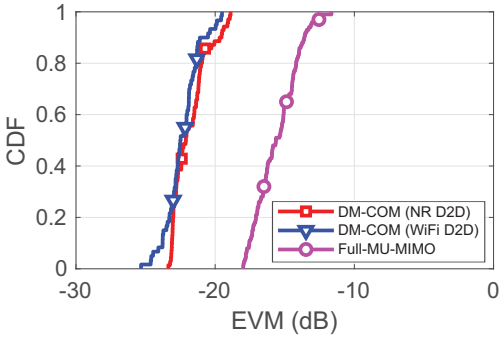
EVM Distribution. Fig. 3.10 presents the distribution of measured EVM when the three schemes are used. Specifically, Fig. 3.10(a) presents the measured EVM of demodulated signals at the BS in the uplink MU-MIMO when DM-COM, Cellular-MU-MIMO, and Full-MU-MIMO are respectively used. Particularly, we considered two cases for DM-COM: (i) D2D subsystem uses NR-like PHY and (ii) D2D subsystem uses WiFi-like PHY. From the figure, we can see that DM-COM achieves -26.1 dB EVM on average, no matter which PHY (5G NR or WiFi) is used for D2D communications. In contrast, Cellular-MU-MIMO achieves about -27.6 dB EVM on average, and Full-MU-MIMO achieves -20.1 dB EVM on average. The EVM gap between DM-COM



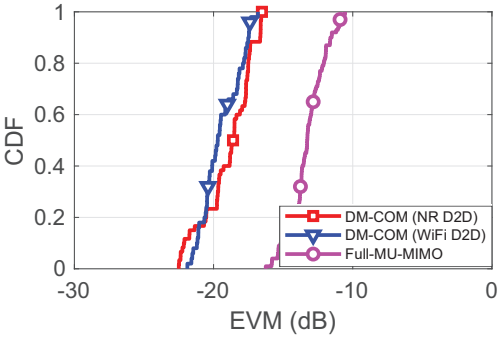
(a) EVM in uplink MU-MIMO.



(b) EVM in downlink MU-MIMO.



(c) EVM in forward D2D.



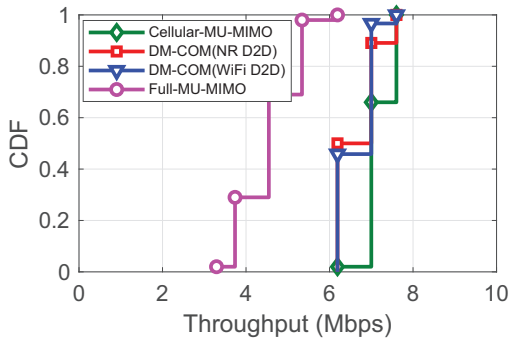
(d) EVM in backward D2D.

Figure 3.10: EVM distribution of demodulated signals when DM-COM, Cellular-MU-MIMO, and Full-MU-MIMO are used.

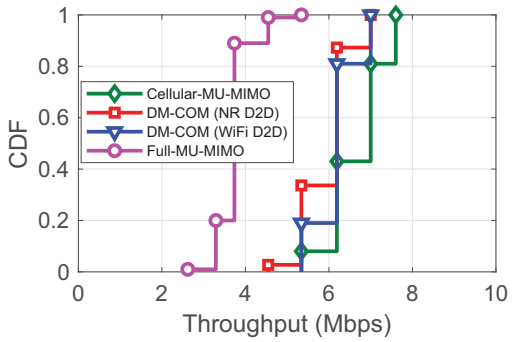
and Cellular-MU-MIMO is only 1.5 dB. This means that, in DM-COM, the EVM degradation at the BS caused by the interference from D2D subsystem is only 1.5 dB.

Fig. 3.10(b) presents the measured EVM of the demodulated signals at the two C-UEs in the downlink MU-MIMO. It shows that DM-COM achieves an average of -24.3 dB EVM in the downlink MU-MIMO. The EVM gap between DM-COM and Cellular-MU-MIMO is about 1.9 dB. This means that, in DM-COM, the EVM degradation at C-UEs caused by the interference from D2D subsystem is only 1.9 dB.

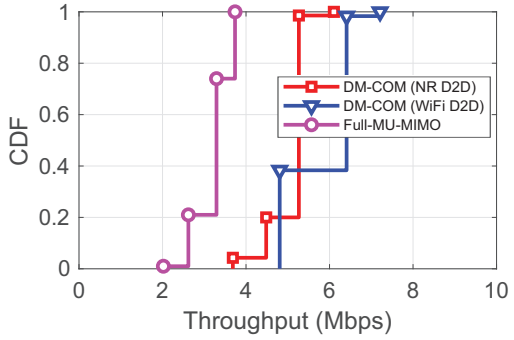
Fig. 3.10(c) and (d) present the measured EVM in forward and backward D2D transmissions when DM-COM and Full-MU-MIMO are used. Note that Cellular-MU-MIMO does not support D2D communication, and thus these two figures do not include the results from Cellular-MU-MIMO. On average, DM-COM achieves -22.1 dB EVM for forward D2D transmission and



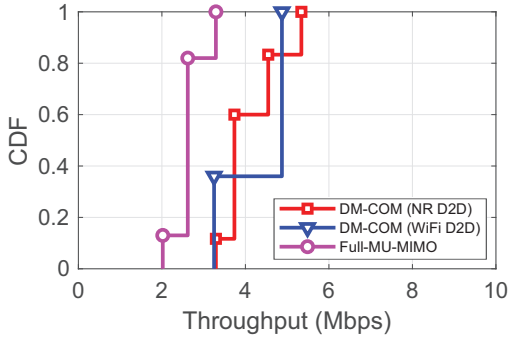
(a) Per-UE throughput in uplink MU-MIMO.



(b) Per-UE throughput in downlink MU-MIMO.



(c) D2D forward throughput.

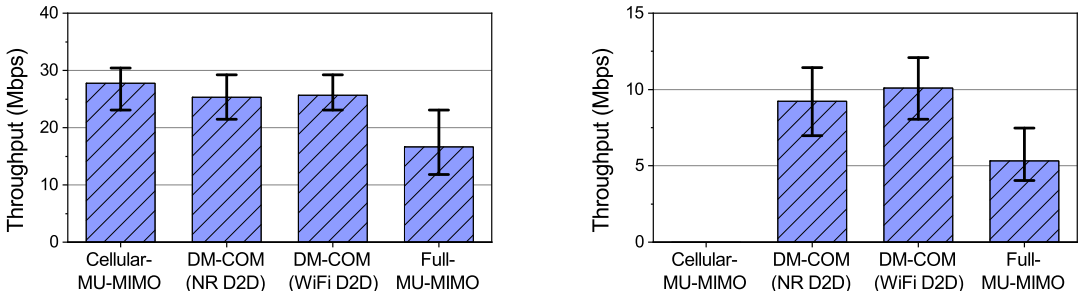


(d) D2D backward throughput.

Figure 3.11: Distribution of extrapolated throughput when DM-COM, Cellular-MU-MIMO, and Full-MU-MIMO are used.

-19.2 dB EVM for backward D2D transmission, no matter which PHY (NR or WiFi) is used for D2D subsystem. In contrast, Full-MU-MIMO achieves -15.6 dB EVM for forward D2D transmission and -13.2 dB EVM for backward D2D transmission. This means that DM-COM outperforms Full-MU-MIMO by 6.5 dB in forward D2D communication and 6.0 dB in backward D2D communication.

Per-UE Throughput Distribution. We extrapolate per-UE throughput (dat rate) based on the measured EVM. Fig. 3.11 presents the results. The staircase shape of the curves stems from the MCS selection, which yields discrete data rate region in nature. On average, DM-COM achieves 6.7 Mbps per-UE throughput in uplink MU-MIMO and 6.1 Mbps per-UE throughput in downlink MU-MIMO. At the same time, it achieves 5.4 Mbps for forward D2D transmission and 4.2 Mbps for backward D2D transmission.



(a) Total throughput of MU-MIMO.

(b) Total throughput of D2D.

Figure 3.12: Total throughput of the MU-MIMO and D2D subsystems when DM-COM, Cellular-MU-MIMO and Full-MU-MIMO are respectively used.

3.7.5 Summary of Observations

Fig. 3.12 presents the total throughput of MU-MIMO and D2D subsystems when DM-COM, Cellular-MU-MIMO, and Full-MU-MIMO are used. The total throughput of MU-MIMO is the summation of its uplink and downlink data rates. The total throughput of D2D is the summation of its backward and forward data rates. The total throughput are averaged over the 50 different locations in Fig. 3.6.

MU-MIMO subsystem. Fig. 3.12(a) shows that DM-COM achieves 25.3 Mbps throughput for MU-MIMO subsystem when using 5G NR PHY for D2D and 25.7 Mbps throughput when using WiFi PHY for D2D. In contrast, Cellular-MU-MIMO achieves 27.8 Mbps throughput for MU-MIMO. This means that, in DM-COM, the throughput degradation of C-UEs caused by the interference from D-UEs is only 8%. Full-MU-MIMO achieves 16.6 Mbps, which is much less than DM-COM.

D2D subsystem. Fig. 3.12(b) shows that DM-COM achieves 9.2 Mbps throughput for D2D subsystem when using 5G NR PHY and 10.1 Mbps throughput when using WiFi PHY. Recall that the system bandwidth is 5 MHz. This means that DM-COM achieves more than 1.9 bit/s/Hz spectral efficiency for D2D communication. In contrast, Full-MU-MIMO achieves 5.3 Mbps through-

put for D2D. This means that DM-COM outperforms Full-MU-MIMO by 82%. This can be partially attributed to the two-hop D2D communication in Full-MU-MIMO.

3.8 Chapter Summary

In this chapter, we presented DM-COM, a practical scheme that combines D2D and MU-MIMO technologies to advance cellular networks. The main challenge in DM-COM is managing the interference between D2D and MU-MIMO subsystems. DM-COM takes the advantage of multiple antennas on the network devices to cancel the interference and recover the desired signals, without requiring channel state information and fine-grained inter-system synchronization. This was achieved through the design of practical yet effective multi-user detection and beamforming methods. We have built a prototype of DM-COM on a custom-built wireless testbed and compared its performance with two existing schemes. Our experimental results show that DM-COM achieves 1.9 bit/s/Hz spectral efficiency for D2D users. Moreover, the throughput degradation of MU-MIMO users due to the spectrum utilization of D2D users is less than 8%.

Chapter 4

Non-Orthogonal Multiple Access for WLANs

4.1 Introduction

Multiple access is a crucial mechanism for wireless network infrastructure to serve multiple users. OMA techniques (e.g., TDMA and Frequency Division Multiple Access (FDMA)), albeit easy to implement, are incapable of approaching network capacity limit due to their exclusivity in resource allocation. This issue becomes particularly acute for networks with strict user fairness requirements. NOMA has recently emerged as a new multiple access paradigm for infrastructure-based wireless networks. Since its inception, NOMA has attracted a large amount of research attention and has been widely regarded as a promising candidate for Radio Access Technologies (RAT) for 5G networks and beyond. In contrast to OMA, NOMA allows multiple users to utilize the same spectrum band for signal transmissions at the same time and, therefore, offers many advantages such as improving spectral efficiency, enhancing resource allocation flexibility, reducing scheduling latency, increasing cell-edge throughput, and enabling massive connectivity.

Recognizing its great potentials, power-domain NOMA has been studied in a variety of network settings in an increasingly sophisticated form, such as power allocation in Single-Input Single-Output (SISO) networks [44, 48, 207], precoder design in Multi-Input Single-Output (MISO) networks [7, 32, 55, 209], and privacy protection [25, 108, 208]. Although a considerable amount of research efforts have been made on the study of NOMA, most of them are limited to theoretical

exploration and performance analysis in cellular networks. Very limited progress has been made so far in the development of practical NOMA schemes and experimental validation of NOMA in real wireless network settings. This stagnation reflects the challenges in the design of practical NOMA schemes and the engineering issues related to their implementations, such as channel acquisition and precoding on the transmitter side and SIC realization on the receiver side.

In this chapter, we aim to make a concrete step forward to bridge this gap by proposing a practical downlink NOMA scheme for WLANs and evaluating its performance on a wireless testbed. We consider an AP that has one or multiple antennas and a set of widely distributed users that have one antenna each. In such a network setting, we first examine the precoder design problem at the AP for downlink NOMA transmissions. We formulate the precoder design problem as an optimization problem, which inevitably includes non-convex constraints due to the intrinsic complication of the problem. To solve this problem, we employ a Minorization-Majorization (MM) approach for convexification of constraints. Based on the convexification results, we further develop an iterative algorithm to solve the precoder design problem.

Based on the solution to the precoder design problem, we develop a downlink NOMA scheme to enable concurrent data transmissions from an AP to multiple users. Our NOMA scheme features a lightweight user grouping strategy and a new SIC method. Specifically, on the transmitter (AP) side, we develop a heuristic algorithm to group the users for downlink NOMA transmission; on the receiver (user) side, we propose a robust SIC algorithm for interference subtraction and signal detection. In contrast to existing SIC methods [167], which first estimate the channels and then use the estimated channels to decode the signal/interference sequentially, our proposed SIC method does not require channel knowledge for interference subtraction and signal detection. Instead, it directly uses the reference signals (the precoded preamble in a frame) to compute the detection filters, which are used for interference subtraction and signal detection. As channel estimation is

vulnerable to interference, the removal of channel estimation in the SIC procedure improves the performance and reliability of signal detection in our NOMA scheme.

We have built a prototype of the proposed NOMA scheme on a GNURadio-USRP2 wireless testbed using IEEE 802.11 legacy parameters and conducted extensive experiments in indoor office wireless environments to evaluate its performance in comparison with a conventional TMDA-based OMA scheme. We consider the following three network settings: (i) the AP has one antenna and it serves two users; (ii) the AP has two antennas and it serves two users; and (iii) the AP has two antennas and it serves three users. Our experimental results show that, compared to OMA, the proposed NOMA scheme can significantly improve the data rate of the weak user and considerably improve the weighted sum rate of all users. Specifically, for the cases that we have examined, the average improvement of data rate of the weak users is about 93.1%, and the average improvement of weighted sum rate of all users is about 36.1%. Moreover, our experimental results show that, on average over all the cases that we have considered, our proposed SIC method outperforms the conventional SIC method (least-squares channel estimation and zero-forcing signal detection) by 13.4% for the AP's weighted sum rate and 39.6% for the data rate of users performing SIC.

4.2 Related work

Since its inception, NOMA has been studied in an increasingly sophisticated form for cellular networks. Given that this work studies power-domain NOMA for the downlink of wireless networks, we focus our literature review on this specific area.

Power Allocation for NOMA. Power allocation for NOMA has been well studied in cellular networks where each node has a single antenna. These research efforts mainly focus on the power allocation strategies for NOMA under different performance considerations, such as user fairness

[162, 188], outage probability [39, 196], and achievable throughput [129, 199]. This research line was then expanded to joint optimization of power allocation and subcarrier assignment for NOMA in OFDMA networks. These research efforts have produced many results, such as maximizing sum rate subject to the power constraints [44, 48, 207], minimizing the power consumption subject to SIC and rate requirements [99], and developing tractable algorithms [106].

Precoder Design for NOMA. When the base station (BS) has multiple antennas, the power allocation problem in downlink NOMA is escalated to precoder design problem as the power allocation and beam steering operations are tightly coupled. The precoder design at the BS needs to jointly optimize NOMA's power allocation and Multi-Input Multi-Output (MIMO)'s beam steering. In the literature, precoder design has been studied toward different objectives, such as maximizing network throughput [55, 209], maximizing transmitter's energy efficiency [7, 32], and preserving users' signal privacy [25, 108, 208]. In what follows, we discuss the papers that are mostly relevant to our work.

In [55], the precoder design problem has been studied for NOMA to maximize sum rate, subject to the SINR constraints in the SIC process at all the users. A non-convex optimization problem was formulated and an iterative algorithm was developed to pursue a feasible solution. In [209], the precoder design problem has been studied to maximize the sum rate of a sophisticated hybrid network where an unmanned aerial vehicle and a BS serve a set of ground users. Precoder design aimed to nullify the cross-network interference or maintain the interference below a certain threshold.

The precoder design problem has also been studied for security enhancement and privacy preservation for power-domain NOMA. In [210], NOMA was studied under eavesdropping attacks, and artificial jamming approach was studied to combat the attacks. A non-convex optimization problem was formulated to maximize the artificial jamming power and, similar to [55], an

iterative algorithm was developed to solve the problem. In [25], precoders were designed to ensure the privacy of a particular user. Specifically, precoders were designed to ensure that the private user's signal is of the weakest strength at all the users except itself. By doing so, none of the users are capable of decoding the private user's message.

As we shall see, our mathematical formulation of the precoder design problem is different from those existing ones. It features practical considerations in the design of precoders for downlink NOMA.

User Grouping for NOMA. User grouping is another key component of NOMA. In [37], the impact of user grouping on the performance of NOMA was studied. It shows that the throughput gain of NOMA (over OMA) becomes more significant when the channel strengths of the users in a group increases. However, reaching the optimal user grouping solution demands an exhaustive search. In [104] and [205], it was shown that the computational complexity of exhaustive-search-based user grouping algorithm can be relaxed by pruning the search space. Greedy grouping algorithms (e.g., [36]) and matching-based grouping algorithms (e.g., [100]) were proposed to reach a near-optimal solution. To further reduce the computational complexity, [38] proposed a random grouping algorithm, which needs a very low computation. However, this random algorithm cannot fully exploit the throughput gain of NOMA. The user grouping algorithm in our work is a lightweight heuristic algorithm, and it is amenable to practical implementation.

Experimental Validation of NOMA. While there is a large body of theoretical work on NOMA, experimental validation of NOMA in real wireless environments remains limited. Some pioneering work can be found in [18–21]. Our work differs from these research efforts in the following two aspects. First, these research efforts study NOMA in cellular networks, while our work focuses on NOMA for WLANs. Cellular networks and WLANs have significant differences in many aspects, including frame format, transmission pattern, transmit power, and receiver sensi-

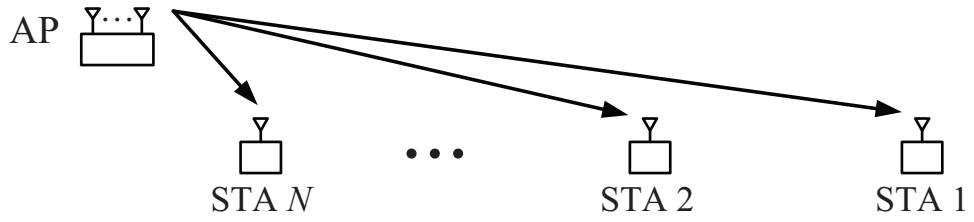


Figure 4.1: Downlink data transmission in a WLAN.

tivity. The results of NOMA in cellular networks cannot be directly applied to WLANs. Second, existing experimental efforts primarily investigated the gain of NOMA over OMA with respect to different system parameters and did not take into account precoder design for the performance optimization of NOMA. Our work considers both precoder optimization and NOMA implementation in WLANs.

4.3 Problem Description

We consider a WLAN as shown in Fig. 4.1, which comprises an AP and a set of user devices (a.k.a. stations, STAs, or users for simplicity). The AP has one or more antennas, and each station has a single antenna. Denote M as the number of antennas on the AP. Denote $\mathcal{N}$ as the set of stations, with N being its cardinality ($N = |\mathcal{N}|$). In this network, we assume that the signal from the AP to the stations experiences significantly different path losses. That is, the signal received by STA i is much stronger than the signal received by STA $i-1$, for $2 \leq i \leq N$. This assumption can be fulfilled through a user selection/scheduling algorithm at the upper layer.

A Premier of NOMA. In power domain, NOMA takes advantage of the power difference between the interference and desired signals to mitigate interference and decode the desired signal at a receiver. SIC is typically used at the receivers for interference mitigation and signal decoding [167]. To illustrate the original concept behind power-domain NOMA, let us consider the

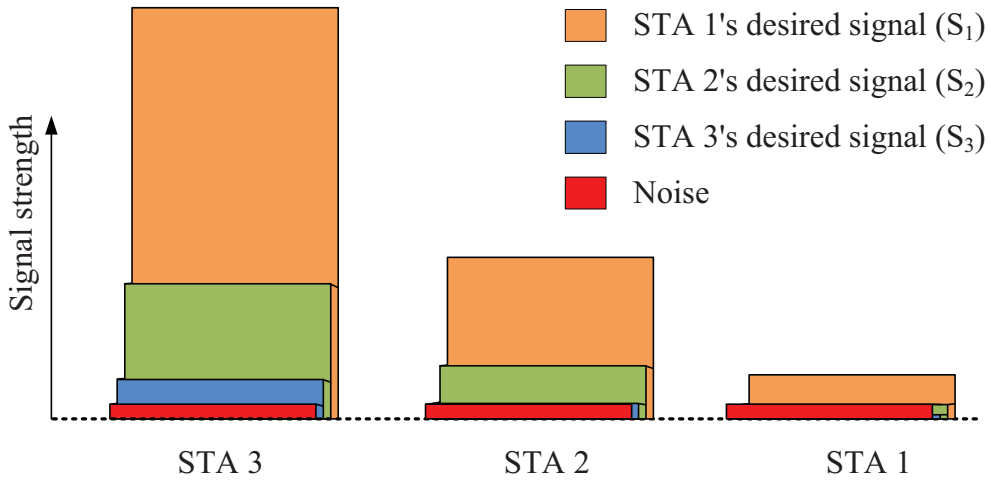


Figure 4.2: Illustration of NOMA in downlink data transmission in a WLAN for $N = 3$.

network in Fig. 4.2 as an example. In this network, a single-antenna AP serves three stations standing far from each other. The AP sends superimposition of three signals to all stations: signal s_1 for STA 1, signal s_2 for STA 2, and signal s_3 for STA 3, with a proper power allocation for these signals. At the stations, the received signals have significantly different strengths as illustrated in Fig. 4.2. The difference in signal strengths makes it possible for the stations to perform SIC. At STA 1, since the undesired signals s_2 and s_3 are relatively weak due to the large path loss, the desired signal s_1 can be easily decoded by treating interference (s_2 and s_3) as noise. At STA 2, the strongest undesired signal s_1 can be first decoded and subtracted from what is received. For the resulting signals, the desired signal s_2 can be easily decoded by treating s_3 as noise. At STA 3, the strong undesired signal s_1 and s_2 can be first decoded and removed successively. After that, the desired signal s_3 can be decoded in a conventional way.

As shown in this example, NOMA can enable concurrent data transmissions for STA 2 and STA 3 without causing much performance degradation for STA 1. Such a multi-user communication approach provides the AP with a new level of flexibility for its resource allocation and user scheduling, which can be leveraged for network throughput maximization.

Design Objectives. We consider the WLAN as shown in Fig. 4.1. We aim at developing a practical NOMA scheme to maximize the weighted sum rate of the users. For fairness, weak users have larger weights while strong users have relatively small weights. To do so, several questions remain open and needed to be addressed: (i) How to design the precoder for each data stream at the AP is a non-trivial task. When the AP has one antenna, the precoders degrade to complex coefficients, which represent the power allocation at the AP. When the AP has multiple antennas, the precoders determine not only the power allocation but also beam steering at the AP. The design of the optimal precoders at the AP involves both Signal-to-Interference-and-Noise Ratio (SINR) and Interference-to-Signal-and-Noise Ratio (ISNR) constraints at each station, making it challenging to reach the optimality. It is noteworthy that we design a unique precoder for each individual data stream in order to pursue performance optimality. Compared to the approach that separates the power allocation and beam-steering design, our joint design promises better possible performance, especially for the networks with a small number of antennas and a small number of users. (ii) How to design a practical scheme to enable downlink NOMA in WLANs is another challenging problem. To support downlink NOMA, the AP needs the knowledge of CSI to compute the precoders; each station needs to perform signal detection in the face of inter-user interference. All these tasks require a sophisticated design of protocols and algorithms that are amenable to practical implementation.

4.4 Precoder Design for Downlink NOMA

In this section, we first formulate the precoder design problem in downlink NOMA transmission of WLANs. Then, we convexify the non-convex constraints and propose an iterative algorithm to pursue a feasible solution. Finally, we offer discussions on the proposed algorithm.

4.4.1 Mathematical Formulation

Consider the downlink data transmission in the WLAN shown in Fig. 4.1. Denote $\mathbf{h}_i \in \mathbb{C}^{1\times M}$ as the channel from theAP to STA i, which includes the effects of path loss, shadow fading, and fast fading. Owing to the large difference in path losses, we assume that $\|\mathbf{h}_1\| \leq \|\mathbf{h}_2\| \leq \ldots \leq \|\mathbf{h}_N\|$. At theAP, denote s_i as the signal intended for STA i, with $\mathbb{E}(|s_i|^2) = 1$; denote $\mathbf{v}_i \in \mathbb{C}^{M\times 1}$ as the precoding vector of this signal. The transmit signals at theAP, which is denoted by $\mathbf{x}$, can be written as $\mathbf{x} = \sum_{j\in\mathcal{N}} \mathbf{v}_j s_j$. Then, the received signal at STA $i \in \mathcal{N}$ can be written as:

$$y_i = \mathbf{h}_i \sum_{j=1}^{N} \mathbf{v}_j s_j + n_i, \quad i \in \mathcal{N}\,. \tag{4.1}$$

where $n_i \sim \mathcal{CN}(0, \sigma_i^2)$ is additive white Gaussian noise.

Transmit Power Constraint. In practice, the transmit power of theAP is bounded by its maximum power budget, which we denote as P_{ap}. This constraint can be written as:

$$\sum_{i=1}^{N} \|\mathbf{v}_i\|^2 \leq P_{\text{ap}}\,. \tag{4.2}$$

SIC and SINR Constraints. At STA $i \in \{2, 3, \cdots, N\}$, we employ SIC to mitigate the strong interference $[s_1,\ s_2,\ \cdots,\ s_{i-1}]$ and decode the desired signal s_i by treating interference $[s_{i+1}, s_{i+2}, \cdots, s_N]$ as noise. Specifically, we first decode the undesired signal s_1 by treating signals $[s_2, s_3, \cdots, s_N]$ as noise. Based on the estimated signal $\hat{s}_1$, we can remove the effect of undesired signal s_1 and the resulting signal can be written as $y_i^{[1]} = \mathbf{h}_i \sum_{j=2}^{N} \mathbf{v}_j s_j + n_i$, where $y_i^{[1]}$ denotes the remaining signal after the first iteration of SIC. By the same token, we can continue to remove undesired signals $[s_2, s_3, \cdots, s_{i-1}]$ sequentially. After removing the undesired signals,

Table 4.1: MCS specification in IEEE 802.11ac [67].

SINR (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
η)	0	0.5	1	1.5	2	3	4	4.5	5	6	20/3
a_i	0.079	0.073	0.050	0.025	0.018	0.012	0.004	0.002	0.001	0.001	0
b_i	0	0.018	0.247	0.747	0.996	1.495	2.746	3.395	3.996	4.075	6.666

we can decode the intended signal s_i by treating $[s_{i+1}, s_{i+2}, \cdots, s_N]$ as noise. Suppose that the SIC procedure is ideal. By denoting $\mathrm{SINR}_{i,j}$ as the SINR in the jth iteration of SIC at STA i, we have

$$\mathrm{SINR}_{i,j} = \frac{\left|\mathbf{h}_i \mathbf{v}_j\right|^2}{\sum_{k=j+1}^{N} \left|\mathbf{h}_i \mathbf{v}_k\right|^2 + \sigma_i^2}, \quad i \in \mathcal{N}, 1 \leq j \leq i. \tag{4.3}$$

By defining $\gamma_{i,j}$ as a non-negative variable less than or equal to $\mathrm{SINR}_{i,j}$, we have

$$\gamma_{i,j} \leq \frac{\left|\mathbf{h}_i \mathbf{v}_j\right|^2}{\sum_{k=j+1}^{N} \left|\mathbf{h}_i \mathbf{v}_k\right|^2 + \sigma_i^2}, \quad i \in \mathcal{N}, 1 \leq j \leq i. \tag{4.4}$$

Data Rate Constraints. In the SIC procedure, STA i needs to decode signals $[s_1, s_2, \cdots, s_i]$ sequentially. When decoding signal s_j ($1 \leq j \leq i$), we known that its SINR is greater than or equal to $\gamma_{i,j}$. To ensure that STA i can successfully decode s_j, the data rate of signal s_j is determined by this SINR value. Theoretically, the relationship between the maximum achievable data rate and the given SINR is governed by Shannon capacity. Denote r_j as the data rate from the AP to STA j in 1 Hz. Then, the achievable data rate constraints can be expressed as:

$$r_j \leq \log_2(1 + \gamma_{i,j}), \quad i \in \mathcal{N}, 1 \leq j \leq i. \tag{4.5}$$

However, Shannon capacity is far from being reached by current WLANs' technologies. It is highly inaccurate to characterize the relationship between the achievable data rate and the SINR in WLANs. In real wireless systems, adaptive MCS is typically used to adjust the data rate based on the SINR value. Table 4.1 lists the MCS selection criteria that are specified in IEEE 802.11ac

standard [67]. Fig. 4.3 shows the gap between Shannon capacity and the data rate achieved by the adaptive MCS approach. It is evident that the gap is large. This indicates that Shannon capacity is not a good formula to compute the achievable data rate in real WLANs. To enhance the practicality of our results, we employ the adaptive MCS approach to calculate the achievable data rate for a given SINR value. However, their relation is expressed as a staircase function, which is non-convex. To ease our optimization problem, we approximate this non-convex region by the following linear constraints:

$$r_j \leq a_k \gamma_{i,j} + b_k, \quad i \in \mathcal{N}, 1 \leq j \leq i, 1 \leq k \leq 11, \tag{4.6}$$

where a_k and b_k are constants given in Table 4.1. We can see from Fig. 4.3 that, compared to (4.5), (4.6) is much more accurate to compute MCS-based achievable data rate in real WLANs. It is worth pointing out that the values of a_k and b_k in Table 4.1 were derived from the MCS specified in IEEE 802.11ac. If we want to apply this method to other networks such as IEEE 802.11ax, the values of a_k and b_k should be updated according to the MCS specified in corresponding standards.

Optimization Formulation. Based on the above constraints, we can formulate the NOMA problem as an optimization problem. Here, we consider the weighted sum rate as the objective function. Other objective functions (e.g., maximizing the minimum data rate) can be formulated in the same way. Denote w_j as the given weight for STA $j \in \mathcal{N}$. These weights are used to prioritize the service for the STAs and maintain the fairness among the STAs. Generally speaking, a STA with strong channel should be given a small weight, while a STA with weak channel should be given a large weight. Suppose that the STAs' weights are pre-defined. Then, the objective function can be written as: $\sum_{j \in \mathcal{N}} w_j r_j$. Then, the optimization problem, which we denote as OPT-NOMA, can be written as:

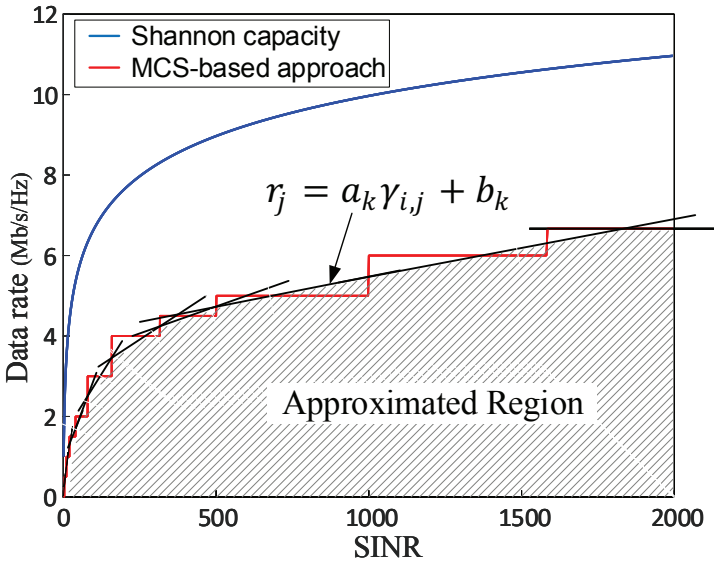


Figure 4.3: The gap between Shannon capacity and the data rate achieved by MCS-based approach.

$$\max \quad \sum_{j\in\mathcal{N}} w_j r_j \tag{4.7a}$$

$$\text{s.t.} \quad r_j \le a_k \gamma_{i,j} + b_k, \quad i\in\mathcal{N}, 1\le j\le i, 1\le k\le 11; \tag{4.7b}$$

$$\gamma_{i,j} \le \frac{\left|\mathbf{h}_i \mathbf{v}_j\right|^2}{\sum\limits_{k=j+1}^{N} \left|\mathbf{h}_i \mathbf{v}_k\right|^2 + \sigma_i^2}, \quad i\in\mathcal{N}, 1\le j\le i; \tag{4.7c}$$

$$\sum_{i\in\mathcal{N}} \|\mathbf{v}_i\|^2 \le P_{\text{ap}}\,; \tag{4.7d}$$

where r_i, $\gamma_{i,j}$, and $\mathbf{v}_i$ are optimization variables; w_j, a_k, b_k, $\mathbf{h}_i$, and σ_i, and P_{ap} are given parameters. Note that we changed the equality in (4.4) to the inequality in (4.7c). It can be verified that this operation does not alter the optimal value.

Compared to many existing optimization formulations of NOMA (see, e.g., [55, 210]), one may notice that our formulation does not have explicit constraints to represent the decoding order

requirements of SIC. Actually, constraints (4.7b) and (4.7c) in our formulation can ensure the success of SIC at every station. Consider STA i for example. Signal s_i is its desired signal and s_j ($1 \leq j < i$) is interference that should be removed. The combination of (4.7b) and (4.7c) ensures that interference s_j ($1 \leq j < i$) can be decoded and subtracted. It also ensures that the desired signal s_i can be decoded after subtracting interference s_j ($1 \leq j < i$). As such, our formulation can ensure the success of SIC, albeit without explicit constraints to enforce the decoding order requirements of SIC. Moreover, our formulation is in a simpler format and relatively easier to solve compared to the existing ones.

OPT-NOMA is a non-convex problem, which is NP-hard in general. There is no efficient algorithm that can find its optimal solution in polynomial time. In the rest of this section, we delve into the development of a tractable approach to pursue a suboptimal solution to OPT-NOMA via disciplinary convexification.

4.4.2 Constraint Relaxation via Disciplinary Convexification

In OPT-NOMA, (4.7a) and (4.7b) are linear and easy to handle by optimization solvers; however, (4.7c) and (4.7d) are not. In what follows, we focus on these two nonlinear constraints.

Constraint (4.7d)**.** This constraint is convex, but in an indisciplined form. To transform it to a disciplined convex constraint, we rewrite it as a *Lorentz cone* [23, Ch. 2]:

$$\left\| \left[\mathbf{v}_1^{\mathsf{T}}, \mathbf{v}_2^{\mathsf{T}}, \ldots, \mathbf{v}_N^{\mathsf{T}} \right] \right\| \leq \sqrt{P_{\text{ap}}} \, . \tag{4.8}$$

Constraint (4.7c)**.** This constraint generates a set of $\frac{N(N+1)}{2}$ non-convex inequations and needs to be convexified into a disciplined form. To convexify (4.7c), we introduce an auxiliary variable $z_{i,j}$ and define $z_{i,j} \geq \sum_{k=j+1}^{N} |\mathbf{h}_i \mathbf{v}_k|^2 + \sigma_i^2$. Then, (4.7c) can be equivalently broken into

the following two sets of constraints:

$$\gamma_{i,j} z_{i,j} \leq \left|\mathbf{h}_i \mathbf{v}_j\right|^2 , \qquad i \in \mathcal{N}, 1 \leq j \leq i; \tag{4.9a}$$

$$\sum_{k=j+1}^{N} |\mathbf{h}_i \mathbf{v}_k|^2 \leq z_{i,j} - \sigma_i^2 , \qquad i \in \mathcal{N}, 1 \leq j \leq i. \tag{4.9b}$$

To convexify (4.9), we first focus on (4.9a) and then on (4.9b). For (4.9a), it is a non-convex constraint because it has a quadratic term on its right-hand side (RHS). To untangle this problem, we employ tangent point and Taylor expansion to approximate the quadratic term with an appropriate affine [66, 210]. To illustrate this idea, let us consider a differentiable convex function $f(\mathbf{v})$ for example. At any feasible point, say $\tilde{\mathbf{v}}$, a tangent function $g(\mathbf{v}, \tilde{\mathbf{v}})$ can be defined such that $f(\mathbf{v}) \geq g(\mathbf{v}, \tilde{\mathbf{v}})$, and the equality holds at $\mathbf{v} = \tilde{\mathbf{v}}$. The tangent function $g(\mathbf{v}, \tilde{\mathbf{v}})$ is a minorant of $f(\mathbf{v})$, and the solution to the approximated problem using tangent point $\tilde{\mathbf{v}}$ will majorize the minorant [66]. To further make this constraint disciplinary, the first-order Taylor expansion of $f(\mathbf{v})$ can be used as the tangent function since it removes the high-order nondisciplinary components of $f(\mathbf{v})$. Using the first-order Taylor expansion, the tangent function can be written as:

$$g(\mathbf{v}, \tilde{\mathbf{v}}) = f(\tilde{\mathbf{v}}) + \nabla f(\tilde{\mathbf{v}})^{\mathsf{H}} (\mathbf{v} - \tilde{\mathbf{v}}). \tag{4.10}$$

We apply this idea to the RHS of (4.9a). If $f_i(\mathbf{v}_j) = \left|\mathbf{h}_i \mathbf{v}_j\right|^2$ is defined, then we have $\nabla f_i(\mathbf{v}_j) = 2\mathbf{h}_i \mathbf{h}_i^{\mathsf{H}} \mathbf{v}_j$. The tangent function at $\tilde{\mathbf{v}}_j$ can be written as:

$$\begin{aligned} g_i(\mathbf{v}_j, \tilde{\mathbf{v}}_j) &= f_i(\tilde{\mathbf{v}}_j) + \nabla f_i(\tilde{\mathbf{v}}_j)^{\mathsf{H}} (\mathbf{v}_j - \tilde{\mathbf{v}}_j) \\ &= \mathbf{h}_i \tilde{\mathbf{v}}_j \tilde{\mathbf{v}}_j^{\mathsf{H}} \mathbf{h}_i^{\mathsf{H}} + 2\mathbf{h}_i \mathbf{h}_i^{\mathsf{H}} \tilde{\mathbf{v}}_j (\mathbf{v}_j - \tilde{\mathbf{v}}_j) \end{aligned}$$

$$= 2\mathbf{h}_i\tilde{\mathbf{v}}_j\mathbf{v}_j^{\mathsf{H}}\mathbf{h}_j^{\mathsf{H}} - \mathbf{h}_i\tilde{\mathbf{v}}_j\tilde{\mathbf{v}}_j^{\mathsf{H}}\mathbf{h}_i^{\mathsf{H}}. \tag{4.11}$$

Given that both sides of original constraint (4.9a) are real values, we use $Re\big(g_i(\mathbf{v}_j, \tilde{\mathbf{v}}_j)\big)$ as the tangent function for $f_i(\mathbf{v}_j)$. Then, the RHS of (4.9a) can be approximated by

$$\left|\mathbf{h}_i\mathbf{v}_j\right|^2 \approx Re\big(g_i(\mathbf{v}_j, \tilde{\mathbf{v}}_j)\big) = 2Re(\mathbf{h}_i\tilde{\mathbf{v}}_j\mathbf{v}_j^{\mathsf{H}}\mathbf{h}_j^{\mathsf{H}}) - \mathbf{h}_i\tilde{\mathbf{v}}_j\tilde{\mathbf{v}}_j^{\mathsf{H}}\mathbf{h}_i^{\mathsf{H}}\,, \tag{4.12}$$

We apply the same method to convexify the left-hand side (LHS) of (4.9a). To convexify the product of two variables, we define a bivariate function $f\left(\gamma, z\right) = \gamma z$. It is neither convex nor concave since its Hessian matrix is neither positive semidefinite nor negative semidefinite, and it also has a saddle point at $\gamma = z = 0$. However, this function can be expressed as a summation of a convex function and a concave one, i.e., $f\left(\gamma, z\right) = f_1\left(\gamma, z\right) + f_2\left(\gamma, z\right)$, where $f_1(\gamma, z) = \frac{(\gamma+z)^2}{4}$ and $f_2(\gamma, z) = -\frac{(\gamma-z)^2}{4}$. To convexify $f\left(\gamma, z\right)$, it suffices to pursue the idea of using a tangent function for its concave component. Since $f_2\left(\gamma, z\right)$ is a differentiable concave function, tangent function $g\left(\gamma, z, \tilde{\gamma}, \tilde{z}\right)$ is a majorant of $f_2\left(\gamma, z\right)$. Indeed, $f_2\left(\gamma, z\right) \leq g\left(\gamma, z, \tilde{\gamma}, \tilde{z}\right)$. This majorant can be expressed as a tangent function at point $(\tilde{\gamma}, \tilde{z})$ as:

$$g\left(\gamma, z, \tilde{\gamma}, \tilde{z}\right) = \frac{1}{2}\left(\tilde{\gamma} - \tilde{z}\right)\left(\gamma - \tilde{\gamma} + \tilde{z} - z\right) - \frac{1}{4}\left(\tilde{\gamma} - \tilde{z}\right)^2. \tag{4.13}$$

Based on the tangent function in (4.13), we can approximate $f\left(\gamma, z\right) = \gamma z$ using $f_1\left(\gamma, z\right) + g\left(\gamma, z, \tilde{\gamma}, \tilde{z}\right)$. Then, the LHS of (4.9a) can be approximated by

$$\begin{aligned}\gamma_{i,j}z_{i,j} \approx & f_1\left(\gamma_{i,j}, z_{i,j}\right) + g\left(\gamma_{i,j}, z_{i,j}, \tilde{\gamma}_{i,j}, \tilde{z}_{i,j}\right)\\ = & \frac{1}{4}\left(\gamma_{i,j} + z_{i,j}\right)^2 - \frac{1}{4}\left(\tilde{\gamma}_{i,j} - \tilde{z}_{i,j}\right)^2\end{aligned}$$

$$-\frac{1}{2}\left(\tilde{\gamma}_{i,j}-\tilde{z}_{i,j}\right)\left(\gamma_{i,j}-\tilde{\gamma}_{i,j}+\tilde{z}_{i,j}-z_{i,j}\right). \tag{4.14}$$

Based on the relaxations in (4.12) and (4.14), the non-convex constraint (4.9a) can be approximated by the following convex constraint:

$$\begin{aligned}&\frac{1}{4}\left(\gamma_{i,j}+z_{i,j}\right)^2-\frac{1}{4}\left(\tilde{\gamma}_{i,j}-\tilde{z}_{i,j}\right)^2\\&-\frac{1}{2}\left(\tilde{\gamma}_{i,j}-\tilde{z}_{i,j}\right)\left(\gamma_{i,j}-\tilde{\gamma}_{i,j}+\tilde{z}_{i,j}-z_{i,j}\right)\\\leq&2Re(\mathbf{h}_i\tilde{\mathbf{v}}_j\mathbf{v}_j^{\mathsf{H}}\mathbf{h}_j^{\mathsf{H}})-\mathbf{h}_i\tilde{\mathbf{v}}_j\tilde{\mathbf{v}}_j^{\mathsf{H}}\mathbf{h}_i^{\mathsf{H}},\quad i\in\mathcal{N},1\leq j\leq i.\end{aligned} \tag{4.15}$$

So far, we have convexified constraint (4.9a). Now, we focus on (4.9b), which is a *restricted hyperbolic* constraint. This constraint is convex but indisciplined. To make it disciplined, we first introduce an existing technique and then apply it to transform (4.9b). Consider an indisciplined convex constraint $\theta^2\leq\alpha\beta$, $\alpha,\beta\in\mathbb{R}^+$ and $\theta\in\mathbb{R}$. Based on [11], we have:

$$\theta^2\leq\alpha\beta\Longleftrightarrow\left\|\left[\theta,\frac{(\alpha-\beta)}{2}\right]\right\|\leq\frac{(\alpha+\beta)}{2}, \tag{4.16}$$

where $\Longleftrightarrow$ means that the two sides are equivalent, and the RHS is a disciplined convex constraint. By taking advantage of this result, indisciplined convex constraint (4.9b) can be equivalently transformed to a disciplined convex constraint as follows:

$$\left\|\left[\left|\mathbf{h}_i\mathbf{v}_{j+1}\right|,\cdots,|\mathbf{h}_i\mathbf{v}_N|,\frac{\left(z_{i,j}-\sigma_i^2-1\right)}{2}\right]\right\|\leq\frac{\left(z_{i,j}-\sigma_i^2+1\right)}{2},\quad i\in\mathcal{N},1\leq j\leq i, \tag{4.17}$$

The relaxed problem using the convexified constraints, which we denote as OPT-NOMA-RELAX, can be written as:

$$\max \quad \sum_{i \in \mathcal{N}} w_i r_i$$

$$\text{s.t.} \quad \text{(4.7b), (4.8), (4.15), and (4.17).}$$

OPT-NOMA-RELAX is a second-order cone programming (SOCP) problem, which can be solved in polynomial time by off-the-shelf optimization solvers such as CVX and CVXOPT [13].

4.4.3 Our Proposed Algorithm

Based on OPT-NOMA-RELAX, we propose an algorithm to solve the original problem OPT-NOMA. The proposed algorithm is an iterative algorithm. In each iteration, we solve OPT-NOMA-RELAX by taking the output results from the previous iteration as the input parameters (tangent points for convexification). The iterative algorithm terminates if the increase of the objective value is less than a pre-defined threshold (ϵ) or the number of iterations reaches a pre-defined bound (N_{iter}). For notational simplicity, when solving OPT-NOMA-RELAX in iteration l, we denote $\left[\underline{\tilde{\mathbf{v}}}^{[l-1]}, \underline{\tilde{\gamma}}^{[l-1]}, \underline{\tilde{\mathbf{z}}}^{[l-1]}\right]$ as the input parameters (the tangent points for convexification) and $\left[\underline{\mathbf{v}}^{[l]}, \underline{\gamma}^{[l]}, \underline{\mathbf{z}}^{[l]}, \underline{\mathbf{r}}^{[l]}\right]$ as the output results (the optimal solution to OPT-NOMA-RELAX). Alg. 4.1 presents our proposed algorithm.

For such an iterative algorithm, an important question is how to construct an appropriate initial tangential set for the OPT-NOMA-RELAX problem in the first iteration. It is well known that the performance of many optimization problems is heavily reliant on their initial search points. A good initial point significantly accelerates the search process and therefore remarkably reduces the computational time of the algorithm. In light of this, we develop an algorithm to construct

Algorithm 4.1 Solving OPT-NOMA.

Inputs: Network parameters $\mathbf{h}_i$, σ_i, $\mathcal{N}$, w_j, P_{ap}, a_k, b_k, and threshold ϵ;
Outputs: A solution to OPT-NOMA $\left[\underline{\mathbf{v}}^*, \underline{\gamma}^*, \underline{\mathbf{z}}^*, \underline{\mathbf{r}}^*\right]$;

Compute initial tangent points $\left[\tilde{\underline{\mathbf{v}}}^{[0]}, \tilde{\underline{\gamma}}^{[0]}, \tilde{\underline{\mathbf{z}}}^{[0]}\right]$ using Alg. 4.2;
Specify the max number of iterations (e.g., $N_{\text{iter}} = 100$);
for ($l = 1$; $l \leq N_{\text{iter}}$; l++) **do**
    $\left[\underline{\mathbf{v}}^{[l]}, \underline{\gamma}^{[l]}, \underline{\mathbf{z}}^{[l]}, \underline{\mathbf{r}}^{[l]}\right] \leftarrow$ solving OPT-NOMA-RELAX using $\left[\tilde{\underline{\mathbf{v}}}^{[l-1]}, \tilde{\underline{\gamma}}^{[l-1]}, \tilde{\underline{\mathbf{z}}}^{[l-1]}\right]$;
    $\left[\tilde{\underline{\mathbf{v}}}^{[l]}, \tilde{\underline{\gamma}}^{[l]}, \tilde{\underline{\mathbf{z}}}^{[l]}\right] \leftarrow \left[\underline{\mathbf{v}}^{[l]}, \underline{\gamma}^{[l]}, \underline{\mathbf{z}}^{[l]}\right]$
    if $\left\|\underline{\mathbf{r}}^{[l]} - \underline{\mathbf{r}}^{[l-1]}\right\| < \epsilon$ **then**
        Break;
$\left[\underline{\mathbf{v}}^*, \underline{\gamma}^*, \underline{\mathbf{z}}^*, \underline{\mathbf{r}}^*\right] \leftarrow \left[\underline{\mathbf{v}}^{[l]}, \underline{\gamma}^{[l]}, \underline{\mathbf{z}}^{[l]}, \underline{\mathbf{r}}^{[l]}\right]$;

a good initial search point for the OPT-NOMA-RELAX problem in the first iteration. Alg. 4.2 shows our proposed algorithm. In this algorithm, we first randomly generate a set of vectors for $\tilde{\underline{\mathbf{v}}}^{[0]}$ and then normalize its amplitude to meet the power constraint. Upon initializing $\tilde{\underline{\mathbf{v}}}^{[0]}$, we then calculate $\tilde{\underline{\gamma}}^{[0]}$ and $\tilde{\underline{\mathbf{z}}}^{[0]}$ based on their respective constraints. In this process, a small number ε is used to ensure the strict feasibility of the tangential set and maximize its corresponding objective value.

4.4.4 Discussions on the Proposed Algorithm

Alg. 4.1 and Alg. 4.2 constitute our proposed algorithm to solve the optimization problem OPT-NOMA. We have the following remarks for the proposed algorithm:

Remark 1 (Feasibility). Our proposed algorithm yields a feasible solution to the original optimization problem OPT-NOMA. We pinpoint this by arguing that any feasible solution to OPT-NOMA-RELAX is also feasible to OPT-NOMA. Comparing the two optimization problems, we can see that the different constraints are (4.7c) in OPT-NOMA and (4.15) and (4.17) in OPT-NOMA-RELAX; other constraints are the same. Now, let us focus on these two constraints.

Algorithm 4.2 Constructing the first tangential set for OPT-NOMA-RELAX.

Inputs: Network parameters $\mathbf{h}_i$, σ_i, $\mathcal{N}$, w_j, P_{ap}, and safety gap ε;

Outputs: An initial tangential set for OPT-NOMA-RELAX $\left[\underline{\tilde{\mathbf{v}}}^{[0]}, \underline{\tilde{\gamma}}^{[0]}, \underline{\tilde{\mathbf{z}}}^{[0]}\right]$;

1: Generate random values for $\left[\hat{\mathbf{v}}_1^{\mathsf{T}}, \hat{\mathbf{v}}_2^{\mathsf{T}}, \cdots, \hat{\mathbf{v}}_N^{\mathsf{T}}\right]$

2: **for** $(i = 1; i \leq N; i++)$ **do**

3: $\quad \mathbf{v}_i = \sqrt{\frac{(1-\varepsilon)P_{\text{ap}}}{\sum_{j=1}^{N} \left\|\hat{\mathbf{v}}_j\right\|^2}} \hat{\mathbf{v}}_i$

4: **for** $(i \in \mathcal{N}, 1 \leq j \leq i)$ **do**

5: $\quad$ Calculate $z_{i,j} = (1+\varepsilon)\left(\sigma_i^2 + \sum\limits_{k=j+1}^{N} |\mathbf{h}_i \mathbf{v}_k|^2\right)$

6: $\quad$ Calculate $\gamma_{i,j} = \frac{(1-\varepsilon)\left|\mathbf{h}_i \mathbf{v}_j\right|^2}{z_{i,j}}$

7: $\left[\underline{\tilde{\mathbf{v}}}^{[0]}, \underline{\tilde{\gamma}}^{[0]}, \underline{\tilde{\mathbf{z}}}^{[0]}\right] \leftarrow \left[\underline{\mathbf{v}}, \underline{\gamma}, \underline{\mathbf{z}}\right]$

The relaxation from (4.7c) to (4.15) and (4.17) is actually a minorization-majorization process [66]. That is, we replaced a concave function on the LHS with a tangent affine function and replaced a convex function on the RHS with a tangent affine function (see (4.12) and (4.14) respectively). Based on the properties of concave and convex functions, we know that (4.15) and (4.17) are more restrictive than (4.7c). In other words, a solution satisfying (4.15) and (4.17) certainly satisfies (4.7c). Therefore, we conclude that a solution feasible to OPT-NOMA-RELAX is also feasible to OPT-NOMA. According to Alg. 4.1 and Alg. 4.2, it is easy to see that the generated solution is feasible to OPT-NOMA-RELAX, which is also feasible to the original optimization problem OPT-NOMA.

Remark 2 (Convergence). Our proposed algorithm converges to a stationary point. Since the feasible region of OPT-NOMA-RELAX is expanding over the iterations in Alg. 4.1 [66], the value returned by the objective function is non-decreasing. Moreover, the solution yielded by each iteration (from solving OPT-NOMA-RELAX) is in the feasible region of OPT-NOMA. Because of the same objective function on both problems, Alg. 1 converges to a stationary point, which could be

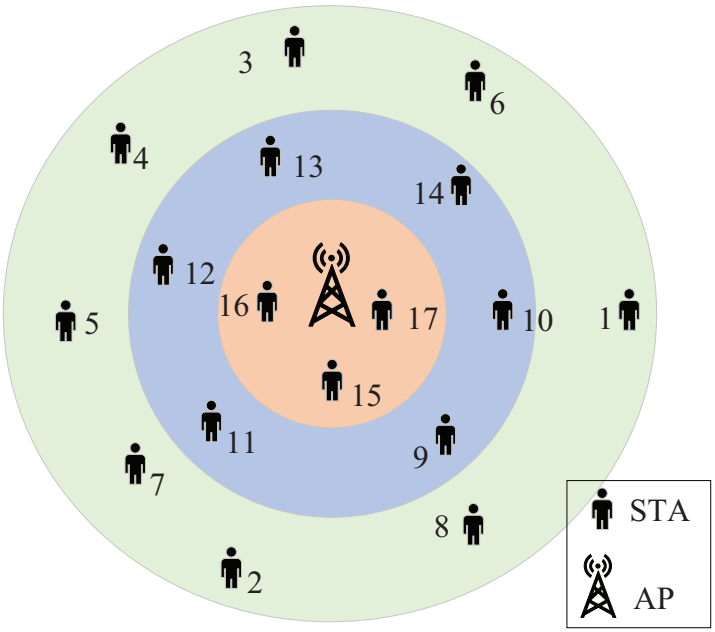


Figure 4.4: An example of WLAN that has a set of widely distributed stations.

either a global or local optimal point.

Remark 3 (Computational Complexity). Our proposed algorithm (Alg. 4.1) has polynomial-time computational complexity. Alg. 4.1 is an iterative algorithm. In each iteration, its main work is solving the OPT-NOMA-RELAX problem (an SOCP problem). Given $M \leq N$ in NOMA, the complexity of each iteration is $O(N^6)$ [120]. Since the number of iterations in Alg. 1 is bounded by N_{iter}, the overall computational complexity of Alg. 4.1 is $O(N_{\text{iter}} \cdot N^6)$.

Remark 4 (Imperfect CSI). In the formulation of OPT-NOMA, we assumed perfect CSI for the design of precoders. However, in real systems, perfect CSI may not be available. In that case, we can use the measured (imperfect) CSI as the input to compute the precoders. Apparently, the imperfection of CSI may lead to a performance degradation.

4.5 A Downlink NOMA Scheme for WLAN

In this section, we propose a practical scheme based on the precoder design in the previous section to enable downlink NOMA transmissions in WLANs. We consider a WLAN as shown Fig. 4.4, which comprises an AP and a set of widely distributed stations (STAs). Denote $\mathcal{S}$ as the set of STAs in the network, with $S = |\mathcal{S}|$. The STAs are sorted in non-decreasing order based on their channel quality (i.e., $\|\mathbf{h}_i\|, i \in \mathcal{S}$). STA 1 is the weakest station and STA S is the strongest one. For such a network, we propose a downlink NOMA framework to support multi-user communications by leveraging the precoder optimization approach in Section 4.4.

4.5.1 User Grouping at AP

We assume that the AP is responsible for user scheduling and grouping for the downlink transmissions. To perform user grouping, the AP needs to determine the number of stations in one group. Theoretical exploration of this problem requires an exhaustive search to identify the best grouping strategy that leads to the maximum network throughput. However, such an approach is overly complicated and not amenable to practical implementation. Therefore, we resort to a heuristic design for user grouping. In what follows, we first study the user grouping in a simple WLAN and then propose a heuristic algorithm for user grouping in a generic WLAN.

User Pairing in SISO Network. We consider a WLAN as shown in Fig. 4.4 and assume that each node (AP or STA/user) has a single antenna. We also assume that each group has two users in NOMA transmission for simplicity. Denote h_w and h_s as the channel coefficients of the weak and strong users in a group, respectively. Denote $p(h_w, h_s)$ as the normalized portion of AP's power allocated to the strong user's message. Based on the notion of NOMA, the AP's

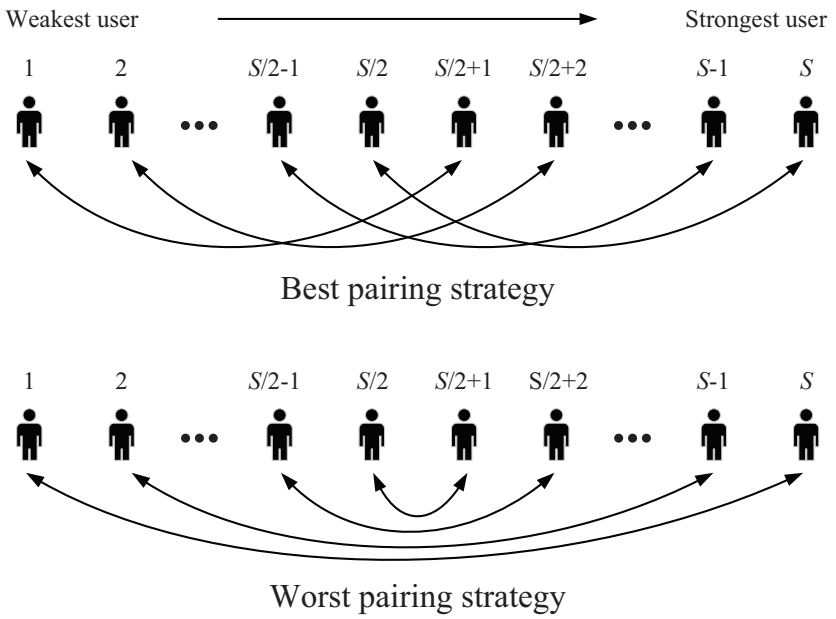


Figure 4.5: Two pairing strategies in NOMA communications.

power allocation for NOMA transmission should have the following property: $p(h_w, h_s)$ is a non-increasing function with respect to $|h_s| \,/\, |h_w|$. Based on this property, we have the following proposition:

Proposition 1. Suppose that the objective is to maximize the weighted sum rate of all users and that round-robin scheduler is used for the paired users. Then, the best pairing strategy is $(i, S/2{+}i)$, and the worst pairing strategy is $(i, S+1-i)$, for $1 \leq i \leq S/2$, as illustrated in Fig. 4.5.

Proof. Consider the pairing strategies in Fig. 4.5. Suppose that the paired users are scheduled in the round-robin way over a set of time slots and that r_{bps} denotes the weighted sum rate yielded by the $(i, S/2+i)$ pairing strategy. Then, we have

$$r_{\mathrm{bps}} = \sum_{i=1}^{S/2} \left[w_1 \log_2 \left(1 + \frac{(1-\alpha_i)\,|h_i|^2}{\alpha_i\,|h_i|^2 + \sigma^2} \right) + w_2 \log_2 \left(1 + \frac{\alpha_i \left| h_{S/2+i} \right|^2}{\sigma^2} \right) \right], \tag{4.18}$$

where $\alpha_i = p(h_i, h_{S/2+i})$, and $(1-\alpha_i)$ is then the normalized portion of AP's transmit power

for the weak user, σ^2 is the normalized power of noise (w.r.t. the signal power), and w_1 and w_2 denote the weight assigned to the weak and strong users in a group, respectively. To show that the $(i, S/2+i)$ pairing strategy yields the highest weighted sum rate among all possible pairing strategies, we argue that any permutation over this pairing strategy would lead to a decrease in the weighted sum rate. Without loss of generality, we assume that the permutation occurs for user pairs $(1, S/2+1)$ and $(2, S/2+2)$. After permutation, the resulting pairs are $(1, S/2+2)$ and $(2, S/2+1)$. For the permuted user pairs, we let $\alpha_1' = p(h_1, h_{S/2+2})$ and $\alpha_2' = p(h_2, h_{S/2+1})$. Then, the change of weighted sum rate from the permutation can be written as follows:

$$\begin{aligned}\Delta r = r_{bps} - r_{perm} = {} & w_1 \log_2 \left(1 + \frac{(1-\alpha_1)\left|h_1\right|^2}{\alpha_1 \left|h_1\right|^2 + \sigma^2}\right) + w_2 \log_2 (1 + \frac{\alpha_1}{\sigma^2}\left|h_{S/2+1}\right|^2) \\ & + w_1 \log_2 \left(1 + \frac{(1-\alpha_2)\left|h_2\right|^2}{\alpha_2 \left|h_2\right|^2 + \sigma^2}\right) + w_2 \log_2 (1 + \frac{\alpha_2}{\sigma^2}\left|h_{S/2+2}\right|^2) \\ & - w_1 \log_2 \left(1 + \frac{(1-\alpha_1')\left|h_1\right|^2}{\alpha_1' \left|h_1\right|^2 + \sigma^2}\right) - w_2 \log_2 (1 + \frac{\alpha_1'}{\sigma^2}\left|h_{S/2+2}\right|^2) \\ & - w_1 \log_2 \left(1 + \frac{(1-\alpha_2')\left|h_2\right|^2}{\alpha_2' \left|h_2\right|^2 + \sigma^2}\right) - w_2 \log_2 (1 + \frac{\alpha_2'}{\sigma^2}\left|h_{S/2+1}\right|^2), \end{aligned} \tag{4.19}$$

where r_{perm} denotes the weighted sum rate after permutation.

Through algebraic operations, (4.19) can be rewritten as:

$$\begin{aligned}\Delta r = {} & w_1 \log_2 \left(\frac{\alpha_2' \left|h_2\right|^2 + \sigma^2}{\alpha_2 \left|h_2\right|^2 + \sigma^2}\right) - w_1 \log_2 \left(\frac{\alpha_1' \left|h_1\right|^2 + \sigma^2}{\alpha_1 \left|h_1\right|^2 + \sigma^2}\right) + \\ & w_2 \log_2 \left(\frac{\alpha_2 \left|h_{S/2+2}\right|^2 + \sigma^2}{\alpha_1' \left|h_{S/2+2}\right|^2 + \sigma^2}\right) - w_2 \log_2 \left(\frac{\alpha_1 \left|h_{S/2+1}\right|^2 + \sigma^2}{\alpha_2' \left|h_{S/2+1}\right|^2 + \sigma^2}\right). \end{aligned} \tag{4.20}$$

Recall that the users are sorted in increasing order by their channel strength, i.e., $|h_1| \leq |h_2| \leq \left|h_{S/2+1}\right| \leq \left|h_{S/2+2}\right|$. Since $p(h_w, h_s)$ is a non-increasing function with respect to $|h_s| / |h_w|$,

Algorithm 4.3 An algorithm for user grouping.

Inputs: The array of sorted STAs ($\mathcal{S} = \{1, 2, \cdots, S\}$) and each STA's channel ($\mathbf{h}_i$, $i \in \mathcal{S}$);
Outputs: The total number of groups (K) and the generated user groups $\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_K$;

$\Delta q \leftarrow 10$;
$k \leftarrow 0$;
while ($\mathcal{S}$ is not empty) **do**
    k++;
    $\mathcal{G}_k = [\mathcal{S}(1)]$; *// $\mathcal{S}(1)$ is the 1st element of $\mathcal{S}$*
    $q\_value \leftarrow q(\mathcal{S}(1))$;
    for ($l = 2$; $l \leq size(\mathcal{S})$; l++) **do**
        if $q(\mathcal{S}(l)) \geq q\_value + \Delta q$ **then**
            $\mathcal{G}_k \leftarrow [\mathcal{G}_k \ \ \mathcal{S}(l)]$;
            $q\_value \leftarrow q(\mathcal{S}(l))$;
    Remove all elements in $\mathcal{G}_k$ from $\mathcal{S}$;
$K \leftarrow k$;

we have $\alpha_1 \geq \alpha_1'$, $\alpha_2' \geq \alpha_2$, $\alpha_2' \geq \alpha_1$, and $\alpha_2 \geq \alpha_1'$. Then, the following two inequalities are imminent.

$$\frac{\alpha_2' \left|h_2\right|^2 + \sigma^2}{\alpha_2 \left|h_2\right|^2 + \sigma^2} \geq \frac{\alpha_1' \left|h_1\right|^2 + \sigma^2}{\alpha_1 \left|h_1\right|^2 + \sigma^2}, \tag{4.21}$$

$$\frac{\alpha_2 \left|h_{S/2+2}\right|^2 + \sigma^2}{\alpha_1' \left|h_{S/2+2}\right|^2 + \sigma^2} \geq \frac{\alpha_1 \left|h_{S/2+1}\right|^2 + \sigma^2}{\alpha_2' \left|h_{S/2+1}\right|^2 + \sigma^2}. \tag{4.22}$$

Based on (4.20), (4.21), and (4.22), it is evident that $\Delta r \geq 0$. This shows that any permutation on user pairing $(i, S/2 + i)$ decreases the weighted sum rate. We therefore conclude that the $(i, S/2+i)$ pairing strategy yields the highest weighted sum rate. By the same token, we can prove that the $(i, S + 1 - i)$ pairing strategy yields the lowest weighted sum rate. We omit this part to converse space. ■

From Proposition 1, we have the following observations on user pairing: (i) it should try to avoid pairing two users with similar channel quality; and (ii) it should try to maintain a similar channel difference for user pairs.

User Grouping in MISO Network. Based on the above two observations, we propose a

heuristic user grouping algorithm for a generic WLAN. For STA $i \in \mathcal{S}$, we define its channel quality indicator as $q(i) = 20\log_{10}(\|\mathbf{h}_i\|)$, where $\mathbf{h}_i$ is STA i's channel that includes path loss, shadow fading, and fast fading. Based on the channel quality indicator, we use the following rules to devise a user grouping algorithm: (i) The STAs in the same group should have at least Δq channel quality difference, where Δq represents the channel quality difference in decibel and should be adaptively set based on the network environment. In our experiments, extensive measurements of wireless channels in an office building show that the average channel quality difference of two users is about 9.3 dB. Based on this observation, we set $\Delta q = 10$ dB for the user grouping algorithm.

(ii) one STA is associated with only one group. Per these two rules, we propose a greedy algorithm as shown in Alg. 4.3 for user grouping. Essentially, Alg. 4.3 is heuristic. We have the following remarks on it.

Remark 5 (Single STA in a Group): Based on our algorithm, it is apparent that there is no guarantee each group has more than one STA. If a group has only one STA, this means that NOMA is not needed, and OMA can be used for its transmission. Essentially, such a grouping algorithm requires a combination of NOMA and OMA at the PHY layer for data transmission.

4.5.2 A MAC-Layer Protocol for NOMA

If a group includes multiple users (STAs), then NOMA is used to enable concurrent data transmission for the STAs. With a bit abuse of notation, we denote $\{1, 2, \cdots, N\}$ as the STAs in the group under consideration. Fig. 4.6 shows our proposed protocol for NOMA transmission. At high level, it comprises three steps: Channel sounding, NOMA transmission, and acknowledgment. Since the acknowledgment step is straightforward, we focus our discussions on channel sounding and NOMA transmission.

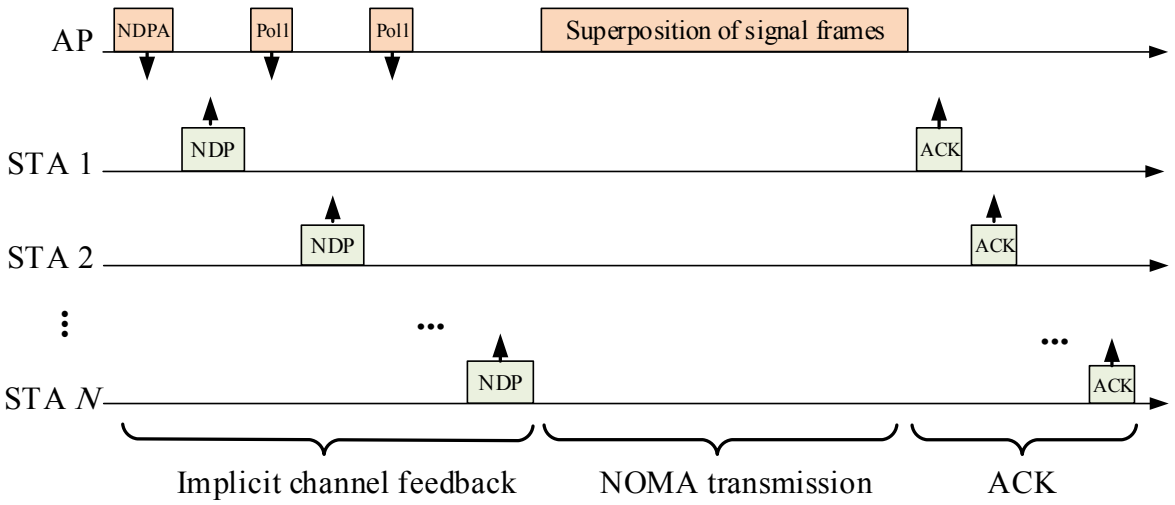


Figure 4.6: A protocol for NOMA transmission in WLANs.

Channel Sounding. To reduce the airtime overhead, we employ an implicit channel feedback mechanism in our protocol by leveraging channel reciprocity. Specifically, the AP first broadcasts a Null Data Packet Announcement (NDPA) to inform the stations of channel sounding and NOMA transmission. Upon reception of the NDPA packet, the stations sequentially respond with a NDP following the poll packets from the AP. The NDP includes the preamble (reference signals) enabling the AP to estimate the uplink channel. At the end of this step, the AP obtains the uplink channels between itself and all the intended stations. The obtained uplink channels will be converted to downlink channels through channel calibration.

In such an implicit channel feedback mechanism, three important problems need to be taken into consideration: (i) For the protocol in Fig. 4.6, the stations should use the same transmit power when transmitting the NDP (e.g., the maximum transmit power specified in the standards). Use of different transmit powers will confuse the AP about the channel quality between itself and the stations, thereby leading to a failure in the downlink NOMA transmission. (ii) Typically, the stations in a WLAN have the same noise power. In some extreme cases where the stations have different noise power, the stations need to feed their noise power back to the AP. This can be

easily done by embedding the noise power information (only a real number) into the NDP when performing uplink channel sounding. (iii) To perform downlink NOMA transmission, the AP actually needs to know the downlink channels information. It is therefore imperative to infer the downlink channels based on the measured uplink channels in our protocol. When the AP has a single antenna, the difference between an uplink channel and its corresponding downlink channel can be represented by a complex number in the mathematical channel model. Such a complex scalar does not affect the NOMA scheduling and transmission results. Therefore, the measured uplink channels can be equivalently treated as downlink channels. When the AP has multiple antennas, the difference between uplink and downlink channels is an array of complex numbers. To compensate the mismatch, a channel calibration procedure is needed at the AP. While there are many calibration methods, we employ the relative calibration method in [150]. This relative calibration method is an internal and standalone calibration method that can be done at the AP without any aid from the stations. In our experiment, we implicitly implement this calibration method to maintain channel reciprocity.

NOMA Transmission and Frame Structure. After obtaining the downlink channel, the AP computes precoders using the proposed method in Section 4.4 and selects an MCS for each STA in the scheduled group. Then, the AP performs downlink NOMA transmission as illustrated in Fig. 4.6. To perform downlink NOMA transmission, we propose a MU-MIMO-like frame structure as shown in Fig. 4.7. The proposed frame structure has three parts: (i) The legacy preamble part comprises a Legacy Short Training Field (L-STF), a Legacy Long Training Field (L-LTF), and a Legacy Signal (L-SIG) field. This part is designed for frame detection and time/frequency synchronization on the STA side. (ii) The reference signal part comprises N precoded L-LTFs, each of which has two identical OFDM symbols. This part is devised for signal detection on the STA side. (iii) The data (payload) part comprises a sequence of OFDM symbols, each of which is

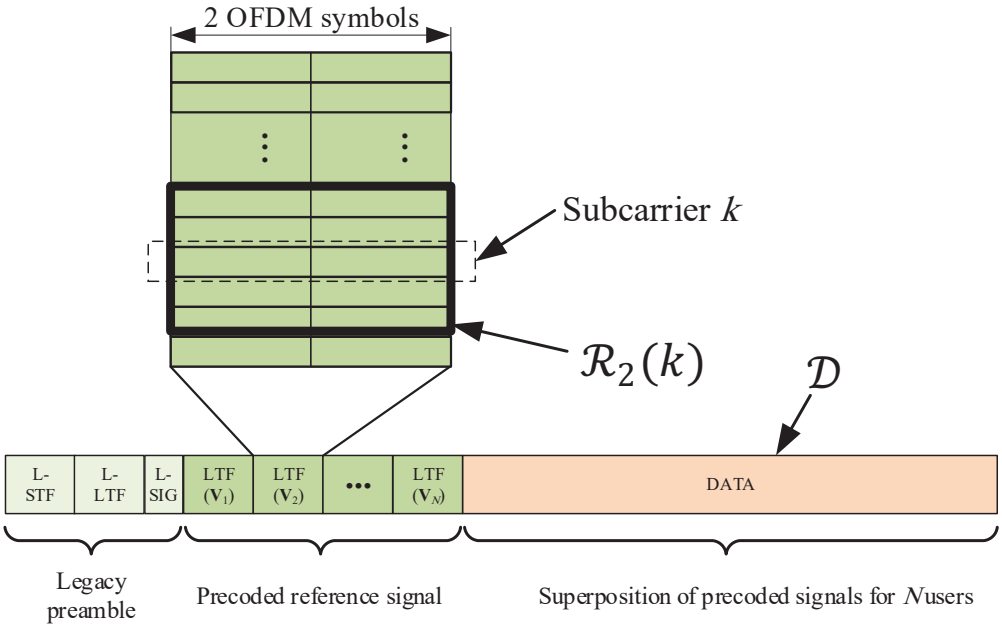


Figure 4.7: Proposed frame structure for NOMA transmission.

a superposition of precoded signals for N stations. In what follows, we propose a new SIC method to decode the desired signal at each STA.

4.5.3 PHY-Layer Signal Processing

At the PHY layer, multiple adjacent subcarriers are bonded together for data transmission in order to reduce computational complexity. The rationale behind this operation is that the channels of adjacent subcarriers are typically similar. Hence, the bonding strategy does not cause much performance degradation but reduces the complexity significantly. Fig. 4.7 illustrates an example of bonding over five adjacent subcarriers. For the bonded subcarriers, we use the same precoder for power allocation and beam steering on the AP side and the same detection filter for signal recovery on the user side.

AP-Side Precoding. After computing the precoders for each user, we assemble a downlink NOMA transmission frame as shown in Fig. 4.7. The first part of the frame is fixed. The second

part of the frame is computed based on the precoders. Specifically, for the jth LTF in this part, its frequency-domain data is generated by $\mathbf{x}(l,k) = \mathbf{v}_j(k)\bar{s}_j(l,k)$, where $\bar{s}_j(l,k)$ is a pre-defined reference signal. The third part is superposition of precoded data in the frequency domain. It is generated by $\mathbf{x}(l,k) = \sum_{j=1}^{N} \mathbf{v}_j(k)s_j(l,k)$. The generated signal vector $\mathbf{x}(l,k)$ is converted into time domain using IFFT operation. The resulting time-domain signal vector will be sent to RF chains for transmission.

User-Side SIC-based Signal Detection. Since each frame has the IEEE 802.11 legacy preamble, the users can perform frame detection, time synchronization, and frequency offset correction in the same way as conventional Wi-Fi devices do. Afterward, each user performs SIC to decode its desired signal. For ease of exposition, we denote l as the index of OFDM symbol in a frame and denote k as the index of subcarrier in the OFDM modulation. Then, the received signal at STA i can be written as:

$$y_i(l,k) = \sum_{j=1}^{N} \mathbf{h}_i(k)\mathbf{v}_j(k)s_j(l,k) + n_i(l,k). \tag{4.23}$$

To decode the desired signal at STA i, one approach is using ZF SIC (ZF-SIC). This approach decodes and subtracts the strongest signal sequentially until its desired signal is obtained. When decoding the strongest signal, it simply treats other (non-strongest) signals as interference. When decoding s_j, it first estimates the compound channel by $\hat{h}_j(k) = \bar{y}_i(l,k)/\bar{s}_j(l,k)$, where $\bar{y}_i(l,k)$ and $\bar{s}_j(l,k)$ are the received and transmitted reference signals, respectively. Then, it uses the estimated channel to decode the strongest signal by letting $\hat{s}_j(l,k) = y_i(l,k)/\hat{h}_j(k)$, where $\hat{s}_j(l,k)$ is the estimated version of the strongest signal $s_j(l,k)$. Although ZF-SIC is amenable to implementation, its performance is highly suboptimal. This is because it does not take into account the effect of noise and interference (non-strongest signals) in the course of its signal detection. To improve its performance, we may consider MMSE SIC (MMSE-SIC), which takes into

account noise and interference. In contrast to ZF-SIC, MMSE-SIC estimates the strongest signal as follows: $\hat{s}_j(l,k) = \hat{h}_j(k)^*[\hat{h}_j(k)\hat{h}_j(k)^* + \frac{1}{\rho}]^{-1} y_i(l,k)$, where ρ is the SINR. MMSE-SIC requires the knowledge of SINR, making it difficult to implement in practice.

To circumvent the above problems, we propose a new SIC scheme, which uses the reference signals to construct a detection filter directly. Specifically, at STA i, we decode signals $\{s_1(l,k), s_2(l,k), \cdots, s_i(l,k)\}$ in sequence. When decoding the strongest signal $s_j(l,k)$, we construct the detection filter as follows:

$$g_j(k) = \frac{\sum_{(l,k)\in\mathcal{R}_j(k)} y_i(l,k)s_j(l,k)^*}{\sum_{(l,k)\in\mathcal{R}_j(k)} y_i(l,k)y_i(l,k)^*}, \quad 1 \le j \le i, \tag{4.24}$$

where $(\cdot)^*$ is the conjugate operator, and $\mathcal{R}_j(k)$ is the set of reference signals in the jth LTF on subcarrier k. Fig. 4.7 illustrates an example of $\mathcal{R}_j(k)$ when $j = 2$. It can been seen that we use not only the reference signals on subcarrier k but also reference signals on its two neighboring subcarriers to construct detection filter $g_j(k)$. The rationale behind this design is that the summation over multiple subcarriers can reduce the effect of noise and interference (non-strongest signals). After calculating the detection filter, we estimate signal $s_j(l,k)$ in the data part of the frame as follows:

$$\hat{s}_j(l,k) = g_j(k)^* y_i(l,k), \quad 1 \le j \le i, \tag{4.25}$$

where $\hat{s}_j(l,k)$ is the estimated version of $s_j(l,k)$.

Based on (4.24) and (7.12), we present the proposed SIC algorithm in Alg. 4.4. Apparently, this algorithm does not require the estimated SINR. But, it can partially reduce the influence of noise and interference. This is important in SIC detection because all of non-strongest signals are considered as interference. Meanwhile, this SIC algorithm has a low complexity, and it is amenable to practical implementation. For its performance, we will show via experimental results

Algorithm 4.4 The proposed SIC at STA i.

Inputs: Received signal $y_i(l,k)$, reference signals in the frame;
Outputs: Estimated signals in the data part of the frame, i.e., $\hat{s}_i(l,k)$ for $(l,k) \in \mathcal{D}$;
1: **for** $(j = 1; j \leq i; j$++**)** **do**
2: Compute decoding filter $g_j(k)$ using (4.24);
3: Estimate current signal $\hat{s}_j(l,k)$ using (7.12);
4: $\tilde{s}_j(l,k) \leftarrow$ QAM-based demodulation of $\hat{s}_j(l,k)$;
5: $y_i(l,k) \leftarrow y_i(l,k) - \tilde{s}_j(l,k)/g_j(k)^*$;

that it considerably outperforms ZF-SIC.

4.6 Performance Evaluation

In this section, we conduct experiments to evaluate the performance of the proposed NOMA scheme in real-world wireless environments.

4.6.1 Prototyping and Experimental Setup

Experimental Testbed. We have prototyped an AP and three users. The AP has been implemented using two USRP N210 devices and one laptop. Each user has been implemented using an USRP N210 device and one laptop. The USRP devices are used for radio signal transmission and reception, and the laptop is used for baseband signal processing. All the baseband signal processing is carried out by the laptop using Python and C++ in GNU Radio software package. The prototyped AP supports up to two antennas for signal transmission and reception, while users support one antenna.

On the AP, the relative calibration method in [150] was implemented to preserve the uplink/downlink channel reciprocity. This relative calibration method is a standalone calibration procedure that can be done by the AP without requiring the involvement of the users.

Prototyping NOMA. We have implemented the NOMA protocol in Fig. 4.6 on the testbed.

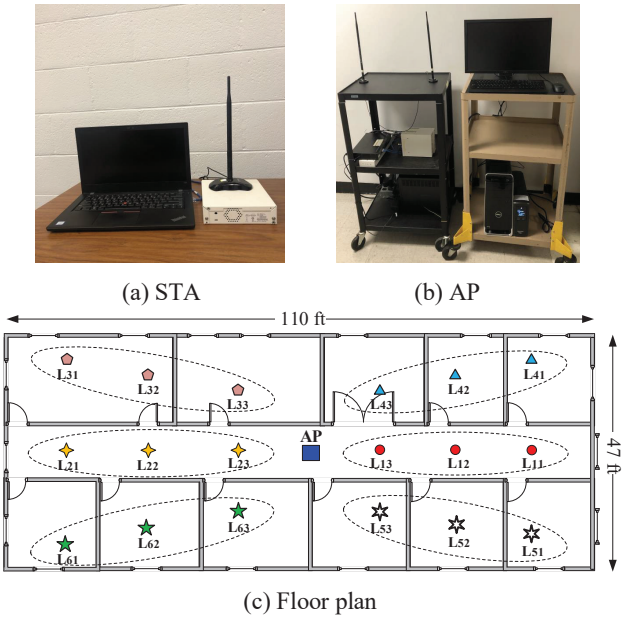


Figure 4.8: Our NOMA testbed and floor plan.

As shown in Fig. 4.6, the protocol first performs uplink channel sounding to obtain the uplink channels. Based on the channel knowledge, Alg. 4.1 is used to compute the precoders ($\mathbf{v}_i$, $i \in \mathcal{N}$) using a convex optimization solver such as CVX and CVXOPT [13] . In OPT-NOMA, we set $w_1 = 3$ for weak user, $w_2 = 2$ for middle user, and $w_3 = 1$ for strong user. These weights are just an example and other weights would also work. After computing the precoders, the AP sends a superimposition of the signals toward users using the frame structure depicted in Fig. 4.7. The users perform SIC to decode their desired signals. In our implementation, we use Schmidl-Cox algorithm for the timing and frequency synchronization at the receivers in both uplink and downlink transmissions. We use least-squares channel estimation at the AP in the uplink channel sounding. For the downlink transmissions, we use the precoded reference signals in the frame (see Fig. 4.7) to construct the channel equalization coefficient $g_j(k)$ and then use this coefficient for signal detection, as detailed in (4.24) and (7.12).

During this protocol, IEEE 802.11 legacy frame parameters are used for both uplink and downlink transmissions. That is, each OFDM symbol has 64 subcarriers in total; 48 subcarriers are used

for data transmission; 4 subcarriers are used for pilot; and 12 subcarriers are null. The 52 valid subcarriers are bonded into two groups for the precoder design in NOMA. The length of cyclic prefix is 16. Due to the hardware limitation, we set the sampling rate to 5 Msps (to avoid the unflat circuit response from the CIC) and set the short interframe space (SIFS) to 2 seconds. Given the 5 MSps sampling rate, the time duration of each OFDM symbol is 16 μs. The data part of each frame consists of 20 OFDM symbols.

Experimental Setup. In our tests, the maximum transmit power for each node (AP or user) is set to 17 dBm. Fig. 4.8 shows the floor plan of our test scenarios and the prototyped AP and STA. Regarding the floor plan, the AP is placed at a fixed location marked "AP". The three users are placed at one of the six different locations. Specifically, STA 1 is placed L_{k1}, STA 2 is placed L_{k2}, and STA 3 is placed L_{k3}, for $k = 1, 2, \cdots, 6$. It is noteworthy that the linear deployment of the three users in a group is for ease of explanation. In a rich scattering environment like an office building, such a deployment does not impose a significant correlation among users' channels. Moreover, it is worth pointing out that our experimentation does not include the user grouping algorithm.

4.6.2 Performance Metrics and Benchmark

Performance Benchmark. We use OMA as the performance benchmark to evaluate the throughput gain of NOMA. In OMA, the round-robin scheduler is used at the AP. Specifically, the AP serves only one user in one time slot, and different users are scheduled in different time slots. When the AP has multiple antennas, the best antenna is selected for spatial diversity.

Performance Metrics. We evaluate the performance of the proposed NOMA scheme using the two metrics: EVM and data rate. (i) EVM is a metric widely used in WLANs. At STA j, its EVM is defined as $\mathrm{EVM} = 10\log_{10}\left(\mathbb{E}_{l,k}\left[\left|\hat{s}_j(l,k) - s_j(l,k)\right|^2\right] / \mathbb{E}_{l,k}\left[\left|s_j(l,k)\right|^2\right]\right)$. (ii) The

data rate is extrapolated based on the measured EVM at each user using the MCS specified in IEEE 802.11 [67]. Specifically, the data rate at STA j is calculated by

$$\text{NOMA:} \quad r_j = \frac{48}{80} \cdot b \cdot \eta(\text{EVM}), \tag{4.26a}$$

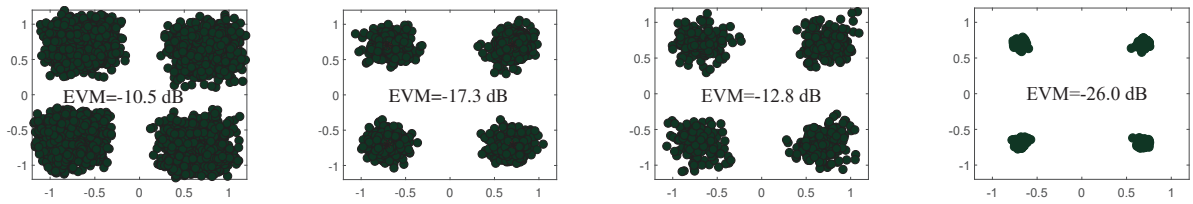
$$\text{OMA:} \quad r_j = \frac{1}{N} \cdot \frac{48}{80} \cdot b \cdot \eta(\text{EVM}), \tag{4.26b}$$

where N is the number of users served by the AP, 48 is the number of payload subcarriers, 80 is the number of samples in an OFDM symbol, b is the bandwidth (5 MHz), EVM is measured at the STA j when NOMA or OMA is used, and $\eta(\text{EVM})$ is the average number of bits carried by one subcarrier in an OFDM symbol and its value is given in Table 4.1.

4.6.3 Experimental Results of (1×2)-NOMA

We first consider the case where the AP has one antenna and it serves two users (one weak user and one strong user). The weak user is placed at L_{k1} and the strong user is placed at L_{k3}, $k = 1, 2, \cdots, 6$.

Case Study. We use location 4 ($k = 4$) as an example to examine NOMA. Fig. 4.9 presents the constellation of the decoded signals at the two users when NOMA and OMA are used, respectively. For the weak user, Fig. 4.9(a) and Fig. 4.9(c) show that NOMA has a small (2.3 dB) EVM degradation compared to OMA. Using (4.26), the data rate at the weak user is extrapolated to 3.0 Mbps in NOMA and 1.5 Mbps in OMA. This indicates that NOMA has a significant throughput gain for the weak user. For the strong user, Fig. 4.9(b) show its decoded signals after SIC in NOMA, respectively. We can see that the strong user can achieve -17.3 dB EVM after SIC. Compared Fig. 4.9(b) with Fig. 4.9(d), we can see that the strong user has a consider-



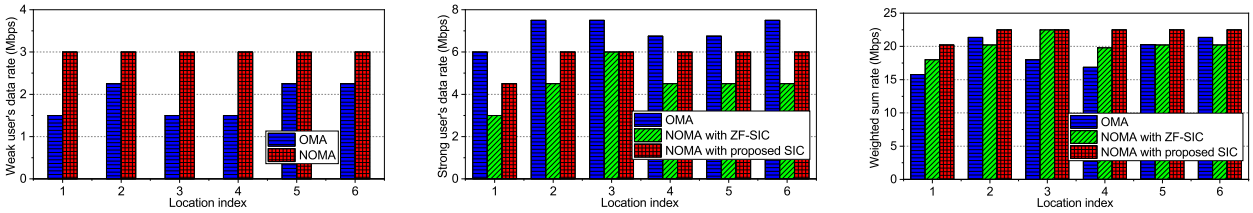
(a) NOMA: signal at weak user.

(b) NOMA: signal at strong user.

(c) OMA: signal at weak user.

(d) OMA: signal at strong user.

Figure 4.9: Constellations of NOMA and OMA in downlink.



(a) Weak user's data rate

(b) Strong user's data rate

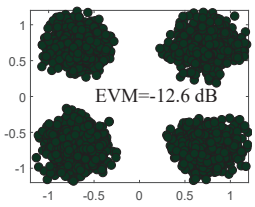
(c) AP's weighted sum rate

Figure 4.10: Performance comparison of NOMA and OMA in downlink transmission of a WLAN where a single-antenna AP serves two single-antenna users.

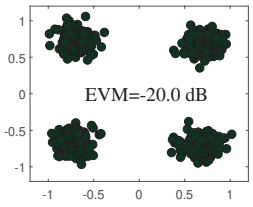
able EVM degradation (about 9.4 dB) when NOMA is used. Using (4.26), the data rate achieved by the strong user is extrapolated to 6.0 Mbps in NOMA and 6.7 Mbps in OMA. Compared to OMA, our NOMA scheme slightly decreases the strong user's data rate. The reasons are two-fold. First, NOMA serves two users while OMA serves one user. Moreover, a higher weight is assigned for the weak user to maintain the fairness in NOMA transmission when we conduct the optimization (OPT-NOMA). Second, SIC in NOMA is not perfect due to the limited ADC resolution, circuit nonlinearity and distortion. The imperfection of SIC degrades the performance of the strong user.

From the AP's perspective, the weighted sum rate of the two users is 22.5 Mbps in our NOMA scheme and 16.9 Mbps in OMA. This means that our NOMA scheme has about 33.5% improvement over OMA in terms of weighted sum rate.

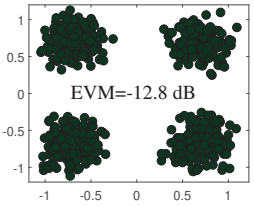
Results from All Locations. Fig. 4.10 presents the extrapolated data rate at each user when NOMA and OMA are used, respectively. Two SIC techniques are implemented for the strong



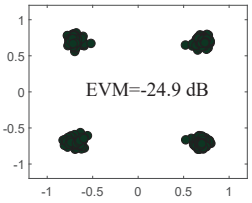
(a) NOMA: signal at weak user.



(b) NOMA: signal at strong user.



(c) OMA: signal at weak user.



(d) OMA: signal at strong user.

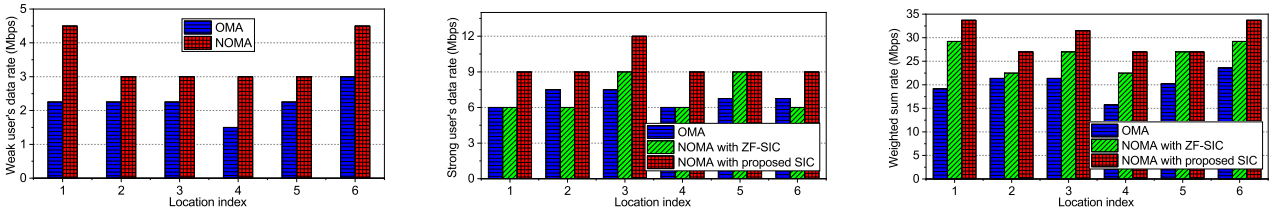
Figure 4.11: Constellations of NOMA and OMA in the downlink of WLAN where a two-antenna AP serves two single-antenna users.

user: ZF-SIC and our proposed SIC. Based on the results, we have the following observations: (i) For the weak user, our NOMA scheme has a 60.0% data rate gain compared to OMA, on average over all the locations that we tested. (ii) For the strong user, NOMA yields a 17.9% data rate degradation on average compared to OMA. This degradation can be attributed to the inter-user interference and the imperfections of SIC as explained above. (iii) For the AP, the proposed NOMA scheme outperforms OMA by 18.0% in terms of weighted sum rate. (iv) The proposed SIC scheme outperforms ZF-SIC by 27.8% for the stronger user's data rate. This throughput gain is from the summation operation in (4.24). Mathematically, this summation operation is equivalent to a low pass filter, which reduces the effects of noise and interference (weak signals) in each iteration of SIC.

4.6.4 Experimental Results of (2×2)-NOMA

We now consider the case where the AP has two antennas and it serves two users. The weak user is placed at L_{k1} and the strong user is placed at L_{k3}, $k = 1, 2, \cdots, 6$.

Case Study. Again, we use location 4 ($k = 4$) as an example to examine the proposed NOMA scheme. Fig. 4.11 presents the constellation of the decoded signals at the two users when NOMA and OMA are used, respectively. We have the following observations from the experimental data. (i) For the weak user, it has similar EVM in NOMA and OMA. Its extrapolated data rate is



(a) Weak user's data rate (b) Strong user's data rate (c) AP's weighted sum rate

Figure 4.12: Performance comparison of NOMA and OMA in downlink transmission of a WLAN where a two-antenna AP serves two single-antenna users.

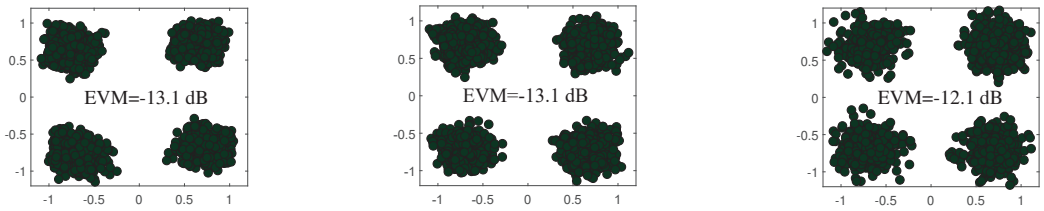
3.0 Mbps in NOMA and 1.5 Mbps in OMA. This shows that NOMA has a significant throughput gain for the weak user. (ii) For the strong user, it achieves -20.0 dB EVM in NOMA and -24.9 dB EVM in OMA. Correspondingly, the extrapolated data rate for this user is 9.0 Mbps in NOMA and 6.0 Mbps in OMA. This shows that NOMA has a considerable throughput gain (50.0%) for the strong user as well. (iii) For the AP, the weighted sum rate is 27.0 Mbps in NOMA and 15.7 Mbps in OMA. This shows that our NOMA scheme has a 72.0% gain over OMA.

Results from All Locations. Fig. 4.12 presents the extrapolated data rate at each user when NOMA and OMA are used, respectively. The experimental results from the six locations corroborate our observations in the case study. On average, NOMA improves the data rate by 55.5% for the weak user, 40.7% for the strong user, 49.8% for the AP's weighted sum rate. Moreover, our proposed SIC outperforms ZF-SIC, yielding 35.7% data rate gain for the strong user.

4.6.5 Experimental Results of (2×3)-NOMA

Finally, we consider the case where the AP has two antennas and it serves three users. The weak user is placed at L_{k1}, the middle user is placed at L_{k2}, and the strong user is placed at L_{k3}, $k = 1, 2, \cdots, 6$.

Case Study. Similar to the previous case studies, we place the users at location 4. Fig. 4.13 presents the constellation of the decoded signals at the users when NOMA is used. When OMA

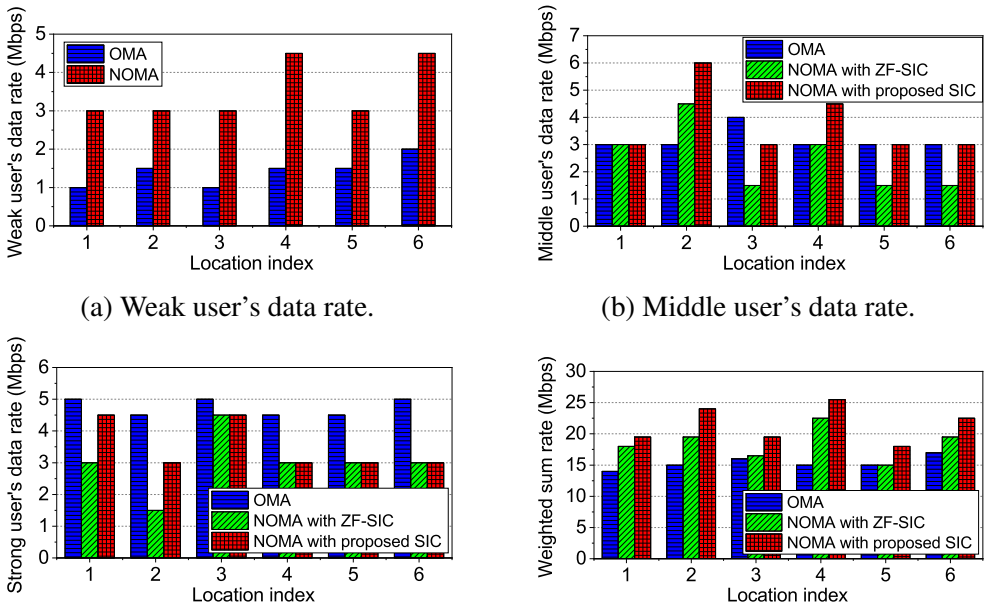


(a) 1st-decoding at weak user. (b) 2nd-decoding at middle user. (c) 3rd-decoding at stronger user.

Figure 4.13: Constellation of the decoded signals in downlink NOMA transmission when the two-antenna AP serves three single-antenna users.

is used, the three stations achieve -14.5 dB, -21.4 dB, and -26.8 dB EVM, respectively. Based on the experimental results, we have the following observations: (i) For the weak user, NOMA has a small EVM degradation (1.4 dB) compared to OMA. Its extrapolated data rate is 4.5 Mbps in NOMA and 1.5 Mbps in OMA. (ii) For the middle user, NOMA has a considerable EVM degradation (8.3 dB) compared to OMA. Its extrapolated data rate is 4.5 Mbps in NOMA and 3.0 Mbps in OMA. (iii) For the strong user, NOMA has a significant EVM degradation (13.7 dB). Its extrapolated data rate is 3.0 Mbps in NOMA and 4.5 Mbps in OMA. (iv) For the AP, the weighted sum rate is 25.5 Mbps in NOMA and 15.0 Mbps in OMA. This means that NOMA has a 70.0% gain over OMA in terms of weighted sum rate.

Results from All Locations. Fig. 4.14 presents the extrapolated data rate at each user when NOMA and OMA are used. From the experimental results, we can see that NOMA significantly increases the weak user's data rate, slightly increases the middle user's data rate, and considerably decreases the strong user's data rate. On average over the six locations, NOMA increases the data rate by 147.1% for the weak user and by 18.4% for the middle user. However, it decreases the data rate of the strong user by 26.3%. For the AP, NOMA achieves a weighted sum rate of 21.5 Mbps and OMA achieves 15.3 Mbps, indicating a 40.5% improvement. Meanwhile, it is evident that our proposed SIC considerably outperforms ZF-SIC. It improves the strong user's data rate by 16.7% and the middle user's data rate by 50.0%.



(a) Weak user's data rate.

(b) Middle user's data rate.

(c) Strong user's data rate.

(d) AP's weighted sum rate.

Figure 4.14: Performance comparison of NOMA and OMA in downlink transmission when the two-antenna AP serves three single-antenna users.

4.6.6 Summary of Observations

Based on our experimental results, we have the following observations on NOMA: (i) NOMA can significantly increase the weak user's data rate when compared to OMA. This phenomenon has been observed in all the cases that we tested in our experiments. (ii) As expected, the use of NOMA will lead to a degradation for the strong user's data rate. But in overall, NOMA can greatly improve the weighted sum rate for the AP. (iii) Our proposed SIC method works in practice and it offers considerably better performance than ZF-SIC.

4.7 Chapter Summary

In this chapter, we proposed a NOMA scheme for WLANs and evaluated its performance in real-world wireless environments. Our NOMA scheme has three key components: precoder design,

user grouping, and a new SIC method. We formulated the precoder design problem as an optimization problem and developed a minorization-majorization algorithm to pursue an efficient solution to it. Moreover, a robust SIC method has been proposed to decode the desired signal in the presence of strong interference. Our SIC method does not require channel estimation and is amenable to practical implementation. We have implemented the proposed NOMA scheme on a GNURadio-USRP2 testbed. Experimental results show that, compared with OMA, the proposed NOMA scheme can significantly improve the weak user's data rate and considerably improve the AP's weighted sum rate.

Chapter 5

Learning-Based Channel Feedback for MU-MIMO in WLANs

5.1 Introduction

The proliferation of wireless devices, combined with the growth of Internet-based wireless applications such as online streaming and video chatting, has led to continuously increasing demands for wireless services in indoor environments such as smart homes, university campuses, football stadiums, and airports. As one of the largest wireless networks in real world, WLANs carry the most wireless data traffic (even more than cellular networks) and play a pivotal role in our society. To meet the increasing demands for data services in WLANs, MU-MIMO is a key technology. It allows an AP to serve multiple users simultaneously and therefore can significantly improve the spectral efficiency. Given its potential, MU-MIMO has been specified in the IEEE 802.11 standards [67,70] and widely been deployed on commercial Wi-Fi devices, e.g., Wi-Fi routers, laptops, and phones.

In real-world WLANs, the downlink typically has higher demands for data services compared to the uplink. To support downlink MU-MIMO communications in WLANs, an AP needs to access short-term CSI for the construction of beamforming filters. The filters will then be used to project the precoded signals onto the AP's multiple antennas so that each user can decode its

data packets. Thus, CSI at the AP is essential to enabling downlink MU-MIMO transmissions. There are two channel acquisition methods for an AP to obtain CSI: i) *implicit channel acquisition*, and ii) *explicit channel acquisition*. The implicit method is based on channel reciprocity. The AP infers the downlink CSI through the estimation of uplink CSI and periodic channel calibrations [124]. This method, however, requires an extra RF chain on hardware or a sophisticated algorithm for channel calibration, and may not be suited for implementation on low-cost Wi-Fi devices [74, 75, 206].

The explicit method is based on channel feedback over uplink over-the-air channel. Each user first estimates the downlink CSI and then reports the estimated CSI to the AP. Given its amenability to implementation, this method has been adopted by the IEEE 802.11 standards [67, 70] and been implemented on commercial Wi-Fi systems. However, due to its reliance on over-the-air CSI feedback, it suffers from large airtime overhead. The large overhead of this method can be attributed to the large number of subcarriers in WLANs' OFDM modulation, each of which has a channel matrix to be reported. Existing 802.11 protocols may group subcarriers for CSI feedback to reduce the overhead. Apparently, such a naive scheme will lead to an inferior beamforming performance and drastically compromises the throughput gain of MU-MIMO. While there are many results of MU-MIMO in the literature, the CSI compression for 802.11 MU-MIMO protocols is highly overlooked and its progress remains limited.

In this chapter, we study explicit channel acquisition in 802.11 MU-MIMO protocols with the objective of minimizing CSI feedback airtime overhead while preserving CSI feedback accuracy. Toward this objective, we propose a learning-based channel feedback framework (called LB-SciFi[1]) for 802.11 protocols to reduce their airtime overhead by taking advantage of recent

[1] LB-SciFi stands for Learning-Based compression for Ψ (Sci) and Φ (Fi), which are the CSI for feedback in 802.11 MU-MIMO protocols [67, 70].

(a) Online training. (b) Real-time exploitation.

Figure 5.1: An overview of DNN-AEs for channel feedback compression in 802.11 MU-MIMO protocols.

advances in Deep Neural Network Auto-Encoder (DNN-AE). Fig. 5.1 shows the basic idea of LB-SciFi, which is composed of two phases: *online training* and *real-time exploitation*. In the *training* phase, LB-SciFi trains DNN-AEs at the AP by leveraging side information from existing 802.11 MU-MIMO protocols, and thus require no extra effort from user devices. In the *exploitation* phase, LB-SciFi uses the trained DNN-AEs to compress CSI for efficient feedback. Given the redundancy of CSI and the effectiveness of DNN-AEs, LB-SciFi can reduce the airtime overhead significantly without sacrificing CSI feedback accuracy.

The main challenge in the design of LB-SciFi is the online training of DNN-AEs, which should be capable of capturing the kernel space of all possible channels in a given wireless environment through the learning of collected CSI at the AP. To address this challenge, we design an efficient training scheme for the DNN-AEs, which jointly optimize the structure of DNN-AEs, the collection of training data, and the preprocessing of collected data by leveraging the existing feedback data in existing 802.11 MU-MIMO protocols. Specifically, the proposed training scheme meticulously chooses the ψ and ϕ angles from Givens Rotation (GR) as the DNN-AEs input based on a defined Power spectral Entropy (PSE). Moreover, several important engineering problems have

been addressed to make DNN-AEs work in real-world wireless environments.

This work advances the state-of-the-art in the following respects.

- We propose to employ DNN-AEs for CSI compression in 802.11 MU-MIMO protocols, and have designed an online training scheme for DNN-AEs while imposing no computation burden on user devices.

- Based on the DNN-AEs, we have designed a learning-based channel feedback framework (LB-SciFi) for downlink MU-MIMO. This framework can dramatically reduce the CSI feedback airtime overhead for 802.11 MU-MIMO protocols without sacrificing CSI feedback accuracy.

- We have built a prototype of LB-SciFi and evaluated its performance in real-world indoor environments. Our experimental results show that LB-SciFi reduces the CSI feedback airtime overhead by 73% and improves the throughput of MU-MIMO by 69% on average.

5.2 Related Works

We focus our literature review on research efforts studying low-overhead channel acquisition methods for MU-MIMO transmissions in WLANs and cellular networks.

Channel Acquisition in WLANs. As the core technology of existing WLANs, MU-MIMO markedly improves users experience with high throughout and low latency. However, airtime overhead from channel acquisition is a real barrier toward fully exploiting the potential of MU-MIMO. Given the severity of this issue, research efforts have been devoted to studying the effect of channel acquisition parameters on network throughput or completely altering the channel acquisition paradigm to enhance network throughput [17, 33, 53, 87, 110, 113, 117, 128, 134, 139, 197, 200].

Pioneering work [128, 134, 139, 197] studied the underlying relationship between network throughput and channel acquisition parameters. The outcome was not surprising; full exploitation of MU-MIMO requires a timely CSI through a frequent channel acquisition. The large airtime overhead, however, drastically compromises the throughput gain of MU-MIMO. [17, 33, 87, 113, 117] aimed at lowering the frequency of channel acquisitions to reduce channel feedback overhead for MU-MIMO protocols. However, the airtime overhead was still too large. [53,110,200] revisited existing channel acquisition paradigm and explored new methods for efficient channel acquisition.

Thus far, there is no efficient method for CSI compression to reduce feedback overhead. Our work fills this gap by leveraging recent advances in artificial neural networks to compress CSI. The resultant CSI feedback will entail much less overhead compared to existing 802.11 protocols.

Channel Acquisition in Cellular Networks. Compared to WLANs, the need for low-overhead channel acquisition methods in cellular networks is appreciated earlier as the emergence of massive-MIMO revealed the drawbacks of traditional techniques. Toward this objective, the underlying correlation of CSI reports has been used for compression by removing the redundant correlated information [54, 97, 111, 172, 179, 193]. In particular, temporal correlation [97, 111, 172], spectral correlation [54, 179, 193], and spatial correlation [97, 101] have been explored to minimize the representation of CSI. Channel reciprocity [109, 143, 194] and outdated CSI [73] have also been studied to enhance the efficiency of channel acquisition.

Our work is orthogonal to these research efforts in the following two aspects: i) Our work focuses on indoor WLANs, which differ from cellular networks in terms of CSI format, network architecture, data collection, data processing, and system implementation. ii) While the above efforts focused on theoretical exploration, our work focuses on practical design based on real-world 802.11 protocols.

5.3 Problem Description

In this section, we first offer a primer of existing 802.11 MU-MIMO protocols and underscore their airtime overhead issue. Then, we will state our design objective and challenges.

5.3.1 Existing 802.11 MU-MIMO Protocols

Consider a WLAN as shown in Fig. 5.1(a), where a multi-antenna AP is serving a set of user devices (a.k.a. stations or STAs for brevity). The AP is equipped with N_{ap} antennas, and an STA is equipped with N_{sta} antennas. Due to the physical size and power limits, an STA typically has less antennas than an AP, i.e., $N_{sta} < N_{ap}$. In such a WLAN, MU-MIMO is widely used to exploit the spatial DoF of asymmetric antenna configuration by enabling the AP to serve multiple STAs simultaneously. The application of MU-MIMO not only improves spectral efficiency and user scheduling flexibility but it also reduces packet delay at the MAC layer and enhances fairness in resource allocation.

To enable MU-MIMO transmissions in real-world WLANs, protocols with explicit channel acquisition have been specified in the IEEE 802.11 standards [67, 70]. Fig. 5.2 shows an existing 802.11 MU-MIMO protocol, which is composed of the following four phases:

- *MU-MIMO Announcement:* The AP selects a subset of STAs for the downlink MU-MIMO transmission based on some pre-defined criteria.[2] After user selection, the AP broadcasts a NDPA to inform the STAs of MU-MIMO transmission, followed by an NDP for those STAs to estimate downlink CSI.

- *Channel Feedback:* After estimating CSI, the selected STAs feed back their CSI to the AP

[2]Note that user selection is not in the scope of our work, and there are many prior results on user selection for MU-MIMO.

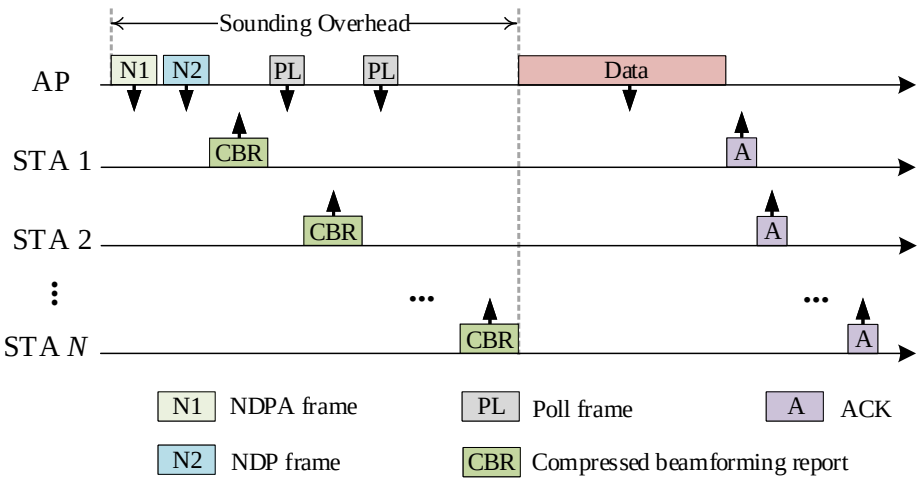


Figure 5.2: An MU-MIMO protocol in IEEE 802.11ac [67].

sequentially following the poll frames from the AP, as shown in Fig. 5.2. The CSI feedback procedure will be detailed shortly.

- *Data Transmission:* Upon obtaining CSI from all STAs, the AP uses CSI to construct beamforming filters and performs downlink data transmission.

- *Acknowledgment:* After decoding the packets, all STAs sends an ACK/NACK to the AP to indicate the success/failure of their packet detection.

In the channel feedback phase, if an STA sends raw CSI to the AP, it entails a huge amount of airtime overhead and thus negates the throughput gain of MU-MIMO. To reduce the airtime overhead, 802.11 protocols have employed angle-based CSI feedback instead of raw CSI feedback in the spatial domain and specified subcarrier grouping in the spectral domain. We detail them below.

Angle Feedback in Spatial Domain. Referring to the protocol in Fig. 5.2, once an STA has received the NDP from the AP, it estimates the downlink CSI, i.e., $\mathbf{H}(k) \in \mathbb{C}^{N_{sta} \times N_{ap}}$, $1 \leq k \leq N_{sc}$, where N_{sc} is the number of valid subcarriers. Instead of reporting the complex entries of $\mathbf{H}(k)$, the STA reports two sets of angles ($\mathbf{\Psi}$ and $\mathbf{\Phi}$) to the AP to reduce the feedback overhead. A

Algorithm 5.1 A high-level description of computing $\mathbf{\Psi}$ and $\mathbf{\Phi}$ at an STA specified in the IEEE 802.11ac/ax [67, 70].

Inputs: Estimated channel at an STA, i.e., $\mathbf{H}(k) \in \mathbb{C}^{N_{sta} \times N_{ap}}$, $1 \leq k \leq N_{sc}$
Outputs: Computed angles, i.e., $\mathbf{\Psi}$ and $\mathbf{\Phi}$
Set $\mathbf{\Psi} = \{\ \}$ and $\mathbf{\Phi} = \{\ \}$
for ($k = 1$; $k \leq N_{sc}$; k++) **do**
    $[\mathbf{U},\ \mathbf{\Sigma},\ \mathbf{V}] = svd\left(\mathbf{H}(k)\right)$
    $\mathbf{V'} = \mathbf{V}\left(:,\ 1{:}N_{sta}\right)$
    for ($l = 1$; $l \leq N_{sta}$; l++) **do**
        $\psi_k := phase\_extraction(\mathbf{V'}\left(:, l\right))$
        $\phi_k := givens\_rotations(\mathbf{V'}\left(:, l\right))$
        $\mathbf{\Psi} := \left\{\mathbf{\Psi}\ \ \psi_k\right\}$ and $\mathbf{\Phi} := \left\{\mathbf{\Phi}\ \ \phi_k\right\}$
Quantizing every angle in $\mathbf{\Psi}$ using p bits, $p \in \{5, 7\}$
Quantizing every angle in $\mathbf{\Phi}$ using q bits, $q = p + 2$.

high-level description of computing $\mathbf{\Psi}$ and $\mathbf{\Phi}$ is given in Alg. 5.1. This conversion is also known as Givens Rotations. Details of computing the angles can be found in [156]. With these two sets of angles, the AP can reconstruct the essential spatial information of $\mathbf{H}(k)$, which suffices for beamforming operations at the AP.

In this method, the number of generated angles in $\mathbf{\Phi}$ is $N_\phi = (N_{ap}N_{sta} - N_{sta}^2/2 - N_{sta}/2)N_{sc}$, so is the number of angles in $\mathbf{\Psi}$. These angles need to be reported to the AP via the uplink over-the-air channels. In 802.11 standards [67], two types of quantization are specified for CSI feedback:

- *Type 0:* 5 bits for angles in $\mathbf{\Psi}$ and 7 bits for angles in $\mathbf{\Phi}$,
- *Type 1:* 7 bits for angles in $\mathbf{\Psi}$ and 9 bits for angles in $\mathbf{\Phi}$.

Subcarrier Grouping in Spectral Domain. In a typical environment of WLANs, adjacent subcarriers experience highly correlated channel responses from the medium. Therefore, instead of reporting CSI for every individual subcarrier, an STA may group multiple neighboring subcarriers together for CSI feedback. Per IEEE 802.11ac [67], the number of subcarriers in a group, denoted by N_g, can be 1, 2, or 4, depending on network configuration. In IEEE 802.11ax [70], N_g can also

be 8 or 16 due to its small subcarrier spacing.

Large Airtime Overhead. Even with the spatial- and spectral-domain compression, 802.11 MU-MIMO protocols still come with a large amount of airtime overhead, which significantly compromises the throughput gain of MU-MIMO [110, 187]. For example, for an STA with 4 antennas and an AP with 8 antennas, the CSI feedback could be as large as 19.7 kbit for 20 MHz bandwidth and 170.4 kbit for 160 MHz bandwidth. The problem of CSI feedback airtime overhead becomes increasingly acute as the evolution of WLANs is accommodating more subcarriers in a certain frequency band. For example, IEEE 802.11ax employs 256 subcarriers over 20 MHz for packet transmissions, which is four times greater than that of IEEE 802.11ac.

5.3.2 Our Objective and Challenges

Objective. We aim to reduce the CSI feedback airtime overhead by taking advantage of recent advances in DNN-AE, which has been successfully used for data compression and feature extraction in other fields such as image and video compression. Toward this aim, we will compress the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ in the spectral domain by removing their information redundancy caused by channel correlation.

Challenges. While the idea is straightforward, there are challenges in the design of practical DNN-AEs that are amenable to real-world applications. The challenges lie in the following respects: i) The configuration of DNN-AEs should be meticulously selected, including the number of layers in autoencoder, the number of neurons on each layer, and the preprocessing of input data. The configuration of DNN-AEs is of great importance as it dictates the compression rate, the information loss, and the required data amount and computational power for training. ii) The training of DNN-AEs should be online and transparent to user devices. User devices are typically constrained by their computational capability and battery power. It is desirable that the training of DNN-AEs,

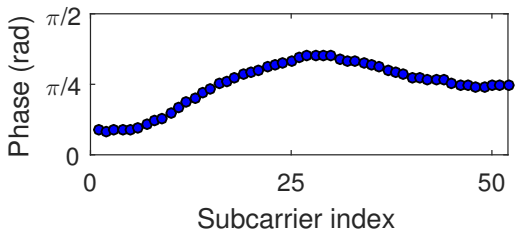
which is computational demanding, will not put a burden on user devices. In what follows, we propose LB-SciFi to address these two challenges.

5.4 LB-SciFi: A Learning-Based Feedback Framework

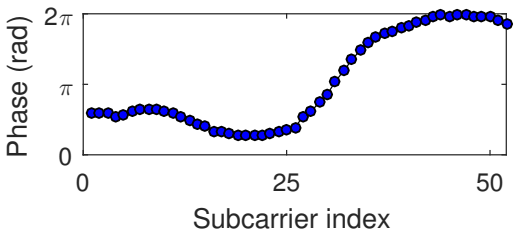
To reduce the CSI feedback airtime overhead, we propose LB-SciFi for CSI compression. The core components of LB-SciFi are two DNN-AEs, which compress CSI at each STA and decompress CSI at the AP. Fig. 5.1 shows the basic idea of LB-SciFi, which is composed of two phases: *online training* and *real-time exploitation*. As shown in Fig. 5.1(a), the online training is done at the AP by taking advantage of the side information ($\mathbf{\Psi}$ and $\mathbf{\Phi}$) from existing 802.11 protocols. Once the training of DNN-AEs is completed, the AP broadcasts the weights of the DNN-AEs to all STAs and enters into the exploitation phase as shown in Fig. 5.1(b). In the exploitation phase, each STA uses DNN-AEs to compress its CSI and reports the compressed CSI to the AP. The AP uses DNN-AEs to decompress the received CSI for the construction of beamforming filters.

5.4.1 DNN-AEs

Autoencoder is a type of artificial neural network used to learn efficient data coding in a self-supervised manner. One of its applications is to learn a representation for a set of data for dimensionality reduction. Autoencoders are effectively used for solving many applied problems, ranging from face recognition to acquiring the semantic meaning of words. In this work, we take advantage of recent advances in DNN-AEs to compress CSI for 802.11 MU-MIMO protocols. We consider a DNN-AE as shown in Fig. 5.1, which is composed of two parts: encoder and decoder. The encoder will be used on each STA to compress its estimated CSI for feedback, and the decoder will be used at the AP to recover CSI for construction of beamforming filters.

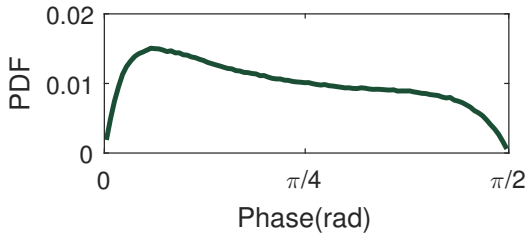


(a) An angle in $\mathbf{\Psi}$ on 52 subcarriers (measured PSE is 0.11).

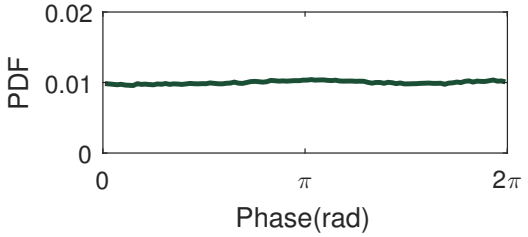


(b) An angle in $\mathbf{\Phi}$ on 52 subcarriers (measured PSE is 0.25).

Figure 5.3: Angle instances in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ as well as their PSE.



(a) PDF of the angles in $\mathbf{\Psi}$.



(b) PDF of the angles in $\mathbf{\Phi}$.

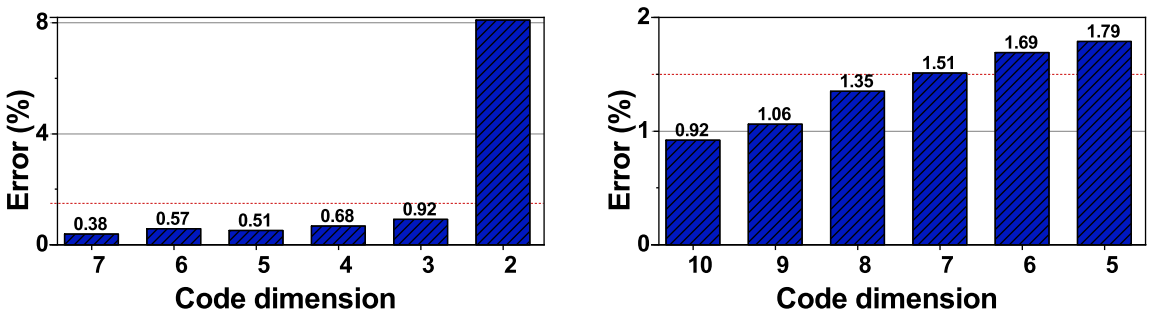
Figure 5.4: Distribution of the measured angles over all subcarriers and at many locations in a real-world office environment.

Compressibility of $\mathbf{\Psi}$ and $\mathbf{\Phi}$. Before delving into the details of DNN-AEs, we introduce a metric to quantify compressibility of angles on an observation basis. The compressibility metric will lay the foundation for our design of DNN-AEs. Consider an angle sequence $\underline{\theta} = [\theta_1, \theta_2, \cdots, \theta_K]$. Denote its FFT output as $\underline{\vartheta} = [\vartheta_1, \vartheta_2, \cdots, \vartheta_K]$. Then, we define PSE of $\underline{\theta}$ as follows:

$$\text{PSE}(\underline{\theta}) = -\frac{1}{\log_2 K} \sum_{k=1}^{K} p(\vartheta_k) \log_2 p(\vartheta_k), \tag{5.1}$$

where $p(\vartheta_k) = \frac{|\vartheta_k|^2}{\sum_{i=1}^{K} |\vartheta_i|^2}$ [65]. Apparently, the PSE of an angle sequence is bounded in $[0\ 1]$. In our case, its value reflects the uncertainty of a random angle or fluctuations of a measured angle over subcarriers. Intuitively, a low value of PSE indicates high compressibility, while a high value of PSE indicates low compressibility.

In WLANs, STAs are semi-stationary and work on a limited bandwidth. In such an envi-



(a) Error for ψ angles.

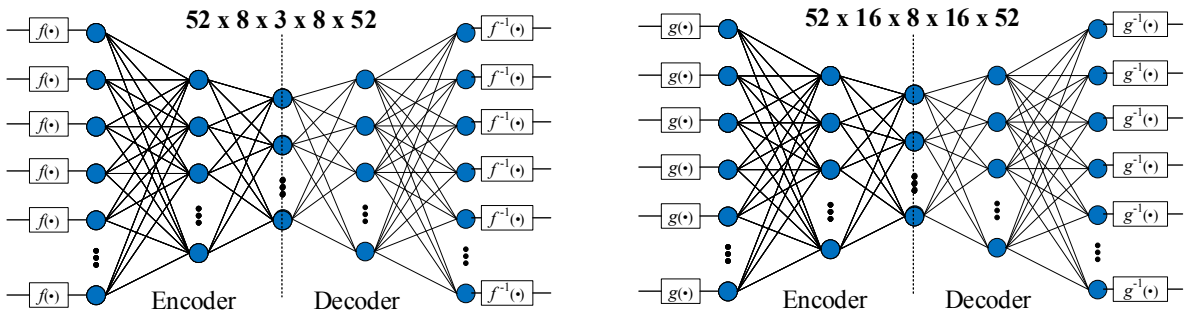
(b) Error for ϕ angles.

Figure 5.5: Compression error for different code dimensions.

ronment, the channels between an AP and STAs are prone to be frequency-flat, and the channel responses on adjacent subcarriers are highly correlated. Fig. 5.3 exhibits an angle in $\mathbf{\Psi}$ and an angle in $\mathbf{\Phi}$ over 52 valid subcarriers in 20 MHz bandwidth at 2.484 GHz as well as their PSE values. It is evident that both PSE values are much less than 1, indicating the compressibility of the angles.

Separate DNN-AEs for $\mathbf{\Psi}$ and $\mathbf{\Phi}$. For an STA, it needs to first compress $\mathbf{\Psi}$ and $\mathbf{\Phi}$, and then report compressed $\mathbf{\Psi}$ and $\mathbf{\Phi}$ to the AP. A natural question to ask is whether an STA should use the same DNN-AE for both $\mathbf{\Psi}$ and $\mathbf{\Phi}$. To explore an answer to this question, we empirically study the compressibility of the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$. Specifically, we collected the CSI angles for $\mathbf{\Psi}$ and $\mathbf{\Phi}$ at the STAs that were widely distributed in a real-world office environment, and plotted the PDF of the collected angles. Fig. 5.4 shows our measured results. We can see that the angles in $\mathbf{\Psi}$ is non-uniformly distributed, while the angles in $\mathbf{\Phi}$ are almost uniformly distributed. Based on collected CSI angles, the measured PSE of $\mathbf{\Psi}$ is 0.09, and the measured PSE of $\mathbf{\Phi}$ is 0.23. The measurement results indicate that the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ have different levels of compressibility. Given that the compression ratio is determined by the DNN-AE structure (the ratio of dimension of the input layer to that of the latent layer), we employ two different DNN-AEs for the compression of $\mathbf{\Psi}$ and $\mathbf{\Phi}$.

DNN-AE Settings. Another question to ask is about the parameter selection of the two DNN-



(a) DNN-AE for $\mathbf{\Psi}$. (b) DNN-AE for $\mathbf{\Phi}$.

Figure 5.6: Illustration of two different DNN-AEs for $\mathbf{\Psi}$ and $\mathbf{\Phi}$.

AEs, including the number of layers, the number of neurons on each layer, quantization bits, and dimension of the latent layer. Unfortunately, there is no systematic approach that we can utilize to determine the optimal values for these parameters. Therefore, we focus only on the dimension of the latent layer (a.k.a. code dimension) as it is the most important parameter for a DNN-AE. Fig. 5.5 presents the compression error of DNN-AEs for different code dimensions. Using 1.5% error as reference, we select the code dimension that offers the best compression rate. As such, our design choices are $52 \times 8 \times 3 \times 8 \times 52$ for $\mathbf{\Psi}$'s DNN-AE and $52 \times 16 \times 8 \times 16 \times 52$ for $\mathbf{\Phi}$'s DNN-AE, as shown in Fig. 5.6.

5.4.2 Online Training: Data Collection

As illustrated in Fig. 5.1, the AP takes advantage of existing 802.11 protocols to train the DNN-AEs. That is, AP and STAs perform downlink MU-MIMO transmissions using the 802.11 protocol as shown in Fig. 5.2. In the meantime, the AP trains the DNN-AEs using reported CSI (uncompressed $\mathbf{\Psi}$ and $\mathbf{\Phi}$) from the STAs. By doing so, the AP can train the DNN-AEs by collecting side information from the existing MU-MIMO protocol, and the training remains transparent to the STAs. In the course of data collection, care should be taken for the following two tasks.

Avoiding Garbage-In/Garbage-Out. To collect a meaningful dataset for training DNN-AEs, the AP needs to block out garbage CSI reports from STAs. In real WLANs, an STA may fail in es-

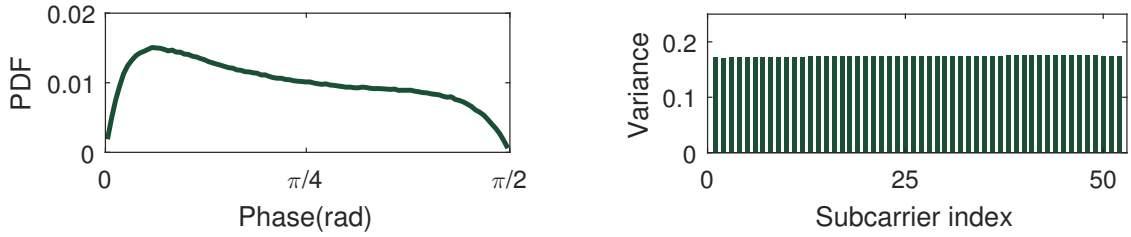
timating accurate CSI due to various sources of errors such as time and frequency synchronization errors. As a garbage report has intrinsically a noise-like behavior, several dominant components exist in its spectral representation. Therefore, the PSE of such a report is high likely to be overly high. The AP leverages PSE metric in (5.1), and blocks out the sequences with abnormal PSE. The abnormality is detected by adjusting appropriate thresholds. In our experiment, we assumed that an abnormal angle in $\mathbf{\Psi}$ has PSE ≥ 0.25 and that an abnormal angle in $\mathbf{\Phi}$ has PSE ≥ 0.5.

Avoiding Overrepresentation. Another important task of the AP is to prepare a balanced data set. In a typical WLAN, a *static* STA like smart TV remains at a fixed location without quitting the WLAN, while a *mobile* STA wanders through coverage range and may quit the WLAN for a while. A static STA may temporally experience correlated large-scale fading, making its historical CSI reports highly correlated. In light of this, the CSI reports from static STAs might be over-represented, making the DNN-AEs biased in favor of themselves. To avoid overrepresentation, the AP divides the PSE range into 100 uniform bins. If the AP receives 20 consecutive CSI samples of the same PSE value from the same STA, it will ignore the subsequent CSI samples from this STA, until the PSE value of its CSI samples changes. Here, PSE values within a PSE bin are considered the same.

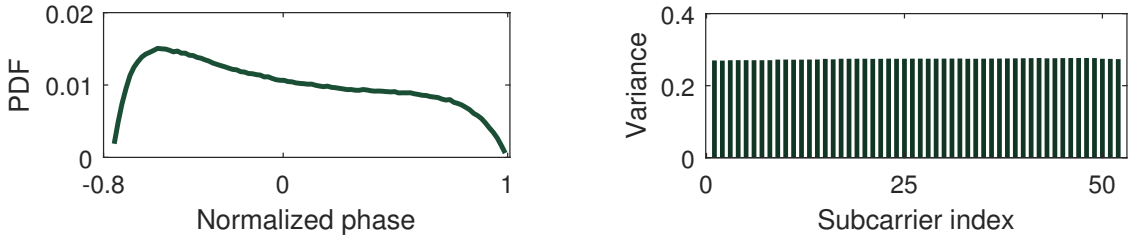
5.4.3 Online Training: Data Preprocessing

After clearing and balancing the collected datasets, the AP preprocesses the datasets before feeding them into DNN-AEs for training. In what follows, we first describe the purpose of data preprocessing and then present the preprocessing procedure for the two sets of angles.

Purpose of Data Preprocessing. To avoid biased training and boost the convergence for DNN-AEs, we wish to obtain the training datasets with a normalized zero-mean probability density function and uniform subcarrier-wise variance in the feasible space [83]. Such datasets are likely



(a) Probability density function and variance before preprocessing.



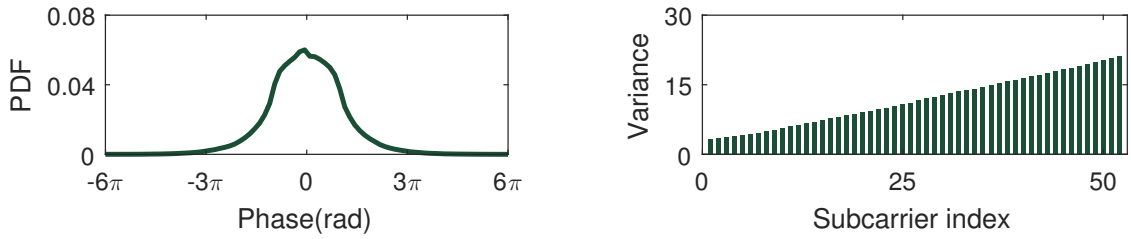
(b) Probability density function and variance after preprocessing.

Figure 5.7: The probability and variance of the angles in $\mathbf{\Psi}$ before and after rectification.

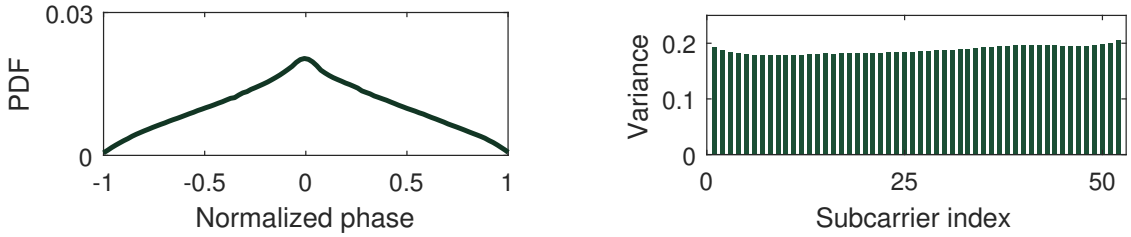
to render an unbiased training for DNN-AEs and yield a high compression ratio. Unfortunately, the collected angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ do not meet these two conditions (normalized zero-mean distribution and flat subcarrier-wise variance). Therefore, we preprocess the collected datasets with the aim of rectifying their distributions to accelerate the training.

Preprocessing of Angles in $\mathbf{\Psi}$. Fig. 5.7(a) shows the probability and variance of the angles in $\mathbf{\Psi}$ before the preprocessing. As it can be seen, the angles in $\mathbf{\Psi}$ are non-uniformly distributed within their range. To alleviate this issue, we apply a rectification function $f(\cdot)$ at the encoder and de-rectification function $f^{-1}(\cdot)$ at the decoder, as shown in Fig. 5.6(a). Here, we employ $f(\psi_k) = \alpha\left(\psi_k - \bar{\psi}\right)$ as the rectification function, where $\bar{\psi}$ is the average of the angles in $\mathbf{\Psi}$ and α is a normalization constant. In our experiments, we use $\bar{\psi} = 0.68$ rad and $\alpha = 1.12$. After the rectification, the angles will have zero mean and uniform variance over different subcarriers, thereby improving the convergence of the DNN-AEs [92] and avoiding zigzag behavior in gradient descent algorithms [93].

Fig. 5.7(b) shows the probability and variance of the angles in $\mathbf{\Psi}$ after the preprocessing. As it



(a) Probability density function and variance before preprocessing.



(b) Probability density function and variance after preprocessing.

Figure 5.8: The probability and variance of the angles in $\mathbf{\Phi}$ before and after rectification.

can be seen, the probability density function has a zero mean after the preprocessing, which leads to a disciplined training for the corresponding DNN-AE.

Preprocessing of Angles in $\mathbf{\Phi}$. Compared to $\mathbf{\Psi}$, the preprocessing of $\mathbf{\Phi}$ is a bit more tricky. Fig. 5.8 shows the probability density function and subcarrier-wise variance of the angles in $\mathbf{\Phi}$ measured in real WLANs. The non-uniform probability distribution, non-uniform variance, high variance on each subcarrier, and the large range (even beyond $[-4\pi,\ 4\pi]$) make the angles in $\mathbf{\Phi}$ unsuited for training. Preprocessing is needed to rectify the dataset to improve the convergence of the DNN-AE and avoid biased training.

One approach that one may think of to rectify the angles is to wrap the angles into $[0,\ 2\pi)$ using a simple function $g(\phi) = \mathrm{mod}(\phi, 2\pi)$. This approach, however, is not an effective one. Fig. 5.9 shows an example of this rectification function. It can be seen that the rectified angle curve appears to be discontinuous. However, the discontinuity of the rectified data cannot be captured by the DNN-AE, as illustrated in the figure. Therefore, a continuous rectification function is needed for the preprocessing of $\mathbf{\Phi}$.

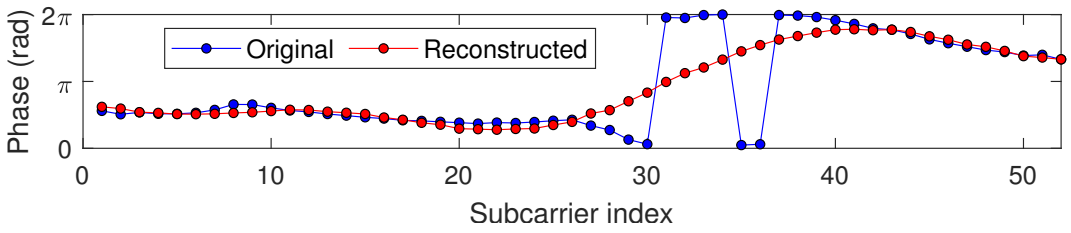


Figure 5.9: Illustrating the underlying problem of the rectification function $g(\phi) = \mathrm{mod}(\phi, 2\pi)$ for the angles in $\mathbf{\Phi}$.

In light of this requirement, we propose a piece-wise function to rectify the angles in $\mathbf{\Phi}$ before feeding them into the DNN-AE:

$$g(\phi_k) = \begin{cases} \frac{1}{2\pi}(\phi_k - 0.07) & \text{if } \min_k(\phi_k) < 0, \\ \frac{1}{2\pi}(\phi_k - 6.16) & \text{if } \max_k(\phi_k) > 2\pi, \\ \frac{1}{\pi}(\phi_k - 3.13) & \text{otherwise,} \end{cases} \tag{5.2}$$

for $k = 1, 2, \cdots, 52$. In this equation, the values of 0.07, 6.16, and 3.13 are the mean of the angles in their respective category and obtained from our experimental measurements. It is noteworthy that coefficient $1/\pi$ in the third equation in (5.2) differs from the other two. This is because the angles in this category have a small range and thus a small normalization coefficient is used for scaling.

Fig. 5.8(b) shows the probability density function and subcarrier-wise variance of all the angles in $\mathbf{\Phi}$ after preprocessing. Compared to the distribution and variance before preprocessing as shown in Fig. 5.8(a), it is evident that this preprocessing can flatten both probability and variance distributions, making the DNN-AE easy to converge.

Given that $g(\phi_k)$ is used for data preprocessing on the encoder side, an inverse function is needed on the decoder side to recover the original angles. However, $g(\phi_k)$ is a piece-wise function and it is not invertible. To address this challenge, we use two bits to indicate the sub-function used

for rectification, i.e., "00" means $g(\phi_k) = \frac{1}{2\pi}(\phi_k - 0.07)$, "01" means $g(\phi_k) = \frac{1}{2\pi}(\phi_k - 6.16)$, and "10" means $g(\phi_k) = \frac{1}{\pi}(\phi_k - 3.13)$. With these two bits, the decoder is capable of constructing $g^{-1}(\phi_k)$ and inversing the preprocessing at the encoder. In the exploitation phase, each STA should send these indication bits to the AP via the over-the-air uplink channel. It is worth noting that these indication bits are of very small size compared to conventional CSI feedback.

5.4.4 Online Training: Settings and Procedure

Training Procedure and Hyper-Parameter Tuning. We train the DNN-AEs shown in Fig. 5.6 using the preprocessed datasets. For the two DNN-AEs, each hidden layer is composed of a fully-connected layer followed by a batch-normalization layer to speed up the training convergence [72]. Also, Rectified Linear Unit (ReLU) activation function is used. The DNN-AEs are trained to minimize loss function, which is defined as the relative error:

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|}, \tag{5.3}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the input sample and the corresponding reconstructed sample, respectively. The networks are trained using Adam optimizer [85]. We started the training with an initial learning rate of 0.001 and reduced it with a decay rate of 0.98 following a step-wise approach. All parameters were initialized using Xavier initialization [50]. Dropout [154] is applied to all hidden layers to prevent over-fitting and improve the generalization of the model. The final architectures are the result of random search over hyper-parameters. All DNN-AEs are trained end-to-end using Pytorch v1.4 library [132].

Readiness of DNN-AEs for Exploitation. While the AP trains the DNN-AEs whenever it receives a batch of CSI reports from the STAs, a natural question to ask is about the criteria for the

completion of its training phase. In our experiments, we check the loss function of validation data to determine the readiness of the DNN-AEs. If the loss function of validation data is consistently less than 1.5%, we consider the completion of the training phase and the DNN-AEs are ready to use. The AP then broadcast the weights and bias values of the encoder parts of the two DNN-AEs as well as the preprocessing parameters to the STAs, so that the STAs can reconstruct the encoder part to compress the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$, as shown in Fig. 5.1(b). Using 32 bits to represent each parameter (real number), the total overhead of transmitting the parameters of the trained DNN-AEs is 5.74 kB, where 1.80 kB is for the parameters of $\mathbf{\Psi}$'s DNN-AE, and 3.94 kB is for the parameters of $\mathbf{\Phi}$'s DNN-AE. This airtime overhead of DNN-AEs broadcast is not an issue for two reasons. First, the broadcast takes place once for a very long period of time. Second, the broadcast is not time-sensitive and the AP can broadcast whenever it gets the resource.

Keep Training DNN-AEs. While the AP has broadcast the DNN-AEs to the STAs, there might be some STAs incapable of utilizing the DNN-AEs for CSI compression. For example, some incumbent STAs may support MU-MIMO but do not support autoencoder-based CSI compression. In such a case, the AP can instruct these STAs to report CSI without compression and use the uncompressed CSI reports for the construction of beamforming filters as that in exiting 802.11 protocols. In the meantime, the AP can use the uncompressed CSI reports from those STAs to keep training the DNN-AEs.

Updating DNN-AEs. During the training in exploitation phase, the AP will periodically use validation data to check the loss function. It rebroadcasts the DNN-AEs to STAs whenever it detects a stable improvement in trained DNN-AEs. Furthermore, the AP rebroadcasts the updated DNN-AEs to the STAs whenever it observes an increase (e.g., 5%) in downlink packet error rate. Such an event simply means that the DNN-AEs in use are outdated. It is noteworthy that we did not observe a failure of the DNN-AEs in our experiments even though we moved the testbed

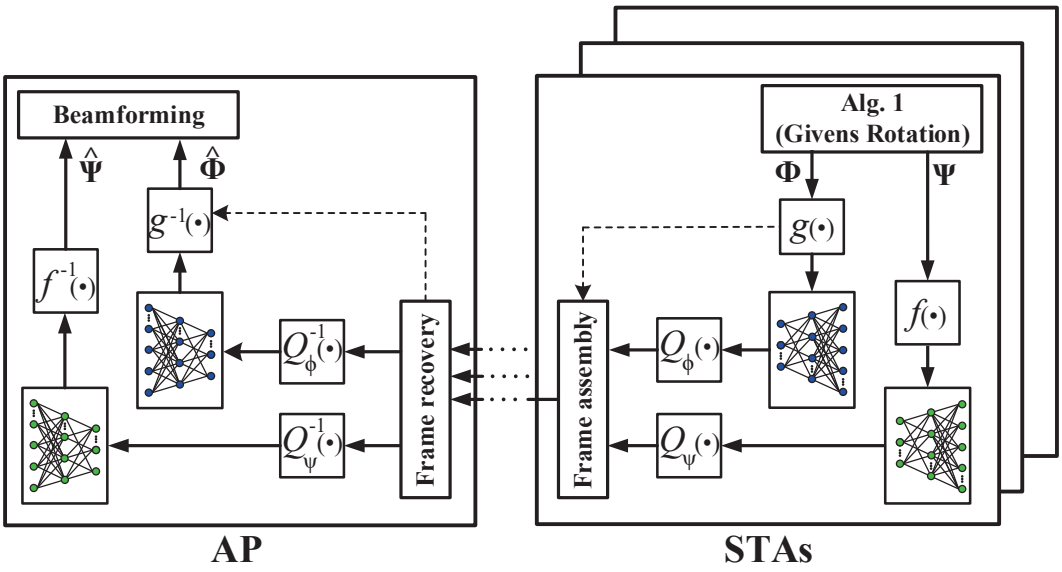


Figure 5.10: CSI compression at STA and decompression at AP.

significantly. We enforce this mechanism just to improve the robustness of our design.

5.4.5 Real-Time DNN-AEs Exploitation: CSI Compression

After the AP completes the training phase, the WLAN enters into the exploitation phase. In this phase, the AP and STAs still use the existing MU-MIMO protocols shown in Fig. 5.2 for downlink MU-MIMO transmissions, except that DNN-AEs are used for CSI compression of the channel feedback. In what follows, we describe CSI compression at an STA and CSI decompression at the AP, respectively.

STA-Side Operations. Fig. 5.10 shows the CSI compression operations at a STA. The STA first estimates the CSI and then converts the estimated CSI to two sets of angles. Then, the two sets of angles are preprocessed and fed into the encoders of DNN-AEs for compression. After that, quantization is performed on the output, followed by frame assembly for uplink CSI report. A question to ask is how many bits should be used for the quantization of the output of DNN-AEs' encoders. While there is no analytical guidance to answer this question, we resort to experimental tests. We found that the angles in Φ are more sensitive to quantization errors than the angles in

$\mathbf{\Psi}$. We also observed that the setting of 5 bits for each output of $\mathbf{\Psi}$'s DNN-AE and 8 bits for each output of $\mathbf{\Phi}$'s DNN-AE is a good trade-off between performance and airtime overhead. In our experiments, we will stick to this quantization setting.

AP-Side Operations. Fig. 5.10 shows the CSI decompression operations at the AP, which try to recover the original CSI based on the compressed angles from an STA. The decompressed CSI will be used to construct the beamforming filters (e.g., using SVD-based precoding methods) for downlink MU-MIMO transmissions.

5.4.6 Compression Ratio and Airtime Overhead

As presented in Section 5.3.1, the existing MU-MIMO protocols employ two types of CSI feedback quantization options and can group different numbers of subcarriers for CSI feedback. Then, the number of bits required for CSI feedback can be expressed as $N_{sc}N_a(p+q)/N_g$, where N_{sc} is the number of valid subcarriers, N_a is the number of angle-sequences in $\mathbf{\Psi}$ or $\mathbf{\Phi}$, p and q are the number of quantization bits as shown in Alg. 5.1, and N_g is the number of subcarriers in a group. Per the IEEE 802.11ac, we have $(p,q) \in \{(5,7),(7,9)\}$, $N_g \in \{1,2,4\}$.

LB-SciFi uses two DNN-AEs to compress the angle sequences in $\mathbf{\Psi}$ or $\mathbf{\Phi}$. Based on the DNN-AE settings and quantization bits as shown in Fig. 5.10, the number of feedback bits is $N_a(5\times 3+8\times 8+2)=81N_a$. Therefore, the compression ratio of LB-SciFi can be written as:

$$compression\_ratio = 1 - \frac{81N_g}{52(p+q)}, \tag{5.4}$$

where $(p,q) \in \{(5,7),(7,9)\}$ and $N_g \in \{1,2,4\}$ as specified in the IEEE 802.11ac [67].

Based on (5.4), it is easy to check that LB-SciFi can achieve significant compression compared to the existing protocols. The compression ratio ranges from 48.1% to 90.3%, depending on the

setting of the existing channel feedback protocol. While LB-SciFi can significantly reduce the quantity of CSI feedback, a question to ask is about the quality of its compressed feedback, including the feedback error and the impact on downlink MU-MIMO. We will provide experimental results to answer this question in the next section.

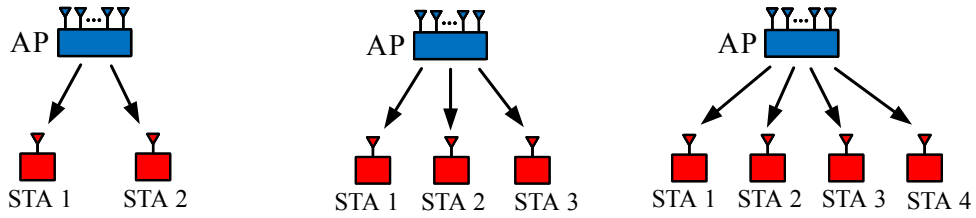
5.4.7 Limitations

Some limitations of LB-SciFi are discussed as follows.

Compression Settings. LB-SciFi involves many parameters such as the number of layers in DNN-AEs, the number of neurons on each layer, the number of bits for quantization, and the pre-processing function parameters. These parameters are empirically chosen in our design, and there is no systematic approach to determine the optimal values of those parameters. So, essentially, LB-SciFi is heuristic and cannot offer any guarantee on its compression loss performance.

Dataset Size. The key phase of LB-SciFi is training the two DNN-AEs. However, there is no guideline on how many data samples suffice for the two DNN-AEs' training. Our experiments show that $13,100$ data samples can achieve at least 98.5% compression accuracy. However, this number is not generic and may change in other network environments. In general, the required dataset size for the DNN-AEs' training remains unknown.

Variability of Physical Environment. When there is a significant change in the surroundings of the AP (e.g., a metal desk placed in front of the AP or the AP is moved into a distinct environment), re-training will be triggered to update the DNN-AEs. LB-SciFi cannot offer a time guarantee on the re-training as it depends on the speed of data collection.



(a) Two-user case. (b) Three-user case. (c) Four-user case.

Figure 5.11: Experimental setup for downlink MU-MIMO.

5.5 Experimental Evaluation

In this section, we evaluate the performance LB-SciFi in comparison with existing 802.11 protocols in an indoor wireless environment. For ease of exposition, we use 802.11-TiGj to denote the IEEE 802.11 MU-MIMO protocol with Type i feedback and j subcarriers in a group, where $i \in \{0, 1\}$ and $j \in \{1, 2, 4\}$ (see Section 5.3.1 and [67]). Since T1G1 represents the finest feedback and T0G4 represents the coarsest feedback, we will use these two protocols as our performance comparison baseline.

5.5.1 Experimental Setup and Implementation

Downlink MU-MIMO. We consider a WLAN as shown in Fig. 5.11, where the AP can serve two, three, or four STAs simultaneously in downlink. While there are many different beamforming methods in the literature, we used ZF beamforming method in our experiments owing to its popularity and ease of implementation.

Implementation of AP and STAs. Fig. 5.12(a–b) shows our wireless testbed. The AP and STAs are built using USRP N210 devices and general-purpose computers. Each USRP N210 device is equipped with VERT2450 Antenna for radio signal transmissions at 2.484 GHz. The computers are used for baseband signal processing and MAC protocol implementation. More specifically, the AP is implemented using a Dell Inspiron 3671 Desktop, which serves eight USRP N210

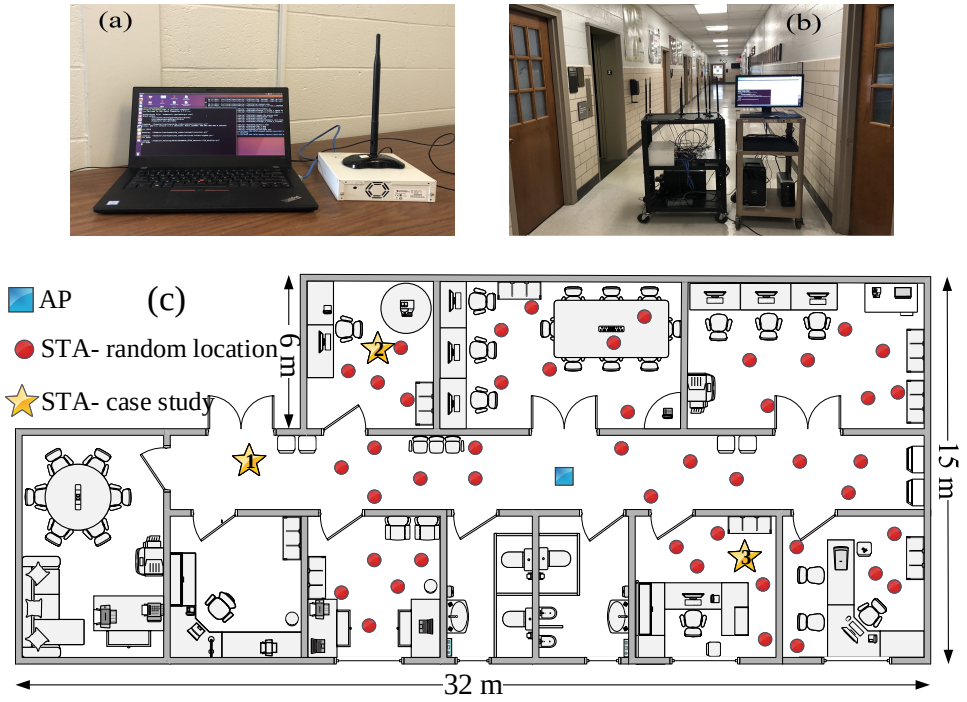


Figure 5.12: Illustrating our wireless testbed and test environment. (a) Prototyped STA. (b) Prototyped AP. (c) Floor plan of tests.

devices through a 10Gb fiber optic cable and a DGS-1210-20/ME Ethernet switch. Each STA is prototyped with a Lenovo ThinkPad T480 and one USRP N210 device.

Implementation of 802.11 Protocols. IEEE 802.11 protocols are implemented with the legacy PHY and MAC layers specifications. We use IEEE 802.11 frame format with 64 subcarriers for OFDM modulation. Out of these 64 subcarriers, 48 subcarriers carry payload and 4 subcarriers contain pilots. The sampling rate and carrier frequency are set to 20 MSps and 2.484 GHz, respectively. Also, the maximum transmission power is set to 15 dBm. All the necessary 802.11 baseband signal processing modules are realized with C++ in GNU Radio. For ease of implementation, our 802.11 protocols do not include user scheduling.

Implementation of LB-SciFi. LB-SciFi is implemented on top of 802.11 protocols. It mainly deals with collecting datasets and training DNN-AEs. On our testbed, the training datasets are automatically generated in the 802.11 protocols. With the collected datasets, DNN-AEs are trained end-to-end using Pytorch v1.4 library [132] and Adam optimizer [85].

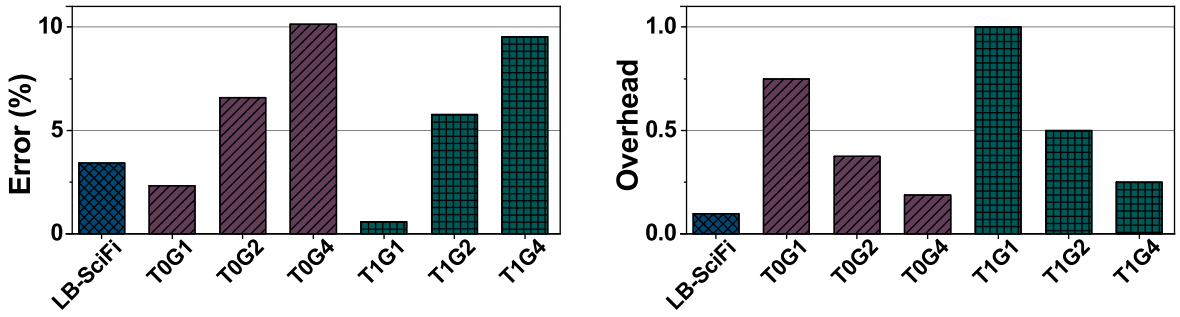
Experimental setting. Fig. 5.12(c) shows an office scenario where we conducted the experiments. The AP is placed at the spot marked as a blue square in the figure, while each STA is placed at a random location marked as a red circle.

5.5.2 DNN-AEs Training and Feedback

Data Collection Campaign. We ran the MU-MIMO communications shown in Fig. 5.11(c) to collect data for DNN-AEs training at the AP. Specifically, the AP performs downlink MU-MIMO communications using the 802.11 protocols and, at the same time, it takes advantage of the CSI reports from the STAs for DNN-AEs training. The data collection campaign was conducted during two business days from 10am to 8pm. The human activity level in the environment was high between 11am to 2pm and low to moderate in other periods of time. To cover all areas, we were moving the STAs around all locations. This can be achieved in real systems thanks to the mobility of some Wi-Fi devices such as phones and laptops. In our experiments, the AP eventually collected $60,000$ samples from the STAs for training the DNN-AEs.

Sufficiency of Collected Data. A question to ask is whether $60,000$ samples suffice for DNN-AEs' training. To answer this question, we conduct convergence test under two criteria: i) the test loss of DNN-AE should be less than 1.5%, and ii) the loss difference for two validations should be less than 0.1%. With such two criteria, $\mathbf{\Psi}$'s DNN-AE converges with $7,300$ samples, and $\mathbf{\Phi}$'s DNN-AE converges with $13,100$ samples. This indicates that $60,000$ samples suffice for training.

Computational Complexity of Training. In our experiments, the training process takes less than 5 seconds on a Desktop PC with i5 CPU and 16 GB memory. A question is how much time is needed for training DNN-AEs on a commodity AP (Wi-Fi router). Since most commodity APs are equipped with an ARM processor, we expect that a commodity AP may take minutes to complete the training. In addition, we note that the training process is not time-sensitive, and an AP can take



(a) Normalized feedback error.

(b) Normalized feedback overhead.

Figure 5.13: Feedback comparison between LB-SciFi and 802.11 protocols (T0G1, T0G2, T0G4, T1G1, T1G2, and T1G4).

its spare time to complete the training. If an AP is not capable of doing the training by itself, it can take advantage of its wired Internet connection and a cloud server to run the training.

Feedback Error. With the completion of the first training, we examine the performance of the DNN-AEs. LB-SciFi introduces CSI error during the feedback. The feedback error can be attributed to two sources: *compression* and *quantization*. The compression error comes from the imperfection of the DNN-AEs, and the quantization error comes from the limited quantization bits. The normalized feedback error can be quantified by the loss function in (5.3). As a comparison baseline, we also measure the normalized feedback error in 802.11 protocols, where the error comes from the quantization of $\mathbf{\Psi}$ and $\mathbf{\Phi}$ as well as the subcarrier grouping.

Fig. 5.13(a) shows our measured normalized feedback errors. It can be seen that LB-SciFi has a larger feedback error than 802.11-T0G1/T1G1 protocols, and it has a smaller feedback error compared to 802.11-T0G2/T0G4/T1G2/T1G4 protocols. This is because 802.11-T0G1/T1G1 protocols do not compress the CSI in the spectral domain while other protocols naively compress CSI in the spectral domain.

Feedback Overhead. While LB-SciFi introduces larger error than 802.11-T0G1 and 802.11-T1G1, it uses much smaller uplink airtime resource for CSI feedback and therefore entails much smaller overhead. Fig. 5.13(b) compares the normalized feedback overhead of LB-SciFi with the

existing 802.11 protocols. It can be seen that LB-SciFi entails much less overhead compared to 802.11 protocols. LB-SciFi's overhead is 0.1 while the lowest normalized overhead among IEEE 802.11 protocols is 0.2. Also, LB-SciFi's compression ratio ranges from 48.1% to 90.3%, thereby conserving much airtime resource for data transmissions.

5.5.3 LB-SciFi: Performance Metrics

We now focus on the overall performance of downlink MU-MIMO. We will consider the following performance metrics.

EVM. EVM is widely used to assess the quality of received signals at a receiver device. It is defined as follows: $\mathrm{EVM} = 10\log_{10}\left(\frac{\mathbb{E}[|\hat{X}-X|^2]}{\mathbb{E}[|X|^2]}\right)$, where X and $\hat{X}$ are the original and estimated signals on a subcarrier of an OFDM symbol, respectively.

Gross Throughput. Gross throughput refers to the data rate achieved by a device (AP or STA) without taking into account the CSI overhead. For STA i, based on the EVM of its decoded signal, its gross throughput can be extrapolated as follows: $r_i = \frac{N_{\mathrm{sp}}}{N_{\mathrm{fft}}+N_{\mathrm{cp}}} \cdot b \cdot \gamma\left(\mathrm{EVM}_i\right)$, where $N_{\mathrm{sp}} = 48$ is the number of subcarriers carrying payload, $N_{\mathrm{fft}} = 64$ is FFT points, $N_{\mathrm{cp}} = 16$ is the length of cyclic prefix, $b = 20$ is the sampling rate, and EVM_i is EVM of the STA i's decoded signal, and $\gamma(\mathrm{EVM}_i)$ is the average number of bits carried by one subcarrier. This parameter is given in Table 2.2. As such, the gross throughput at the AP can be computed by $r = \sum_i r_i$.

Net Throughput. The net throughput refers to the data rate achieved by a device after subtracting the overhead mainly caused by CSI feedback in the MU-MIMO protocols. Denote $\bar{r}$ as the net throughput achieved by the AP. Then, it can be expressed by: $\bar{r} = \frac{\sum_i t_i r_i}{\max_i\{t_i\}+t_{\mathrm{overhead}}}$, where t_{overhead} is the time duration of overhead (NDPA, NDP, Poll, CBR, and ACK) and t_i is the time duration required by STA i for its downlink data transmission (see Fig. 5.2). While the value of t_{overhead} is fixed, the value of t_i is not. t_i is determined by the downlink data packet

size and selected modulation and coding scheme. In real WLANs, a data packet should not exceed 2304 bytes [135]. In our experiments, we consider the maximum packet size to measure the lowest throughput gain that can be achieved by LB-SciFi.

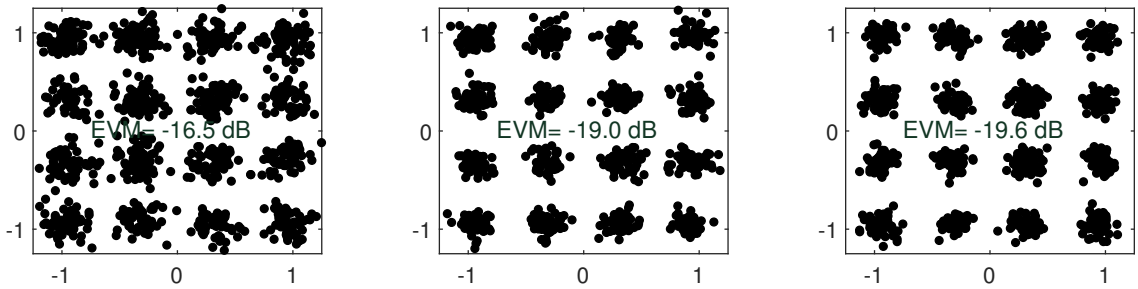
5.5.4 Micro Performance of LB-SciFi: A Case Study

We use a case study to examine the micro performance of LB-SciFi. We consider the network shown in Fig 5.11(b) and place the three STAs at the spots marked with golden stars in Fig. 5.12(c). We compare the performance of LB-SciFi with 802.11-T1G1/T0G4 protocols.

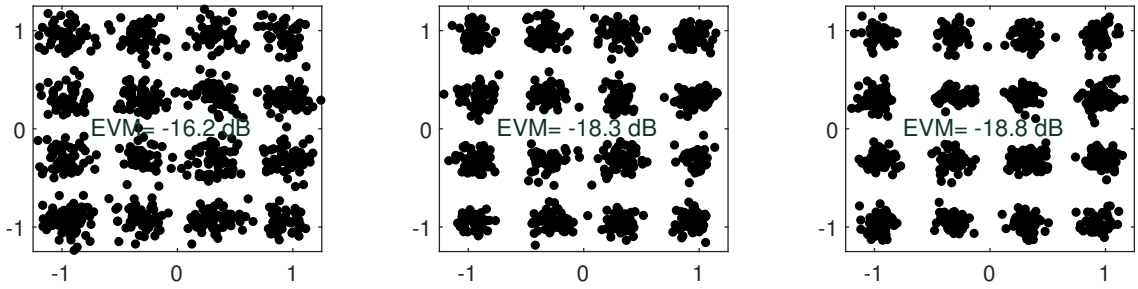
EVM.: We conduct downlink MU-MIMO transmissions using LB-SciFi, 802.11-T0G4, and 802.11-T1G1. Fig. 5.14 exhibits the constellation of the decoded data packet at each STA with the three protocols. As shown in Fig. 5.14(a), LB-SciFi achieved -16.5 dB EVM at STA 1, -19.0 dB EVM at STA 2, and -19.6 dB EVM at STA 3. In contrast, Fig. 5.14(b) shows the achieved EVM at the three STAs when 802.11-T0G4 is used; and Fig. 5.14(c) shows the achieved EVM at the three STAs when 802.11-T1G1 is used. It can be seen that LB-SciFi achieves an EVM performance similar to 802.11-T1G1 and outperforms 802.11-T0G4. We note that the constellations in Fig. 5.14 can be successfully decoded thanks to the powerful LDPC channel code. It is also worth pointing out that LB-SciFi can support any modulation and coding scheme as long as channel quality permits.

Feedback Overhead. In the MU-MIMO transmissions, the CSI reports are transmitted from STAs to the AP using BPSK rate to ensure the feedback reliability [110]. Table 5.1 lists the feedback overhead using different protocols. As we can see from the table, LB-SciFi entails 0.6 kbit feedback overhead per STA. In contrast, 802.11-T0G4 entails 1.1 kbit feedback overhead per STA, and 802.11-T1G1 entails 5.8 kbit feedback overhead per STA.

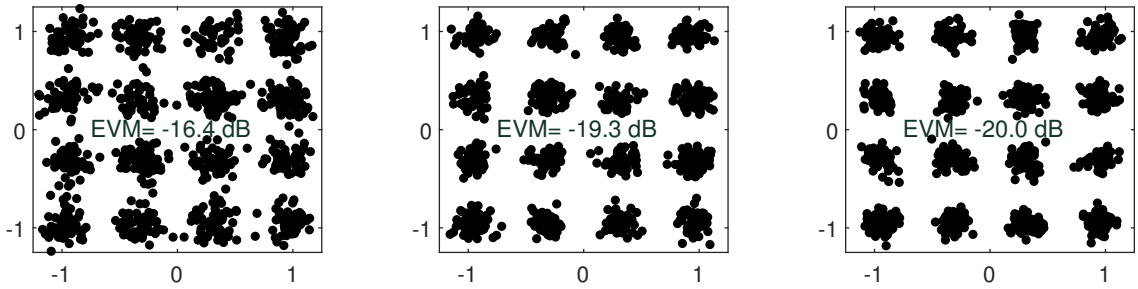
Gross and Net Throughput. Table 5.1 lists each STA's and the AP's gross/net throughput.



(a) LB-SciFi: Constellation measured at three STAs.



(b) 802.11-T0G4: Constellation measured at three STAs.

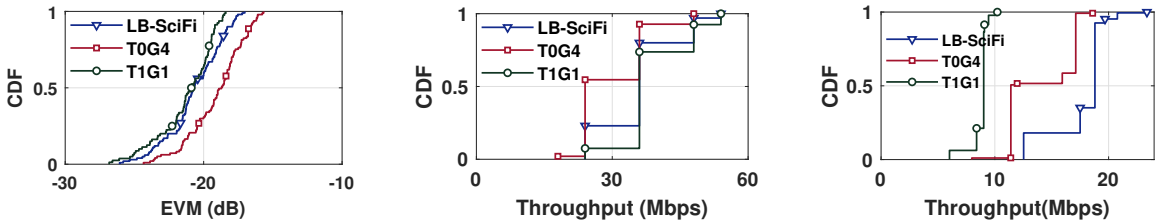


(c) 802.11-T1G1: Constellation measured at three STAs.

Figure 5.14: Constellations of decoded signals at STAs when using LB-SciFi, 802.11-T0G4, and 802.11-T1G1.

Table 5.1: Experimental results of the case study for LB-SciFi and 802.11 protocols.

		STA 1	STA 2	STA 3	AP
LB-SciFi	EVM (dB)	-16.5	-19.0	-19.6	–
	Feedback overhead (kbit)	0.6	0.6	0.6	–
	Gross throughput (Mbps)	24.0	36.0	36.0	96.0
	Net throughput (Mbps)	15.9	23.9	23.9	63.7
T0G4	EVM (dB)	-16.2	-18.3	-18.8	–
	Feedback overhead (kbit)	1.1	1.1	1.1	–
	Gross throughput (Mbps)	24.0	24.0	24.0	72.0
	Net throughput (Mbps)	15.0	15.0	15.0	45.0
T1G1	EVM (dB)	-16.4	-19.3	-20.0	–
	Feedback overhead (kbit)	5.8	5.8	5.8	–
	Gross throughput (Mbps)	24.0	36.0	36.0	96.0
	Net throughput (Mbps)	9.4	14.1	14.1	37.6



(a) EVM (b) Gross throughput (c) Net throughput

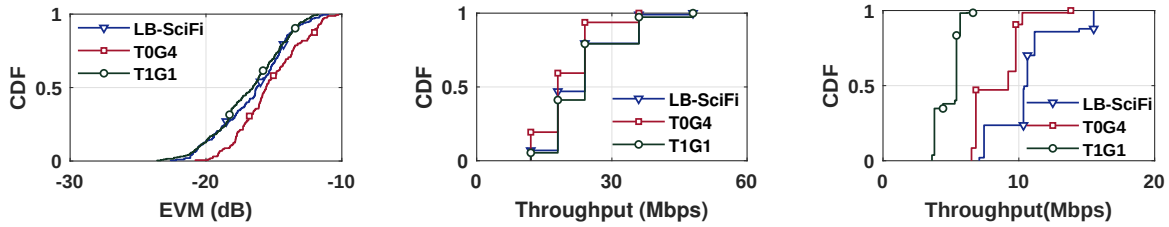
Figure 5.15: Comparison of LB-SciFi and 802.11 protocols in the two-user MU-MIMO network. We can see that LB-SciFi's gross throughput is larger than 802.11-T0G4 but less than 802.11-T1G1. However, LB-SciFi's net throughput is larger than both of them. The overall net throughput gain of LB-SciFi is 41.7% over 802.11-T0G4 and 68.8% over 802.11-T1G1.

5.5.5 Macro Performance of LB-SciFi: Extensive Results

We now extend our case study to a more generic scenario. We consider the three networks in Fig. 5.11 and measure their performance at many different locations as shown in Fig. 5.12. Our evaluation methodology follows the previous case study.

Two-User MIMO. Fig. 5.15 presents the CDF of our measured EVM, gross throughput, and net throughput over all locations when the AP serves two STAs. Per Fig. 5.15(a), the average EVM of decoded signals at the two STAs is -20.7 dB for LB-SciFi, -19.1 dB for 802.11-T0G4, and -21.2 dB for 802.11-T1G1. Compared to T0G4, LB-SciF has 1.6 dB EVM improvement. Compared to T1G1, LB-SciF has 0.5 dB EVM degradation. Per Fig 5.15(b), LB-SciFi achieves an average of 35.8 Mbps per-STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 30.2 Mbps and 38.7 Mbps, respectively. Per Fig 5.15(c), LB-SciFi achieves an average of 17.6 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 14.1 Mbps and 8.8 Mbps, respectively. The results indicate that LB-SciFi offers 25.0% net throughput gain over 802.11-T0G4 and 99.8% gain over 802.11-T1G1.

[187] proposed a 3-dimensional (time, frequency, and quantization) Adaptive Feedback Com-



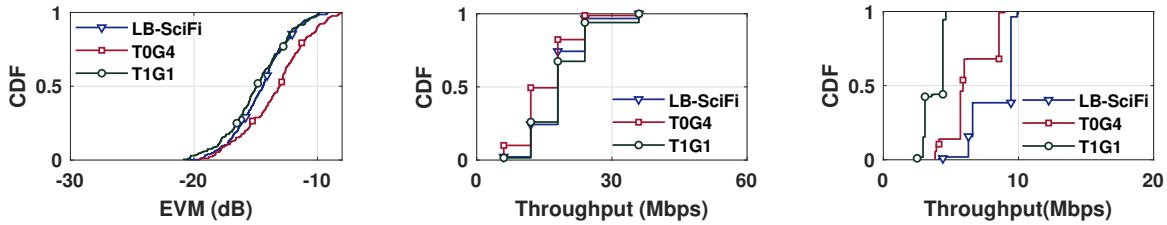
(a) EVM. (b) Gross throughput. (c) Net throughput.

Figure 5.16: Comparison of LB-SciFi and 802.11 protocols in the three-user MU-MIMO network.

pression (AFC) scheme for WLANs. While LB-SciFi is orthogonal to the time-domain AFC, we compare LB-SciFi with the frequency-domain AFC. Experimental results in [187] show the frequency-domain AFC achieves 12.7% throughput gain when compared to 802.11-T1G1 ("Size1" in its Fig. 13b). LB-SciFi achieves an average of 99.8% throughput gain over 802.11-T1G1. The comparison result is not surprising because LB-SciFi exploits DNN-AEs to reduce channel's inter-subcarrier correlation for feedback compression, rather than simply grouping a subset of subcarriers for feedback compression.

Three-User MIMO. Fig. 5.16 presents the CDF of our measured EVM, gross throughput, and net throughput over all locations when the AP serves three STAs. Per Fig 5.16(a), the average EVM of decoded signals at the three STAs is -16.5 dB for LB-SciFi, -15.3 dB for 802.11-T0G4, and -16.8 dB for 802.11-T1G1. Per Fig 5.16(b), LB-SciFi achieves an average of 23.3 Mbps per-STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 20.0 Mbps and 24.0 Mbps, respectively. Per Fig 5.16(c), LB-SciFi achieves an average of 10.5 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 8.4 Mbps and 4.9 Mbps. Therefore, LB-SciFi offers 25.7% net throughput gain over 802.11-T0G4 and 116.8% net throughput gain over 802.11-T1G1.

Four-User MIMO. Fig. 5.17 presents the CDF of our measured EVM, gross throughput, and net throughput over all the locations when the AP serves two STAs. Per Fig 5.17(a), the average EVM of decoded signals at the four STAs is -14.5 dB for LB-SciFi, -13.4 dB for 802.11-T0G4, and -14.9 dB for 802.11-T1G1. Per Fig 5.17(b), LB-SciFi achieves an average of 18.3 Mbps per-



(a) EVM. (b) Gross throughput. (c) Net throughput.

Figure 5.17: Comparison of LB-SciFi and 802.11 protocols in the four-user MU-MIMO network.

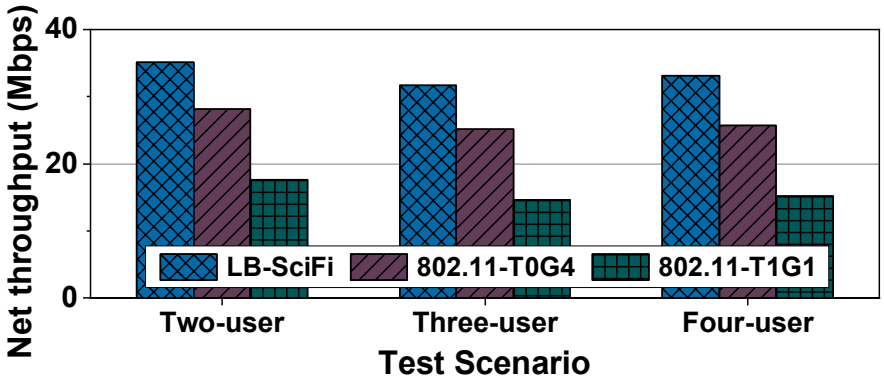


Figure 5.18: Net throughput of LB-SciFi and 802.11 protocols.

STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 15.6 Mbps and 19.0 Mbps. Per Fig 5.17(c), LB-SciFi achieves an average of 8.3 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 6.4 Mbps and 3.8 Mbps, respectively. LB-SciFi offers 28.9% net throughput gain over 802.11-T0G4 and 117.3% net throughput gain over 802.11-T1G1.

Summary of Observations. We now focus on the net throughput achieved by the AP. Fig. 5.18 depicts the total net throughput achieved by the AP when it employs these three protocols. As it can be seen, the three protocols yield similar throughput in two-user, three-user, and four-user MIMO cases. On average, LB-SciFi achieves 26.5% net throughput gain compared to 802.11-T0G4 and 111.3% throughput gain over 802.11-T1G1.

5.6 Chapter Summary

In this chapter, we presented LB-SciFi, an online learning-based channel feedback framework for existing IEEE 802.11 MU-MIMO protocols. LB-SciFi reduces the CSI feedback overhead for 802.11 protocols by leveraging recent advances in deep neural networks to compress CSI in the spectral domain without compromising the CSI feedback accuracy. The key component of LB-SciFi is an online training scheme, which requires no dedicated training datasets but takes advantage of available side information from existing 802.11 protocols to train the autoencoders. As such, LB-SciFi can be easily plugged into existing 802.11 protocols and thus amenable to practical implementation. We have built a prototype of LB-SciFi on a wireless testbed and evaluated its performance in indoor wireless environments. Experimental results show that LB-SciFi can reduce the feedback overhead by 73% and increases the network throughput by 69% on average.

Chapter 6

A Learning-Based Channel Sounding and Resource Allocation for IEEE 802.11ax

6.1 Introduction

After two decades of evolution from its genesis, Wi-Fi technology has become the dominant carrier of the Internet traffic [181] and penetrated every aspect of our lives. With the continuous proliferation of the Internet-based applications, Wi-Fi market is growing at an unprecedented rate, and more than four billion Wi-Fi devices have shipped in 2019 alone [181]. To serve the large number of Wi-Fi devices and meet their high data rate demands, Wi-Fi networks are evolving from 802.11n/ac to 802.11ax so that a Wi-Fi AP is capable of utilizing the spectrum more efficiently and accommodating more Wi-Fi clients at the same time. Compared to the carrier-sense-based 802.11n/ac, 802.11ax features centralized resource allocation and fine-grained inter-device synchronization. With these two features, it introduces OFDMA and uplink MU-MIMO techniques for the first time.

Although OFDMA and MU-MIMO has been well studied in cellular networks (see Table 6.1), their joint optimization in Wi-Fi networks remains scarce because OFDMA is introduced to Wi-Fi networks in 802.11ax for the first time. Given that cellular and Wi-Fi networks have different PHY and MAC layers, and that BSs and APs have very different computational power, the MU-

MIMO-OFDMA transmission schemes designed for cellular networks may not be suited for Wi-Fi networks, necessitating research efforts to innovate the MU-MIMO-OFDMA design for 802.11ax networks. Particularly, the MU-MIMO-OFDMA transmission in 802.11ax faces two challenges. *First*, to perform downlink MU-MIMO transmissions, an AP needs to have CSI for the construction of beamforming filters so that it can concurrently send independent data streams to multiple Wi-Fi clients on the same Resource Unit (RU). However, existing 802.11 channel sounding protocols are notorious for their large airtime overhead, which significantly compromises the throughput gain of MU-MIMO. Therefore, a low-overhead channel sounding protocol is needed. *Second*, the marriage of MU-MIMO and OFDMA largely expands the optimization space of resource allocation at an 802.11ax AP, making it infeasible to pursue an optimal resource allocation solution in real time due to the limited computational power of APs. Therefore, a low-complexity, yet efficient, algorithm is needed for an AP to solve the resource allocation problem.

In this chapter, we study the channel sounding and resource allocation problems for downlink transmissions in an 802.11ax Wi-Fi network, where an AP serves many STAs on a set of pre-defined RUs jointly using MU-MIMO and OFDMA techniques. We assume that the AP is equipped with multiple antennas, while each STA is equipped with one antenna. In such an 802.11ax network, we propose a practical scheme, called DeepMux, to enhance the efficiency of downlink MU-MIMO-OFDMA transmissions by leveraging recent advances in DL. DeepMux addresses the above two challenges using DNNs, and it mainly comprises the following two key components: DLCS and DLRA. Both of them reside in APs and impose no computational/communication burden to the STAs.

To reduce the channel sounding overhead, DLCS in DeepMux compresses the frequency-domain CSI during the feedback procedure by leveraging the compression capability of DNNs. Specifically, instead of reporting CSI on all the grouped tones, each STA only reports the quan-

tized CSI on a *small* number of tones to the AP. Based on the limited CSI, the AP infers CSI over all tones using well-trained DNNs. Particularly, the AP takes advantage of channel reciprocity and uses uplink CSI, which is easy to obtain, to train the DNNs for downlink CSI, making the training process easy to conduct.

To obtain a near-optimal resource allocation solution in real time at the AP, DLRA in DeepMux employs a DNN to solve a Mixed-Integer Non-Linear Programming (MINLP) optimization problem. Specifically, DLRA decouples the complex resource allocation optimization problem into two sub-problems: RU assignment and power allocation. A DNN is then employed to compute a sub-optimal solution to the RU assignment sub-problem. Once RU assignment is determined, the original MINLP problem degrades to a Linear Programming (LP) problem, which is easy to solve.

The contributions of this work are summarized as follows.

- We have designed DLCS, a DL-based channel sounding protocol for 802.11ax networks. DLCS employs an online training process and requires no efforts from STAs. Numerical results show that DLCS is capable of reducing the channel sounding overhead by $62.0\%\sim90.5\%$ without sacrificing CSI feedback accuracy.

- We have designed DLRA, a DL-based resource allocation algorithm for 802.11ax APs to perform efficient downlink transmissions. Numerical studies show that DLRA is capable of yielding a sub-optimal solution to MINLP resource allocation problems in polynomial time.

- By combining DLCS and DLRA, we have designed DeepMux to enable efficient downlink MU-MIMO-OFDMA transmissions in 802.11ax networks. We have evaluated DeepMux on a wireless testbed. Experimental results show that DeepMux improves network throughput by $26.3\%\sim43.6\%$ compared the greedy utilization of DoF by strongest STAs on each RU.

6.2 Related Work

We focus our literature review on channel sounding and resource allocation in both Wi-Fi and cellular networks.

6.2.1 Channel Sounding

Channel Sounding for Wi-Fi. The sounding overhead issue in Wi-Fi networks has been in focal point of view since accommodation of MU-MIMO in IEEE 802.11 standards. Existing research efforts have been invested to tackle this issue by optimizing channel sounding parameters [17, 33, 113], seeking new channel sounding paradigms [53, 110], or compressing CSI frames [79, 187]. As the pioneering trials of reducing sounding overhead, research efforts in [17, 33, 113] have exploited the semi-static nature of Wi-Fi networks to adaptively reduce the frequency of channel sounding and avoid unnecessary sounding overhead. Implicit channel sounding has also been studied for rectifying sounding overhead [53, 110]. Although implicit channel sounding can significantly lower the overhead, it requires extra hardware for channel calibration and thus may not be amenable to low-cost Wi-Fi networks. DeepMux is orthogonal to these works as DLCS neither manipulates the channel sounding frequency nor employs implicit channel sounding.

[187] and [79] are two prior efforts that reduce the channel sounding overhead by compressing CSI in the frequency domain. However, these two efforts require coordination from Wi-Fi clients to fully or partially compress CSI. In contrast, DLCS runs solely on Wi-Fi routers and requires no coordination from Wi-Fi clients. Simply put, DLCS is transparent to Wi-Fi clients. DLCS also differs from these two works in terms of computational complexity. Specifically, [187] and [79] require Wi-Fi clients to estimate CSI for all frequency tones while DLCS requires Wi-Fi clients to

Table 6.1: A summary of resource allocation schemes in Wi-Fi and cellular networks.

	Objective				Network		Mode		MU-MIMO	Polynomial complexity
	sum-rate	Fairness	Latency	Energy	Wi-Fi	Cellular	Uplink	Downlink		
DeepMux	✓				✓			✓	✓	$\mathcal{O}\left(n^{2.5}\right)$
[41]	✓	✓			✓			✓		$\mathcal{O}\left(n^{3}\right)$
[88]	✓				✓		✓	✓		
[169, 170]	✓				✓			✓	✓	
[71]			✓		✓		✓			
[16]		✓			✓		✓			$\mathcal{O}\left(n^{3}\right)$
[184]	✓				✓		✓			
[121]	✓		✓			✓		✓	✓	
[186]				✓		✓		✓		$\mathcal{O}\left(n^{3}\right)$
[58]				✓		✓		✓	✓	
[116]				✓		✓		✓	✓	$\mathcal{O}\left(n^{2.5}\right)$
[45, 76, 141]	✓					✓		✓	✓	

estimate CSI only for a small number of tones.

Learning-Based Channel Sounding in Cellular Networks. Sounding overhead is also a critical problem in cellular networks. Temporal correlation [97, 111, 172] and spatial correlation [97] have been harvested to remove the redundancy of CSI and reduce the airtime overhead of CSI acquisition. DeepMux differs from these works as it focuses on the frequency domain. Frequency-domain correlation of CSI has been studied in [179] and [54] to reduce the channel sounding overhead in cellular networks. DeepMux differs from these works because DLCS is transparent to users (i.e., imposing no computation on users). In addition, CSI in cellular networks is very different from that in Wi-Fi networks. DeepMux is meticulously tailored for Wi-Fi networks. Finally, most prior works are limited to theoretical investigations and numerical evaluations while DeepMux takes into account incumbent Wi-Fi protocols and has been validated in practical indoor wireless environments.

6.2.2 Resource Allocation

Table 6.1 summarizes existing resource allocation schemes in cellular and Wi-Fi networks, where n denotes the number of active users served by an AP or a BS. Clearly, DeepMux differs from

existing works in terms of objective, network scenario, transmission mode, or computational complexity. In what follows, we elaborate the existing studies and point out the differences between DeepMux and these works.

Resource Allocation for Wi-Fi Networks. Recently, [178] has studied downlink OFDMA in wireless local area networks (WLANs) and showed that its performance is highly dependent on the resource assignment strategies at APs. This problem has been followed in [41], with the objective of improving the fairness among users. DLRA differs from the proposed resource allocation scheme in [41] as it focuses on pursuing a sub-optimal resource allocation scheme with a low computational complexity. [88] has considered the throughput maximization under the assumption that a user can be assigned to at most one RU and offered a solution for both uplink and downlink transmissions. Compared to [88], DLRA expands the problem scope by allowing multiple RUs to serve a user and also by allowing an RU to serve multiple users concurrently. [169] and [170] are the only works considering downlink MU-MIMO-OFDMA in WLANs. However, these two works employ greedy iterative algorithms to compute a feasible solution. In contrast, DLRA employs learning-based approach and offers a solution in polynomial time. [16, 71, 184] studied resource allocation in uplink OFDMA WLANs, which is not the scope of our work.

Resource Allocation in Cellular Networks. Since there are many research results of resource allocation in cellular networks, we focus our review on MIMO-OFDMA techniques. [121] has studied the resource allocation problem under latency constraint. However, the complexity of the proposed solution is prohibitively large. [186] has studied the resource allocation problem with the objective of enhancing energy efficiency. The authors has proposed an algorithm with polynomial-time complexity. However, it only works for single-user MIMO-OFDMA networks. [58] and [116] have investigated the resource allocation problem for MU-MIMO-OFDMA cellular networks and proposed low-complexity algorithms to compute the solutions. However, these two

works focus on maximizing energy efficiency. In contrast, DeepMux aims to maximize network throughput. [45, 76, 141] have explored downlink MU-MIMO-OFDMA transmissions in different network scenarios. These research efforts have proposed greedy algorithms to pursue optimal solutions for maximizing network throughput. DeepMux is very different from these works in terms of network settings and computational complexity.

6.3 Problem Description

Consider an 802.11ax network comprising a multi-antenna AP and many single-antenna STAs. Denote N_{ap} as the number of antennas on the AP. Denote N_{sta} as the number of STAs in the network. We consider a dense network where $N_{\mathrm{sta}} > N_{\mathrm{ap}}$. In 802.11ax standard, OFDMA and MU-MIMO techniques have been included for efficient communications between the AP and its serving STAs. Fig. 6.1 shows the four possible RU configurations when the network works on 20 MHz bandwidth. As the figure shows, the total number of valid tones is 242, and an RU could consists of 26, 52, 106, or 242 tones. When MU-MIMO is enabled, an RU can serve multiple STAs, depending on the channel condition, data traffic, and network setting. In the downlink transmissions, in order for an AP to serve multiple STAs per RU, it needs to first perform channel sounding to obtain the CSI and then construct the spatial filters for beamforming. By doing so, independent data streams can be delivered to different STAs simultaneously. In this process, CSI is crucial. In what follows, we first present the existing channel sounding protocol and then state our design objectives.

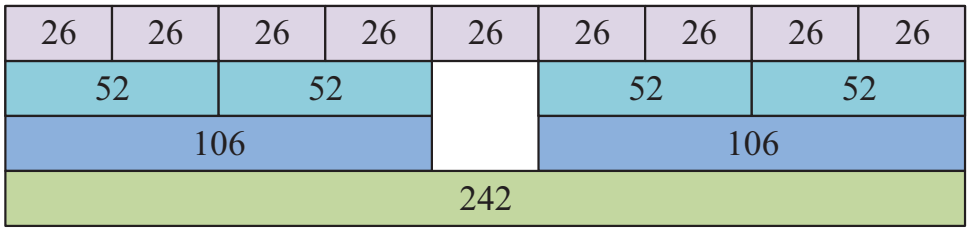


Figure 6.1: Four different RU configurations over 20 MHz as specified in IEEE 802.11ax [70].

6.3.1 802.11 Channel Sounding Protocol in Nutshell

Fig. 6.2 shows the channel sounding protocol specified in 802.11ax, and we elaborate on it in the following.

Announcement. The AP initiates the channel sounding procedure by broadcasting an NDPA frame, which contains the addresses of intended STAs. Then, the AP sends out an NDP frame for STAs to estimate the downlink channels between themselves and the AP.

Channel Estimation. Each STA leverages the preamble in the NDP frame to estimate the complex-valued channel vectors between the AP and itself. Reporting the raw channel vectors to the AP, however, entails too much airtime overhead. To reduce the airtime overhead, each STA employs GR and *tone grouping* to pre-process its estimated channel vectors. The pre-processing leads to a CSI compression in both spatial and spectral domains.

Spatial compression. In its general form, the spatial compression includes a series of GR, pre-multiplications, and post-multiplications applied to the right singular vectors of a channel matrix to extract its spatial information [67, 68, 70]. Each rotation or pre-multiplication is realized by an angle, which stores a part of spatial information [156]. On each tone, two sets of angles will be generated: N_ψ ψ-type angles from GR and N_ϕ ϕ-type angles from pre-multiplications, where $N_\psi = N_\phi = \left(2N_{\text{ap}}N_r - N_r^2 - N_r\right)/2$ and N_r is the number of the STA's antennas in general case (we assumed $N_r = 1$ throughout this chapter). For notional simplicity, we denote these two sets

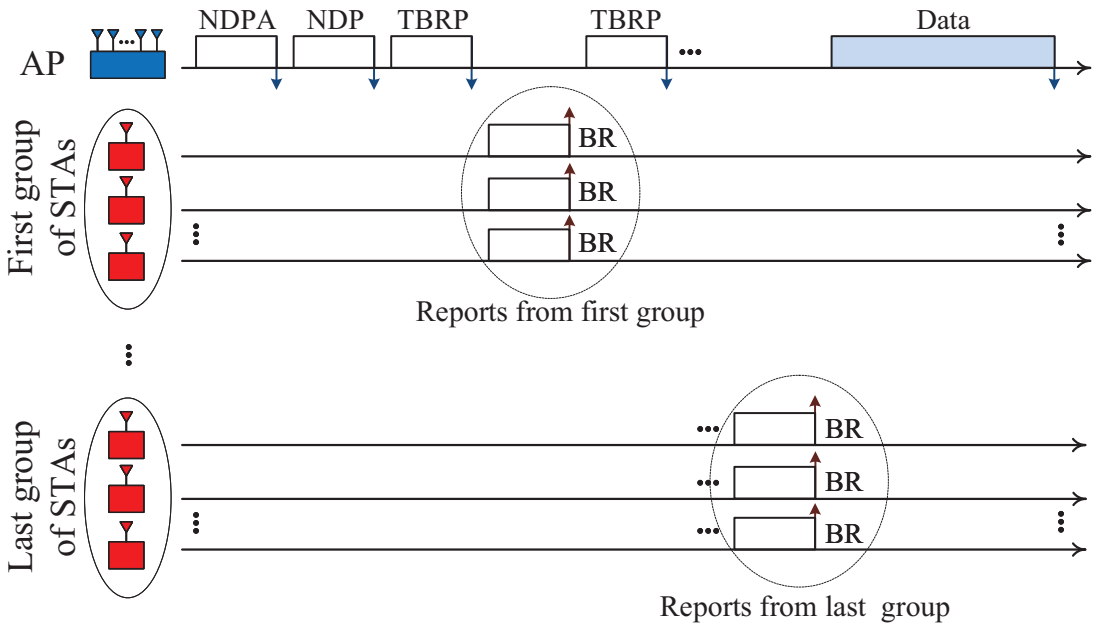


Figure 6.2: Existing channel sounding protocol in IEEE 802.11ax.

over all tones as $\mathbf{\Psi} = \{\psi_{i,k}\}_{\forall i,k}$ and $\mathbf{\Phi} = \{\phi_{i,k}\}_{\forall i,k}$, where i is the angle index ($1 \leq i \leq N_\psi$), k is the tone index ($1 \leq k \leq N_{tone}$), and N_{tone} is the number of tones.

Generally speaking, $\psi_{i,k} \in [0, \pi/2)$ and $\phi_{i,k} \in [0, 2\pi)$. The angles will be quantized before being sent to the AP. In 802.11 standards, two types of quantization are specified for feedback:

- *Feedback type 0* uses 5 bits for each angle in $\mathbf{\Psi}$ and 7 bits for each angle in $\mathbf{\Phi}$.
- *Feedback type 1* uses 7 bits for each angle in $\mathbf{\Psi}$ and 9 bits for each angle in $\mathbf{\Phi}$.

Tone Grouping. As Wi-Fi networks typically work in indoor scenarios for short-range communications, their coherence bandwidth tends to be large. Hence, *tone grouping* has been employed to bond N_g tones. In 802.11ax standard [70], $N_g = \{1, 4, 16\}$. Particularly, $N_g = 1$ means that no grouping is employed. Also, $N_g = 16$ is only allowed with feedback type 1.

Beamforming Report (BR). The BR frames carry the quantized angles ($\mathbf{\Psi}$ and $\mathbf{\Phi}$) from each STA to the AP. These frames are also used to carry the channel strength information (average SNR and SNR deviation for each group of tones) from each STA to the AP. Based on the reported SNR

information, the AP manages available spectral and power resources to serve STAs.

Polling: Polling is a mechanism to coordinate the report process among STAs. Once all STAs have prepared their BR frames, the AP sends Trigger Beamforming Report Poll (TBRP) frames sequentially. Each TBRP frame coordinates a group of STAs to send their BR frames through uplink MU-MIMO as illustrated in Fig. 6.2. The AP decodes the BR frames and identifies the sender of each report using the MAC address in the corresponding frame. After polling all the groups, the AP obtains information required for downlink MU-MIMO transmission.

6.3.2 Design Objectives and Challenges

The objectives of this work are to design and evaluate a practical, yet efficient, downlink MU-MIMO-OFDMA transmission scheme for 802.11ax networks. Towards these objectives, we face the following two challenges.

Challenge 1 – Channel Sounding Overhead. Channel sounding is crucial for beamforming in downlink MU-MIMO transmissions. However, the existing channel sounding protocol in Fig. 6.2 entails a large airtime overhead and significantly compromises the throughput gain of MU-MIMO. For instance, consider an AP with 8 antennas and a single-antenna STA working on 160 MHz bandwidth. Even with the tone grouping, the angles information in a single report could be as large as 7.0 kB[1], which is far beyond a maximum transmission unit (2.3 kB) in WLANs [135]. This means that a BR frame in Fig. 6.2 can take more than 3 packets for CSI feedback. Such a large airtime overhead not only consumes network bandwidth but it also ruins the freshness of CSI for beamforming.

Challenge 2 – Joint Resource Allocation. The marriage of MU-MIMO and OFDMA cre-

[1] In this case, $N_\psi = N_\phi = 7$ and feedback type 1 is used over 498 groups of tones. Representation of angles requires $55,776$ bits ≈ 7.0 kB.

ates a joint resource allocation problem for the AP, which involves RU assignment for users and power allocation for MIMO streams. This problem is complicated as it crosses spectral and power domains. Solving the resource allocation problem is time-constrained as the coherence of wireless channels degrades over time. It is therefore important for an AP to have a low-complexity algorithm that can find an efficient resource allocation solution in real time. A classical approach for solving this problem is to first formulate the problem as an optimization problem and then employ existing optimization solvers to compute the optimal solution. This approach, however, is infeasible in practice due to the high computational complexity from an exhaustive search over RU assignment instances. For example, consider a small 802.11ax network where a 4-antenna AP serves 6 single-antenna STAs over four 52-tone RUs on 20 MHz bandwidth. We formulate the resource allocation problem as an MINLP optimization problem and employ CVX package to solve it for a given RU assignment. Our observation is that it takes up to 342 minutes to find an optimal solution with search over $2^{23.3}$ RU assignment instances. Such a large delay makes resource allocation infeasible for practical use and urges us to devise a low-complexity resource allocation mechanism.

6.4 Overview of DeepMux

In this section, we present an overview of DeepMux, which leverages recent advances in DNNs to address the challenges for downlink MU-MIMO-OFDMA transmissions in 802.11ax networks. Fig. 6.3 shows a high-level structure of DeepMux. It mainly comprises two components: DLCS and DLRA. In what follows, we present the basic idea of these two components.

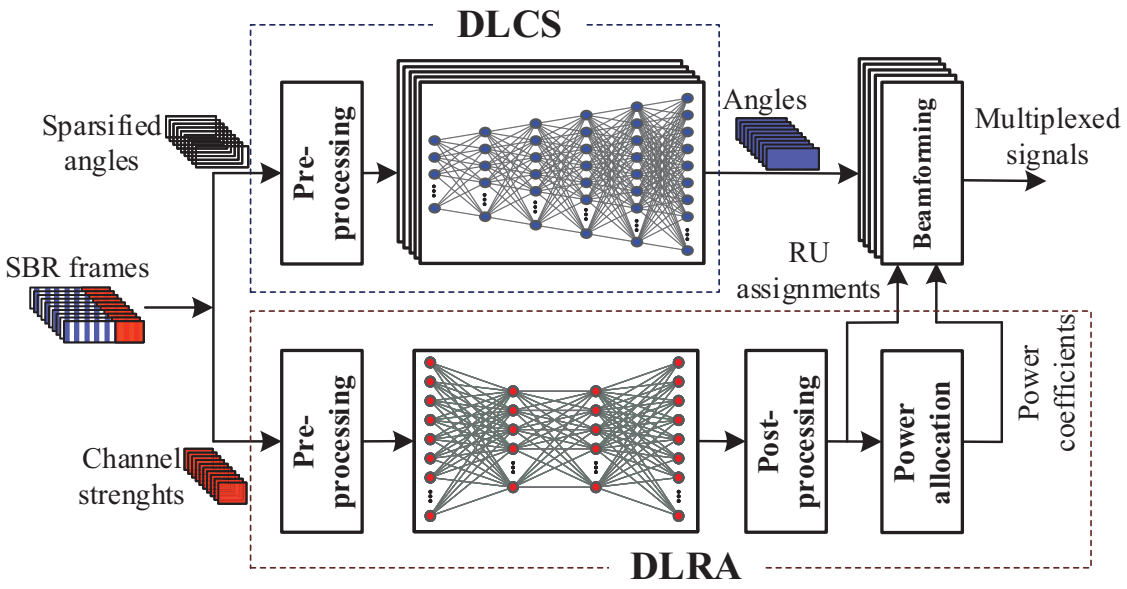


Figure 6.3: The overview of DeepMux.

6.4.1 Basic Idea of DLCS

DLCS is an enhanced 802.11 channel sounding protocol aiming to reduce the sounding overhead. Its design is based on the following two observations: i) wireless channels in local area networks are highly correlated in the frequency domain; and ii) tone grouping in the current 802.11 sounding protocol is not an efficient approach for feedback compression. Motivated by the success of DNNs for image compression, we propose to use DNNs to reduce the channel sounding overhead in the CSI feedback process. Specifically, instead of reporting CSI over a large number of tones, each STA only reports CSI over a small number of tones. Based on the reported CSI over sparse tones, the AP attempts to infer the CSI over all tones using DNNs.

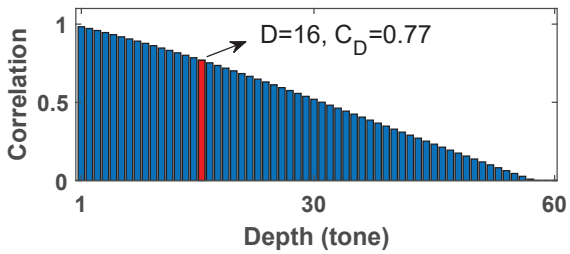
While the idea is straightforward, an important question is how to train the DNNs so that they can infer the full CSI based on the limited feedback. For this question, one solution is that the AP asks every STA to report a large amount of CSI over all tones at the beginning and uses the large amount of CSI to train the DNNs. This solution, however, imposes heavy computational

and communication burdens on STAs, and thus is not amenable to implementation. To circumvent this issue, we use uplink CSI, instead of downlink CSI, for the training of DNNs. This is because uplink and downlink channels have the same profile in the frequency domain, thanks to the channel reciprocity [109]. In other words, uplink and downlink channels bear the same shape over frequency domain even without channel calibration, making it possible for DNNs to learn the downlink frequency-domain CSI correlation using uplink CSI samples in the absence of channel calibration.

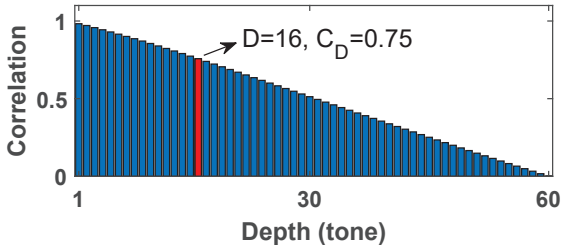
Additionally, an AP can easily obtain uplink CSI over all tones. Obtaining uplink CSI requires no effort from STAs, making the training process transparent to the STAs. Whenever an AP receives packets from STAs, it can measure the uplink channel based on the packets' preamble. We note that, different from prior channel reciprocity applications, channel calibration is not needed for our application. Details of DLCS are presented in Section 6.5.

6.4.2 Basic Idea of DLRA

The marriage of MU-MIMO and OFDMA creates a challenge for an 802.11ax-enabled AP to optimally allocate the available spectral and power resources in a reasonable amount of time. To address this challenge, DeepMux formulates the resource allocation problem as an optimization problem. In its original form, the optimization problem is an MINLP problem, where its binary variables correspond to RU assignment sub-problem and its continuous variables correspond to power allocation sub-problem. DeepMux approaches the MINLP problem by reformulating it into a Mixed-Integer Linear Programming (MILP) problem. Unlike an MINLP problem, an MILP problem can be systematically solved in two steps: i) an organized search mechanism over discrete instances of the feasible region (RU assignment instances), and ii) an interior-point algorithm that solves the convex sub-problem (power allocation) for a given RU assignment.



(a) Correlation of angles in $\mathbf{\Psi}$.



(b) Correlation of angles in $\mathbf{\Phi}$.

Figure 6.4: Spectral correlation of angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$.

Given that MILP is NP-hard in general, we take advantage of recent advances in DNNs to determine the optimal RU-assignment in the first step. Specifically, DeepMux employs a DNN to compute the values for the binary optimization variables in the MILP problem. Such a DNN is trained offline, in a supervised manner, using the SNR reports from STAs, as shown in Fig. 6.3. After the binary variables (corresponding to the RU assignment sub-problem) are determined, the MILP problem degrades to a linear programming problem, which is easy to solve. Details of DLRA are presented in Section 6.6.

6.5 DLCS: A Low-Overhead Channel Sounding

DLCS enhances the 802.11 channel sounding protocol in Fig. 6.2 by reducing the airtime consumed by BR frames. This is done through *sparsification* of $\mathbf{\Psi}$ and $\mathbf{\Phi}$ angles in the frequency domain. That is, each STA reports CSI angles over a few tones, and the AP infers the CSI angles for all tones based on the sparsified feedback using DNNs.

Before diving into DLCS, we first take a look at the frequency-domain correlation of CSI angles. We collected $50,000$ $\mathbf{\Psi}$ and $\mathbf{\Phi}$ samples in an office environment to measure the frequency-domain correlation. For a sequence $\mathbf{x} \in \mathbb{R}^{1\times L}$, we define C_D as its correlation at depth D by

letting:

$$C_D = \mathbb{E}_m \left[\frac{\mathbf{x}_{(m+1:m+D)} \mathbf{x}^T_{(m+D+1:m+2D)}}{\left|\mathbf{x}_{(m+1:m+D)}\right| \left|\mathbf{x}_{(m+D+1:m+2D)}\right|} \right], \tag{6.1}$$

where $\mathbf{x}_{(i:j)} \triangleq \left[x_i, x_{i+1}, \cdots, x_j\right]$ with x_i being the ith element in $\mathbf{x}$, and $(\cdot)^T$ is transpose operator. Fig. 6.4 shows the correlation of the collected CSI angles at different tone depths. It can be seen that, when the tone depth is greater than 16 (i.e., $D > 16$), the correlation is still considerable for both $\mathbf{\Psi}$ and $\mathbf{\Phi}$ angles. This means that, grouping the angles over N_g tones (simply by averaging operation) cannot fully harvest such a significant correlation for compression purpose. On the other hand, tone grouping may lead to an inaccurate feedback when $N_g > 16$. DLCS is a more sophisticated compression approach to reduce the sounding overhead by exploiting inter-tone CSI correlation.

In what follows, we first present the settings of DNNs and then elaborate on their training (exploration) and sparsification (exploitation) phases separately.

6.5.1 DNNs Settings

As shown in Fig. 6.5, DLCS employs DNNs at the AP to infer full CSI angles based on a sparsified feedback. One DNN is used for the angles in $\mathbf{\Psi}$ and another DNN is used for the angles in $\mathbf{\Phi}$. The dimension of input layer is S, corresponding to the quantized CSI angles over S tones. The value of S is selected through experimental studies, which will be shown shortly. The DNNs have N_{tone} neurons on the output layer, corresponding to the inferred CSI angles over all tones (e.g., $N_{\text{tone}} = 234$ for all the nine 26-tone RUs over 20 MHz bandwidth). DNNs have multiple hidden layers, say L hidden layers. The dimension of the ith hidden layer is d_i. Each hidden layer is fully-connected, followed by a batch-normalization layer to speed up the training convergence [72].

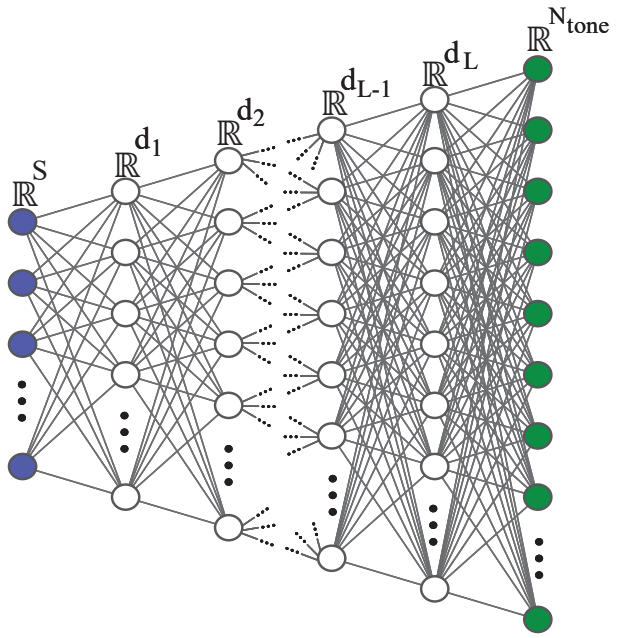


Figure 6.5: DNNs' structure at the AP for inferring $\mathbf{\Psi}$ and $\mathbf{\Phi}$ based on limited feedback.

ReLU activation function is used for each layer. Since the DNNs are designed for interpolation purpose, they are in an enlarging trapezoid shape.

6.5.2 Training Phase

As we explained before, the AP does not require STAs to report a large amount of downlink CSI angles for training DNNs because doing so imposes heavy computational and communication burdens on STAs. Instead, the AP uses its estimated uplink channels to calculate CSI angles and train the DNNs by taking advantage of wireless channel reciprocity. Since the DNNs focus only on learning the frequency-domain properties of CSI, channel calibration is not necessary to compensate the response difference between Tx and Rx RF chains.

Using the uplink CSI to train the DNNs have two benefits. *First*, it is easy for an AP to collect a large amount of samples for training purpose. As long as an STA sends a packet, the AP can estimate the uplink channel and use it for generating angles and training DNNs. Simply put, the

AP requires zero effort to obtain dataset for training DNNs. *Second*, it tends to offer better training results as uplink CSI does not suffer from tone grouping and quantization errors. If the AP wants to use downlink CSI for training DNNs, quantization of the estimated downlink CSI at STAs is needed to facilitate the feedback. This introduces quantization error and degrades the training performance. In contrast, using uplink CSI for training purpose does not suffer from this issue.

In what follows, we describe the operations of DNNs training at the AP. No extra effort is needed at the STAs.

Data Collection. AP and STAs work in their ordinary mode. Whenever the AP receives a packet, it decodes the packet and records its estimated uplink channel on all tones. Then, the AP performs spatial compression on the estimated uplink channel over every tone, as specified in 802.11 standards [70] to collect CSI angles (i.e., $\mathbf{\Psi}$ and $\mathbf{\Phi}$). The generated CSI angles are organized in batches and used for training DNNs.

Data Preprocessing. As shown in Fig. 6.3, each batch of CSI angles are pre-processed before being used for training the DNNs. The pre-process is to make the angles zero-mean and unite-variance over all tones [83]. Albeit simple, this pre-process significantly improves the convergence of DNNs [83], especially when gradient descent algorithms are used for weight adaptation [92]. The AP also quantizes these pre-processed angles with different numbers of bits and keeps all versions to examine their performance.

Training Parameters and Provisions. Normalized Mean Squared Error (NMSE) loss function is employed to measure the sparsification error. The DNNs are trained using Adam optimizer [85]. The training is performed with an initial learning rate of 0.001 and decaying rate of 0.98 following a step-wise approach. The batch size is set to 128. All parameters are initialized using Xavier initialization [50]. Dropout [154] is applied to all hidden layers to prevent over-fitting and improve the generalization of the model. All DNNs are trained end-to-end using Pytorch v1.4 library [132].

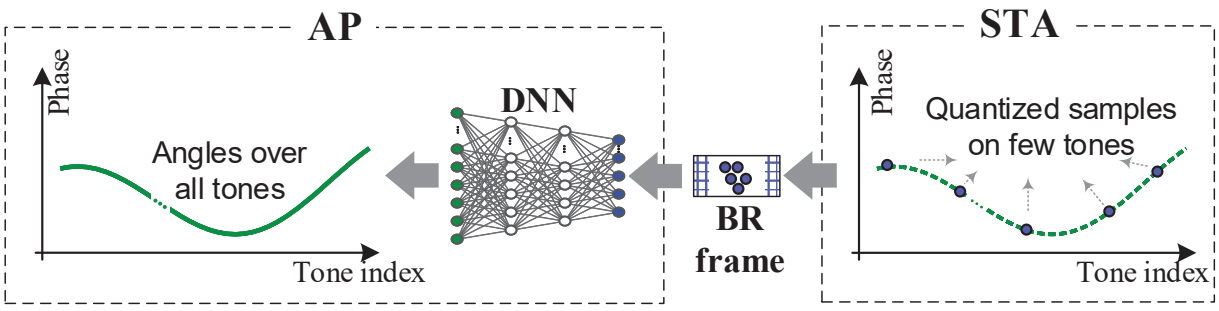


Figure 6.6: DLCS workflow in sparsification (exploitation) phase.

6.5.3 Sparsification Phase

After completing the training phase, the AP initiates the sparsification phase. That is, the network begins to use the trained DNNs to reduce the channel sounding overhead when applicable. To do so, the AP informs all STAs of S_ψ, S_ϕ, q_ψ, and q_ϕ, where S_ψ and S_ϕ are the number of tones for which STAs report angles of $\mathbf{\Psi}$ and $\mathbf{\Phi}$, respectively. q_ψ and q_ϕ are the number of bits for quantizing each angle in $\mathbf{\Psi}$ and $\mathbf{\Phi}$, respectively. Fig. 6.6 illustrates the CSI reporting process when the AP is equipped with the trained DNNs. In what follows, we elaborate the operations at an STA and the AP, respectively.

Operations at an STA. Referring to Fig. 6.2, when MU-MIMO transmission is triggered by an NDPA frame, each STA estimates the downlink channel vector $\mathbf{H}(k)$ based on the received NDP frame, where $k = \{k_\psi, k_\phi\}$ is the selected tone indices, $k_\psi \in \{\lfloor 0.5N_{tone}/S_\psi \rfloor, \lfloor 1.5N_{tone}/S_\psi \rfloor, \cdots, \lfloor (S_\psi - 0.5)N_{tone}/S_\psi \rfloor\}$ is the set of tone indices for which STAs report $\mathbf{\Psi}$ and $k_\phi \in \{\lfloor 0.5N_{tone}/S_\phi \rfloor, \lfloor 1.5N_{tone}/S_\phi \rfloor, \cdots, \lfloor (S_\phi - 0.5)N_{tone}/S_\phi \rfloor\}$ is the set of tone indices for which STAs report $\mathbf{\Phi}$. Spatial compression is performed on $\mathbf{H}(k)$ to obtain the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$, which are then quantized using q_ψ and q_ϕ bits (using the quantization method in [67]), respectively. In the BR frame shown in Fig. 6.2, instead of reporting CSI angles on all groups of tones, the STAs report ψ and ϕ angles only on those S_ψ tones and S_ϕ tones, respectively. In addition,

Table 6.2: End-to-end error of DNNs in inferring the angles in $\mathbf{\Psi}$.

	S_ψ=5	S_ψ=6	S_ψ=7	S_ψ=8	S_ψ=9
q_ψ=3 bits	10.55%	10.63%	12.00%	8.99%	9.88%
q_ψ=4 bits	5.85%	4.95%	5.03%	3.86%	3.29%
q_ψ=5 bits	3.97%	2.77%	2.52%	1.93%	1.32%
q_ψ=6 bits	3.52%	2.16%	1.53%	1.35%	1.14%
q_ψ=7 bits	3.19%	2.08%	1.16%	1.14%	0.80%

Table 6.3: End-to-end error of DNNs in inferring the angles in $\mathbf{\Phi}$.

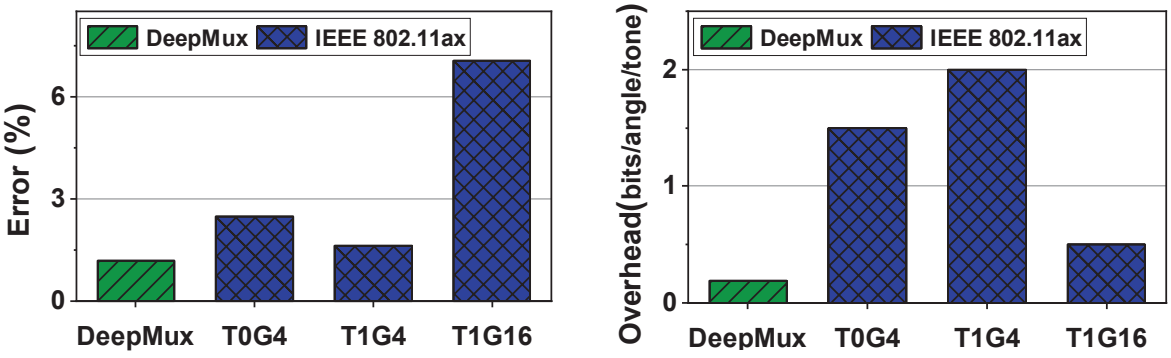
	S_ϕ=5	S_ϕ=6	S_ϕ=7	S_ϕ=8	S_ϕ=9
q_ϕ=3 bits	26.51%	22.70%	27.39%	29.83%	21.57%
q_ϕ=4 bits	8.30%	6.63%	6.33%	6.09%	5.73%
q_ϕ=5 bits	3.01%	2.40%	2.19%	2.14%	1.85%
q_ϕ=6 bits	2.67%	2.06%	1.10%	1.01%	0.76%
q_ϕ=7 bits	2.30%	1.07%	0.82%	0.77%	0.57%

each STA also reports the measured SNR values to the AP in the BR frame, following the existing 802.11 protocol [67].

Operations at the AP. Upon receiving the reports from an STA, the AP extracts the quantized angles and SNR reports. As illustrated in Fig. 6.6, the received angles are then fed into the DNNs to infer the angles over all tones. The output of the DNNs are then used to construct the beamforming vectors for downlink MU-MIMO transmissions.

6.5.4 Parameter Selection and Numerical Results

A question to ask is how to choose the values for sparsification parameters S_ψ, S_ϕ, q_ψ, and q_ϕ. In our design, the parameter values are selected to ensure the end-to-end errors below a pre-defined threshold, which is empirically set. Specifically, after the AP collects the sufficient channel data, it first trains DNNs under different values of sparsification parameters and then records the end-to-end error in the test phase. The AP selects the values for sparsification parameters that yield the



(a) Error.

(b) Overhead.

Figure 6.7: Error and overhead comparison between DLCS and existing 802.11 protocols [70].

lowest sounding overhead while meeting the end-to-end error requirement (below a pre-defined threshold).

To illustrate this selection approach, we resort to experiments. We implemented DLCS in an indoor environment and collected about $50,000$ angle samples in the uplink over 20 MHz bandwidth. We tuned those parameters and examined the performance of well-trained DNNs. As a possible end-to-end error threshold in inferring the angles, we use error from the tone grouping mechanism. Table 6.2 and Table 6.3 present our results. In each table, the DNN settings which meet the end-to-end error requirement are highlighted in green color. Based on the results, we choose $(S_\psi = 9, q_\psi = 5)$ which leads to 0.19 bits/angle/tone overhead and 1.32% error for the angles in $\mathbf{\Psi}$. We choose $(S_\phi = 6, q_\phi = 7)$ for the angles in $\mathbf{\Phi}$ which leads to 0.18 bits/angle/tone overhead and 1.07% error. Finally, the DNNs we choose are a $9 \times 16 \times 32 \times 64 \times 128 \times 234$ DNN for sparsification of $\mathbf{\Psi}$ and a $6 \times 16 \times 32 \times 64 \times 128 \times 234$ DNN for sparsification of $\mathbf{\Phi}$. We note that the resultant parameter values are scenario-specific. When an AP is moved to a new scenario, it needs to re-tune the parameters to obtain the "best" values for those parameters. Fortunately, the parameter re-tuning process can be done by the AP automatically without human intervention.

We now compare DLCS with existing 802.11 protocols in terms of error and sounding over-

head. Fig. 6.7 presents our results. Particularly, TiGj in the figure means feedback type i is employed and $N_g = j$ tones are grouped for feedback. Fig. 6.7(a) shows the superior performance of DLCS in terms of error. DLCS reaches 1.19% error, while T0G4, T1G4, and T1G16 reach 2.48%, 1.64%, and 7.05% error, respectively. Fig. 6.7(b) shows that DLCS entails significantly lower overhead compared to existing 802.11 protocols. DLCS reaches a sounding overhead as low as 0.19 bits/angles/tone while T0G4, T1G4, and T1G16 reach 1.50, 2.00, and 0.50 bits/angles/tone overhead, respectively. This means DLCS reduces sounding overhead by $62.0\%\sim90.5\%$.

6.6 DLRA: A Lightweight Resource Allocation

In this section, we employ DNNs to facilitate the resource allocation problem at the AP, which includes two sub-problems: RU assignment and power allocation. Recall that the AP recovers angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ using DNNs, and it also collects SNR values over all tones. The angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ can be used to partially reconstruct the right singular vectors of channel matrices, which can be leveraged to mitigate inter-user interference in the downlink transmissions. The SNR values provide the information of channel quality, which can be used to optimize the resource allocation. In what follows, we first formulate the resource allocation problem as an optimization problem, and then develop a learning-based algorithm to solve it. Finally, we offer numerical results to show the effectiveness of the proposed learning-based algorithm.

6.6.1 Problem Formulation and Reformulation

Problem Formulation. At an AP, denote $\mathcal{N}$ as the set of STAs that it serves in the downlink MU-MIMO-OFDMA transmission. Denote $\mathcal{R}$ as the set of RUs, which are the granularity for assignment. Let $|\mathcal{N}| = N_{\mathrm{sta}}$ and $|\mathcal{R}| = N_{\mathrm{ru}}$. We define a binary variable $z_{i,j}$ to indicate the RU

assignment. Specifically, $z_{i,j} = 1$ if RU j is assigned to STA i; and $z_{i,j} = 0$ otherwise. Denote $p_{i,j}$ as the portion of the AP's power allocated to STA i on RU j. Denote W_j as the bandwidth of RU j. Denote $\gamma_{i,j}$ as reported SNR at STA i on RU j. Denote $r_{i,j}$ as the data rate achieved by STA i on RU j. Denote r_i as the achievable data rate for STA i. Denote $\Omega(\cdot)$ as the mapping function from SNR to data rate.

Then, the resource allocation problem with the objective of maximizing total STAs' data rate can be expressed as:

$$\underset{\underline{p},\underline{z}}{\text{maximize}} \sum_{i\in\mathcal{N}} r_i \tag{6.2a}$$

$$\text{s.t.} \quad r_i \le \sum_{j\in\mathcal{R}} r_{i,j}, \quad i\in\mathcal{N}; \tag{6.2b}$$

$$r_{i,j} \le W_j z_{i,j} \Omega\left(p_{i,j}\gamma_{i,j}\right), \quad i\in\mathcal{N}, j\in\mathcal{R}; \tag{6.2c}$$

$$\sum_{i\in\mathcal{N}} z_{i,j} \le N_{\text{ap}}, \quad j\in\mathcal{R}; \tag{6.2d}$$

$$\sum_{i\in\mathcal{N},j\in\mathcal{R}} p_{i,j} \le 1\,. \tag{6.2e}$$

In this formulation, $\underline{z} = \{z_{i,j}\}_{i\in\mathcal{N},j\in\mathcal{R}}$ and $\underline{p} = \{p_{i,j}\}_{i\in\mathcal{N},j\in\mathcal{R}}$ are optimization variables. $\{\gamma_{i,j}\}_{i\in\mathcal{N},j\in\mathcal{R}}$, $\{W_j\}_{j\in\mathcal{R}}$, and N_{ap} are given parameters. Constraint (7.4b) calculates the achieved data rate by an STA. Constraint (7.4c) defines the achievable rate region. Constraint (6.2d) is spatial DoF constraints on the maximum number of STAs that can be allocated to an RU. Constraint (6.2e) characterizes the power budget at the AP.

Achievable Rate Region. A classical way to map SNR to data rate is Shannon capacity, which is a theoretical bound and hard to reach in practice. In 802.11 networks, adaptive MCS is used

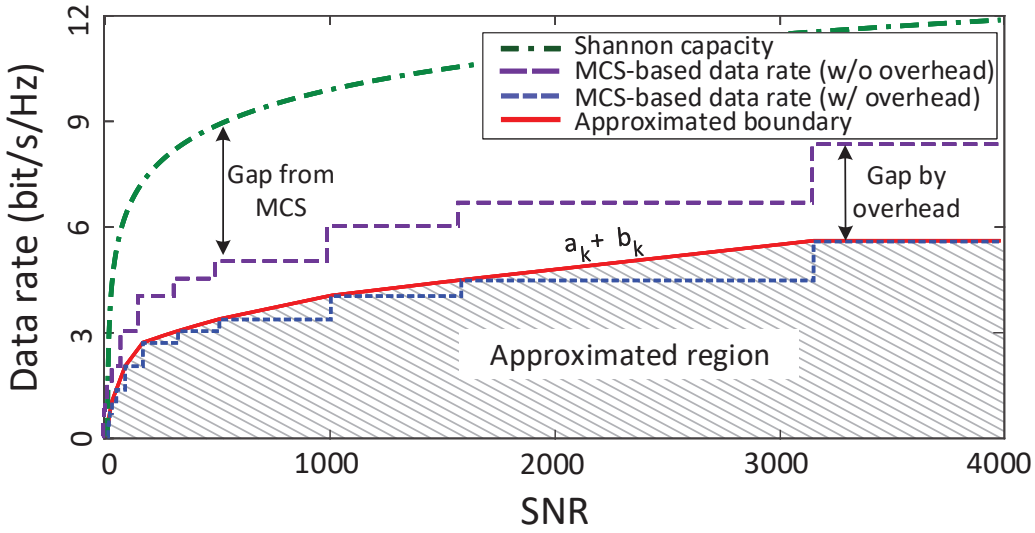


Figure 6.8: Illustration of Shannon capacity, MCS-based data rate, and achievable data rate.

to adjust the data rate based on SNR. As shown in Fig. 6.8, there is a significant gap between Shannon capacity and MCS-based data rate. Therefore, Shannon capacity is not an ideal function for our purpose. Moreover, when taking into account the overhead from OFDM cyclic prefix and pilot tones in 802.11ax[2], the achievable data rate becomes even lower, as shown in Fig. 6.8. The achievable data rate region (MCS-based rate with overhead) is characterized by a staircase curve, which is non-convex function. To simplify the optimization, we approximate the achievable rate region using a series of linear constraints as illustrated by Fig. 6.8.

Mathematically, by defining γ as a measured SNR value, we approximate the achievable rate region as follows:

$$\Omega(\gamma) \leq a_k \gamma + b_k; \qquad k \in \mathcal{K}, \tag{6.3}$$

where a_k and b_k are given in Table 6.4 as per IEEE 802.11ax; and $\mathcal{K} \triangleq \{1, 2, \cdots, 13\}$. We note that the EVM in Table 6.4 is equivalent to the inverse of post-SNR of a decoded data stream

[2]For 802.11ax with 20 MHz bandwidth, every 26-tone RU has 2 tones for pilot.

Table 6.4: EVM specified in IEEE 802.11ax [70].

EVM (dB)	$[+\infty,-5)$	$[-5,-8)$	$[-8,-10)$	$[-10,-13)$	$[-13,-16)$	$[-16,-19)$	$[-19,-22)$
Modulation	N/A	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM
Coding rate	N/A	$1/2$	$3/4$	$1/2$	$3/4$	$1/2$	$3/4$
Γ(EVM)	N/A	$1/2$	$3/4$	1	$3/2$	2	3
a_i	0.1067	0.0536	0.0457	0.0339	0.0170	0.0170	0.0085
b_i	0	0.1679	0.2177	0.3359	0.6734	0.6718	1.3468
EVM (dB)	$[-22,-25)$	$[-25,-27)$	$[-27,-30)$	$[-30,-32)$	$[-32,-35)$	$[-35,-\infty)$	$[-35,-\infty)$
Modulation	64QAM	64QAM	64QAM	256QAM	256QAM	1024QAM	1024QAM
Coding rate	$2/3$	$3/4$	$5/6$	$3/4$	$5/6$	$3/4$	$5/6$
Γ(EVM)	4	$9/2$	5	6	$20/3$	$15/2$	$25/3$
a_i	0.0021	0.0018	0.0013	0.0008	0.0007	N/A	0
b_i	2.3609	2.4605	2.6968	3.2806	3.3696	N/A	5.6250

at a receiver. The relation of γ in (6.3) and the EVM value in Table 6.4 can be expressed as $\gamma = 10^{-\mathrm{EVM}/10}$.

Based on the EVM regions specified in Table 6.4, the approximated achievable rate region with its boundaries is shown in Fig. 6.8. Then, constraints in (7.4c) can be expressed as:

$$r_{i,j} \leq W_j z_{i,j}(a_k p_{i,j}\gamma_{i,j} + b_k), \quad i \in \mathcal{N}, j \in \mathcal{R}, k \in \mathcal{K}. \tag{6.4}$$

Using (6.4), the resource allocation problem in (7.4) can be re-defined as:

$$\underset{\underline{p},\underline{z}}{\text{maximize}} \sum_{i\in\mathcal{N}} r_i \tag{6.5}$$

$$\text{s.t.} \quad \text{(7.4b)}, \ \text{(6.2d)}, \ \text{(6.2e)}, \ \text{and (6.4)}.$$

The optimization problem in (6.5) is an MINLP problem. The non-linear term is from (6.4), where binary and continuous optimization variables are multiplied.

Problem Reformulation. To reduce the processing time, we reformulate the MINLP problem (6.5) to an MILP problem by leveraging a classic linearization technique [151]. To do so, we assume that the SNR value is bounded. This is a valid assumption in practice. Denote γ_{max} as the

maximum value of SNR (e.g., 45 dB in our design) and define a constant $A = \max_{j,k}\{W_j(a_k\gamma_{max}+b_k)\}$. Then, (6.4) can be equivalently expressed as:

$$r_{i,j} \leq W_j(a_k p_{i,j}\gamma_{i,j} + b_k), \quad i \in \mathcal{N}, j \in \mathcal{R}, k \in \mathcal{K}. \tag{6.6a}$$

$$0 \leq r_{i,j} \leq z_{i,j}A, \quad i \in \mathcal{N}, j \in \mathcal{R}. \tag{6.6b}$$

Therefore, the MINLP problem in (6.5) can be reformulated to the following MILP problem:

$$\underset{\underline{p},\underline{z}}{\text{maximize}} \sum_{i\in\mathcal{N}} r_i \tag{6.7}$$

$$\text{s.t.} \quad \text{(7.4b)}, \ \text{(6.2d)}, \ \text{(6.2e)}, \ \text{and (6.6)}.$$

We note that the MINLP problem in (6.5) and the MILP problem in (6.6) have identical feasible region. The reformulation does not alter the solution space. The new optimization problem involves $2N_{\text{sta}}N_{\text{ru}}+N_{\text{sta}}$ continuous variables, $N_{\text{sta}}N_{\text{ru}}$ binary variables, and $14N_{\text{sta}}N_{\text{ru}}+N_{\text{sta}}+N_{\text{ru}}+1$ constraints. Recall the example in Section 6.3.2, where a 4-antenna AP serves six STAs on four 52-tone RUs. By formulating the resource allocation problem in the form of (6.7), off-the-shelf optimization solver MOSEK [14] can find an optimal solution within 5 seconds for most cases. In general, MILP is NP-hard. Its computational complexity is still beyond the acceptable range of a wireless AP device.

6.6.2 DLRA: A Deep-Learning-Based Resource Allocation

Solving an MILP problem is still beyond the computational capacity of an 802.11ax-enabled AP to allocate its resources for downlink transmissions. To reduce the computational complexity, we take advantage of recent advances in DNNs. Specifically, we first reformulate the resource

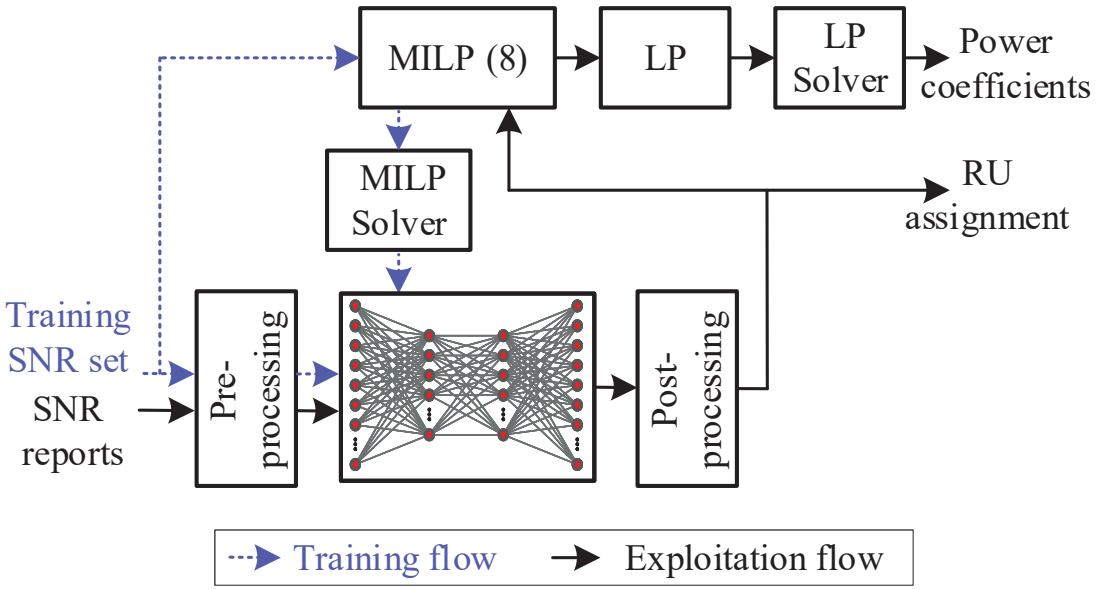


Figure 6.9: DLRA workflow in training and exploitation phases.

allocation problem as an MILP problem as shown in (6.7), and then employ a DNN to compute the binary variables. Once the binary variables are determined, the MILP problem degrades to a linear programming problem, which is easy to solve. In what follows, we focus on the design of a DNN to determine the binary variables in (6.7).

DNN Settings. Fig. 6.9 shows the DNN-based approach in training and sparsification (exploitation) phases. The input of the DNN is the SNR values reported by the STAs. The dimension of input layer is $N_{\mathrm{sta}}N_{\mathrm{ru}}$. The DNN consists of multiple hidden layers. Each hidden layer is fully-connected, followed by a batch-normalization layer to speed up the training convergence [72]. Sigmoid activation function is used for each layer. The output layer has $N_{\mathrm{sta}}N_{\mathrm{ru}}$ neurons, each of which corresponds to a binary variable in RU assignment sub-problem. In our experiments, we consider the case where an 8-antenna AP serves 20 STAs on 9 RUs. For this case, the input and output layers both have 180 neurons, and the overall DNN's structure we trained for RU assignment is $180 \times 128 \times 128 \times 180$.

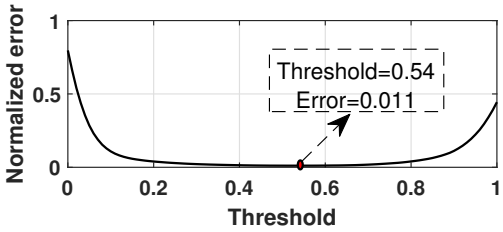
Data Collection and Pre-processing. We collect $60,000$ SNR reports from an office envi-

ronment. Each report consists SNR values over all the nine 26-tone RUs on 20 MHz bandwidth. Every set of SNR values (20 SNR reports) will be flattened, normalized, and then used for training the DNN as an instance of its input. At the same time, the set of unprocessed SNR values will be fed into (6.7). The output of (6.7) includes RU assignment and power allocation coefficients. The resultant RU assignment will be used as the reference output of the DNN in its supervised training procedure. We use MOSEk v.9 [14] to solve (6.7) for a given set of SNR values. Since data generation process is pretty slow, we augment the training data set by adding negligible noise to the original input samples. Moreover, we set aside one third of input-output sample pairs for test purpose. We augment the remaining samples 4.5 times.

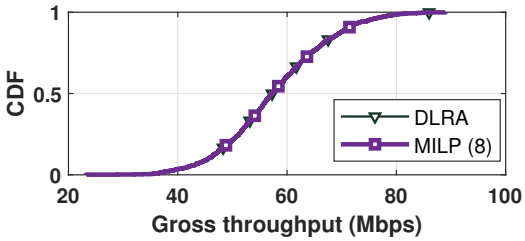
Training Process. To train the DNN, we use NMSE loss function. The outputs of (6.7) for given sets of SNR values are used as reference outputs of the DNN in training loss calculation. For training the DNN, we use Adam optimizer [85] and PyTorch v1.4 library [132]. We also apply batch normalization [72] and Xavier initialization [50] approaches to accelerate the training process.

Post-Processing. The output of DNN will be post-processed in two steps: *binarization* and *correction*. The output of DNN is a vector comprising real values bounded between 0 and 1. We apply a threshold-based *binarization* on outputs of the DNN to transform them into binary entities. Once the binary vector is obtained, we can use our *domain knowledge* to further polish this vector. Two rules are followed in the *correction* step: i) If the DoF constraint is violated on an RU, the STA with the lowest SNR will be removed until the DoF constraint is met. ii) When the DoFs on an RU are under-utilized, the STA with the highest SNR will be activated if there is an assigned STA with a lower SNR.

Computational Complexity. Referring to Fig. 6.9, the computational complexity of pre-processing and post-processing operations is $\mathcal{O}\left(N_{\mathrm{sta}}\right)$, provided that $N_{\mathrm{ru}}<N_{\mathrm{sta}}$. For the trained



(a) Binarization threshold versus normalized error.



(b) Performance gap between DLRA and the optimum to (6.7).

Figure 6.10: Illustrating the performance of DLRA when compared to an optimal solution.

DNN, assuming that the size of hidden layer is proportional to the size of input, its computational complexity is $\mathcal{O}\left(N_{\text{sta}}^2\right)$. For a given RU assignment, MILP in (6.5) degrades to an LP problem. The computational complexity of solving the LP problem is $\mathcal{O}\left(N_{\text{sta}}^{2.5}\right)$. Therefore, the overall complexity of DLRA is $\mathcal{O}\left(N_{\text{sta}}^{2.5}\right)$.

Numerical Results. After the DNN is trained, we use a set of data samples to test its performance. We examine the accuracy of DNN output when different thresholds are used for the binarization post-processing. Fig. 6.10(a) shows the results. It can be seen, DLRA reaches 98.9% accuracy when using 0.54 as the binarization threshold. This means that DLRA offers a very accurate RU assignment. We measured the performance gap between two cases, where the AP uses DLRA and where the AP uses MILP problem for resource allocation. As shown in Fig. 6.10(b), the results confirm that the DLRA almost reaches the optimal performance.

6.7 Experimental Evaluation

In this section, we evaluate the performance of DeepMux by comparing it with existing 802.11ax protocols.

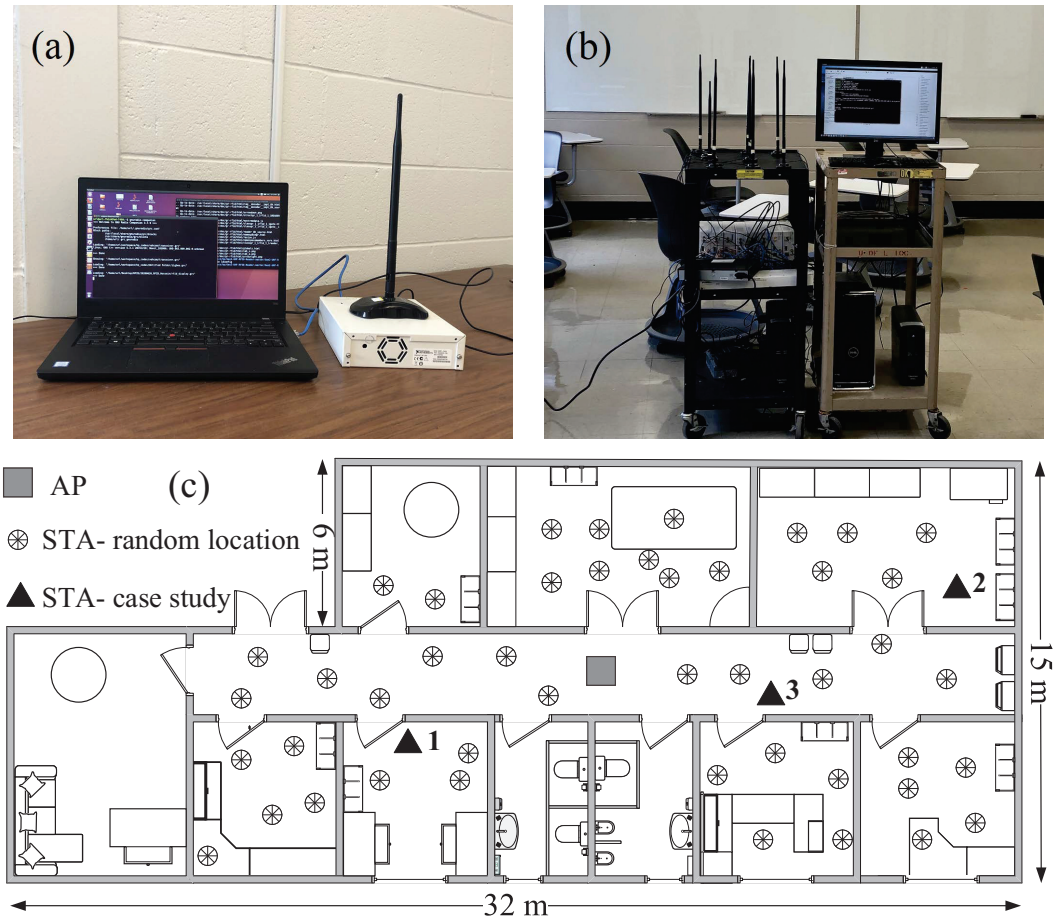


Figure 6.11: Illustrating our wireless testbed and test environment. (a) Prototyped STA. (b) Prototyped AP. (c) Floor plan of tests.

6.7.1 Experimental Settings

Wireless Testbed and Experimental Setting. Fig. 6.11(a) and Fig. 6.11(b) show the wireless testbed that we use to evaluate DeepMux. The testbed has one AP and four STAs which are built using USRP N210 devices and general computers. The AP is equipped with 8 antennas while each STA is equipped with one antenna. As shown in Fig. 6.11(c), the AP is placed at a fixed location, while the four STAs have many random locations to be placed.

Implementation of 802.11ax. The 802.11ax protocol in Fig. 6.2 is implemented on the testbed. The carrier frequency is set to 2.484 GHz, and the bandwidth is set to 20 MHz. Due to the hard-

ware limitation, the inter-frame spacing is equal to one second. A frame has 256 tones in its OFDM modulation, with 18 pilot tones, 216 payload tones, and 22 unused tones. The 26-tone RU configuration (see Fig. 6.1) is used in our study. The transmission power of the AP and STAs is set to 15 dBm. The signal processing modules at both AP and STAs are implemented using C++ in GNURadio-Companion. LDPC channel encoding and decoding are not implemented to reduce the implementation complexity.

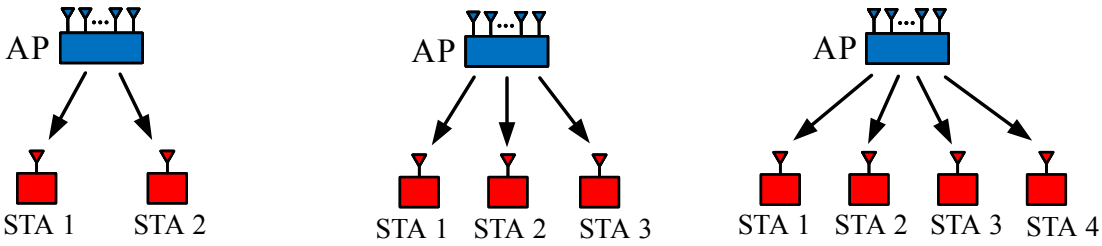
Implementation of DeepMux. DeepMux is implemented on top of the 802.11ax protocol, and its DNNs are trained at the AP using Pytorch v1.4 library [132]. To train DNNs, our data collection campaign lasted three days. During the campaign, low and moderate human activities (i.e., $0\sim5$ persons with brisk walking speed) were observed in the environment shown in Fig. 6.11(c). In this campaign, $100,000$ angles ($50,000$ vectors in $\mathbf{\Psi}$ and $50,000$ vectors in $\mathbf{\Phi}$) on 234 tones were collected for DLCS to train its two DNNs. Meanwhile, $60,000$ SNR reports were collected from the BR frames for DLRA to train its DNN.

6.7.2 Performance Metrics

EVM. EVM is widely used to measure the quality of received signal. Mathematically, EVM is defined as: $\mathrm{EVM} = 10\log_{10}\left(\frac{\mathbb{E}[|\hat{X}-X|^2]}{\mathbb{E}[|X|^2]}\right)$, where X and $\hat{X}$ are original and estimated signals, respectively.

Gross Throughput. Gross throughput is the over-the-air data rate achieved by an STA or the AP. It can be inferred based on the measured EVM by $r = \frac{N_{\mathrm{p}}}{N_{\mathrm{fft}}+N_{\mathrm{cp}}} \cdot b \cdot \Gamma\left(\mathrm{EVM}\right)$, where r is the gross throughput, N_{p} is the number of payload tones, N_{fft} is FFT points, N_{cp} is the length of cyclic prefix, b is the sampling rate, and $\Gamma(\mathrm{EVM})$ is the average number of bits carried by one tone, as specified in Table 6.4. $\Gamma(\mathrm{EVM})$ is determined by modulation order and (LDPC) coding rate.

Net Throughput. Net throughput calculates the data rate while taking into account chan-



(a) Two-user case. (b) Three-user case. (c) Four-user case.

Figure 6.12: Test scenarios used for evaluation of DLCS.

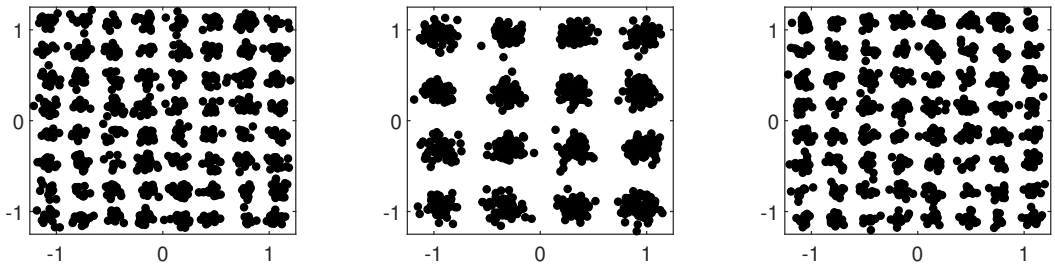
nel sounding airtime overhead. It can be expressed as: $r_{\text{net}} = \frac{t_{payload}}{t_{payload}+t_{\text{overhead}}} \cdot r$, where $t_{payload}$ and t_{overhead} are the time duration of data transmission and channel sounding, respectively. t_{overhead} is determined by the airtime used for transmitting BR, NDPA, NDP, and TBRP frames. For simplicity, we do not consider inter-frame space, re-transmission, and frame aggregation in our calculations.

Comparison Baselines. For DLCS, we compare it with the tone grouping approaches specified in 802.11ax. For notational simplicity, we use TiGj to denote the 802.11 channel sounding protocol with feedback type $i \in \{0, 1\}$ and $j \in \{4, 16\}$ tones in each group. For DLRA, there is not a standardized baseline for comparison. Hence, we implement the best resource allocation effort onto IEEE 802.11ax. The best effort is full utilization of available DoFs on each RU.

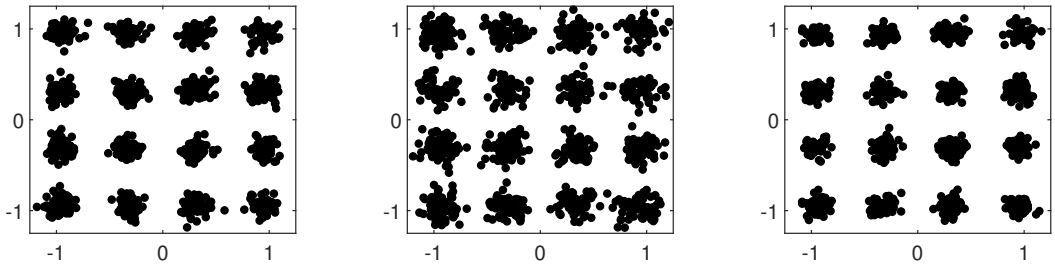
6.7.3 A Case Study for DLCS

We consider the case as shown in Fig. 6.12(b), where the AP serves three STAs. The AP is placed at the square mark in Fig. 6.11(c), and the three STAs are placed at the triangle marks in the figure. Every RU serves these three STAs with equal power allocation, and no resource allocation is involved in this study. In what follows, we present our results.

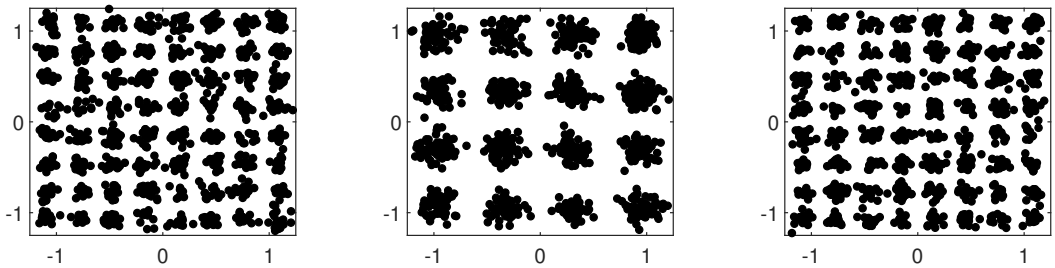
Constellation. We perform downlink MU-MIMO transmissions using both 802.11ax and DLCS channel sounding protocols and collect the decoded signals at the three STAs. Fig. 6.13



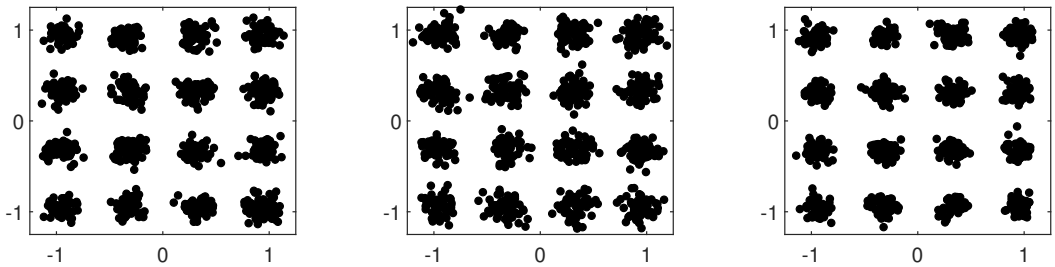
(a) DeepMux: Constellation of decoded signals at the three STAs.



(b) 802.11-T0G4: Constellation of decoded signals at the three STAs.



(c) 802.11-T1G4: Constellation of decoded signals at the three STAs.



(d) 802.11-T1G16: Constellation of decoded signals at the three STAs.

Figure 6.13: Constellations of decoded signals at STA 1 (left), STA 2 (middle), and STA 3 (right), when the WLAN uses different feedback protocols.

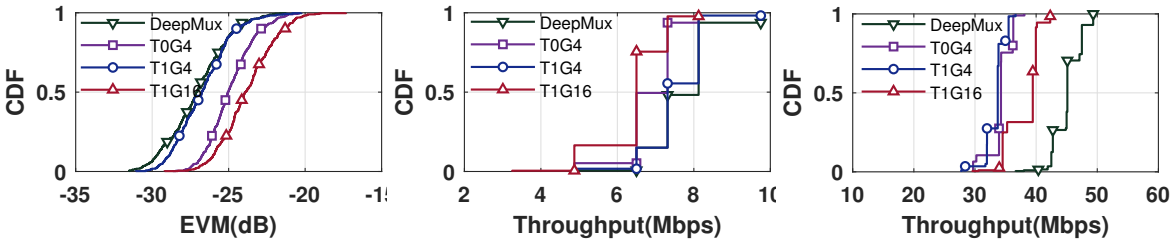
Table 6.5: A case study for comparing DLCS of DeepMux with 802.11 protocols.

		STA 1	STA 2	STA 3	AP
DeepMux	EVM (dB)	-23.5	-19.1	-24.0	–
	Per RU Gross throughput (Mbps)	6.5	4.9	6.5	17.9
	Net throughput (Mbps)	19.1	14.3	19.1	52.5
T0G4	EVM (dB)	-20.1	-17.6	-21.3	–
	Per RU Gross throughput (Mbps)	4.9	3.2	4.9	13.0
	Net throughput (Mbps)	13.7	9.1	13.7	36.5
T1G4	EVM (dB)	-23.0	-17.9	-23.6	–
	Per RU Gross throughput (Mbps)	4.9	3.2	4.9	13.0
	Net throughput (Mbps)	14.7	7.4	14.7	36.8
T1G16	EVM (dB)	-19.8	-18.2	-20.7	–
	Per RU Gross throughput (Mbps)	6.5	3.2	6.5	16.2
	Net throughput (Mbps)	15.6	10.4	15.6	41.6

shows the constellations of decoded signals at the three STAs. The EVMs of the decoded signals are presented in Table 6.5. It can be seen from the measured EVMs that DeepMux offers the best signal quality in the downlink transmissions. This is because the DNNs at the AP can accurately recover CSI over all tones based on the limited CSI feedback. It also can be seen from Fig. 6.13 that DeepMux and 802.11-T1G4 achieve similar signal quality (constellation) in the downlink. This is because we used 802.11-T1G4 as the performance benchmark to select the DNN parameters for DLCS in our experiments.

Feedback Overhead. DeepMux entails 0.6 kbit overhead for CSI feedback from each STA. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 entails 4.9 kbit, 6.5 kbit, and 1.6 kbit overhead for CSI feedback, respectively.

EVM, Gross Throughput, and Net Throughput. Table 6.5 presents our experimental results. We have the following observations. First, in terms of EVM and gross throughput, DLCS is slightly better than 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16. Second, in terms of net throughput, DLCS is significantly superior to 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16. This is not surprising because DLCS consumes much lower airtime for CSI feedback compared to 802.11 channel sounding protocols.



(a) EVM. (b) Gross throughput. (c) Net throughput.

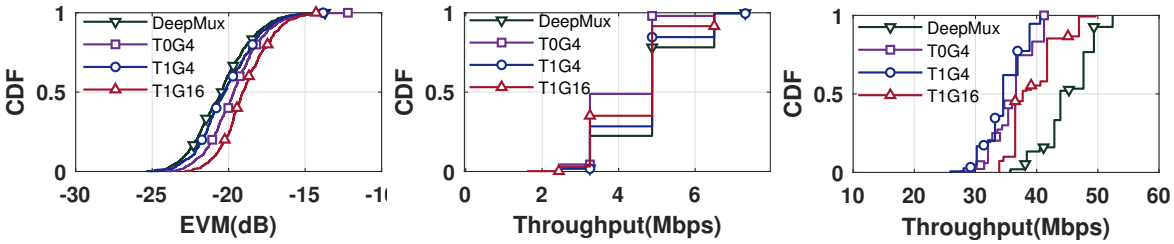
Figure 6.14: Comparison of DeepMux and 802.11 protocols in two-user MIMO downlink transmission.

6.7.4 Extensive Results of DLCS

We extend the case study to extensive experimental trials to thoroughly examine the performance of DLCS. We consider three cases: two-user, three-user, and four-user MIMO as shown in Fig. 6.12. The AP serves these two/three/four users exclusively on all RUs, with equal power allocation. Each STA is placed at a randomly selected spot marked with a filled circle in Fig. 6.11(c).

Two-User Case. Fig. 6.14 presents the comparison results of DeepMux and 802.11 protocols in terms of EVM, gross throughput, and net throughput. Per Fig. 6.14(a), DeepMux achieves -27.1 dB EVM on average, while 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 reach -24.7 dB, -26.7 dB, and -23.8 dB EVM, respectively. Per Fig. 6.14(b), DeepMux slightly outperforms 802.11 protocols in terms of gross throughput. DeepMux achieves 7.7 Mbps gross throughput per RU on average. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 6.8 Mbps, 7.5 Mbps, and 6.4 Mbps gross throughput per 26-tone RU, respectively.

Net throughput reflects the advantage of DLCS as it takes into account airtime overhead in the calculation of throughput. As shown in Fig. 6.14(c), DeepMux obtains 45.2 Mbps net throughput on all RUs on average. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 34.2 Mbps, 33.5 Mbps, and 38.2 Mbps net throughput, respectively. DeepMux offers 31.6%, 34.3%, and 17.8% net throughput gains compared to 802.11-T0G4, 802.11-T1G4, and 802.11-



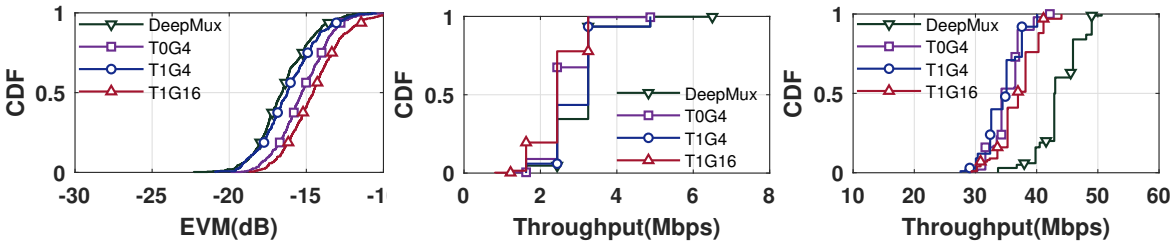
(a) EVM. (b) Gross throughput. (c) Net throughput.

Figure 6.15: Comparison of DeepMux and 802.11 protocols in three-user MIMO downlink transmission.

T1G16, respectively.

Three-User Case. The observations in three-user case are consistent with those in two-user case. Fig. 6.15 shows the experimental results. DeepMux slightly outperforms 802.11 protocols in terms of EVM and gross throughput. Per Fig. 6.15(a), DeepMux achieves -20.4 dB EVM on average, while 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve -19.6 dB, -20.1 dB, and -18.9 dB EVM, respectively. Per Fig. 6.15(b), DeepMux achieves 4.9 Mbps gross throughput on average per RU, while 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 4.1 Mbps, 4.6 Mbps, and 4.4 Mbps respectively. DeepMux offers a significant gain of net throughput over 802.11 protocols. Per Fig. 6.15(c), DeepMux obtains 45.2 Mbps net throughput on average. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 36.1 Mbps, 34.8 Mbps, and 39.4 Mbps net throughput, respectively. This indicates that DeepMux offers 25.2%, 30.0%, and 14.7% gains compared to 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16, respectively.

Four-User Case. The observations in this case are consistent with those in previous two cases. Fig. 6.16 presents the experimental results. In the end, DeepMux achieves 43.7 Mbps net throughput on average. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 35.2 Mbps, 34.6 Mbps, and 37.0 Mbps net throughput, respectively. Numerically, DeepMux offers 24.1%, 26.3%, and 18.1% net throughput gains compared to 802.11-T0G4, 802.11-T1G4, and 802.11-



(a) EVM. (b) Gross throughput. (c) Net throughput.

Figure 6.16: Comparison of DeepMux and 802.11 protocols in four-user MIMO downlink transmission.

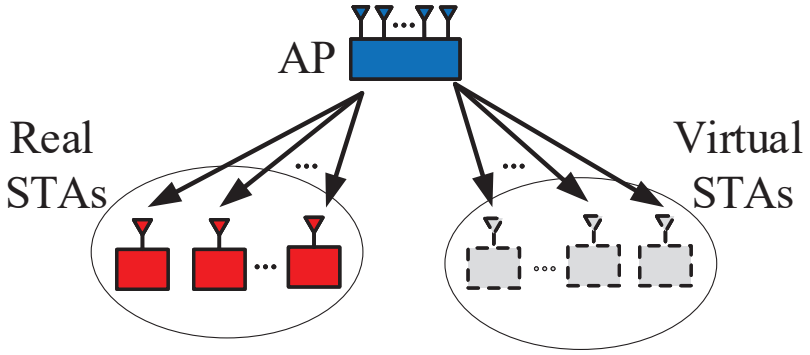
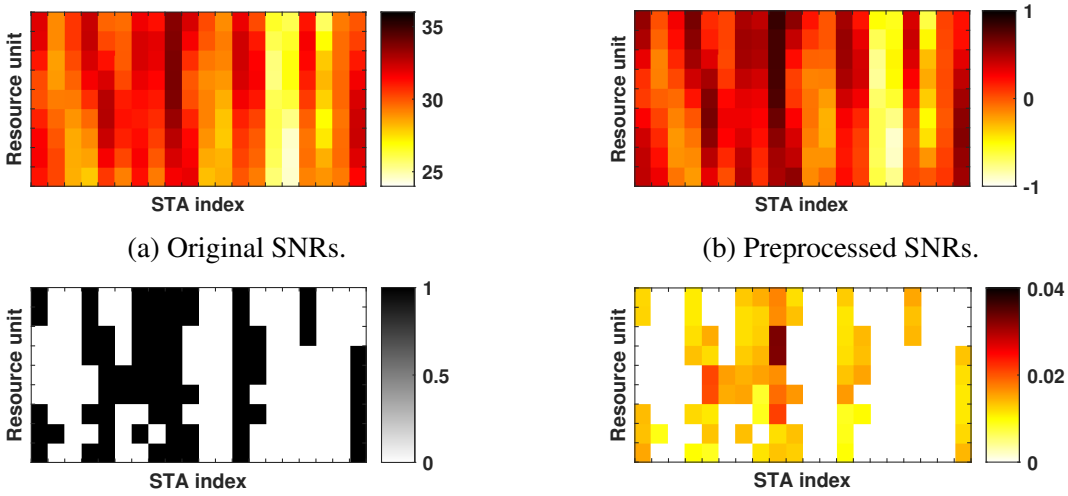


Figure 6.17: Test scenario for evaluating DeepMux in MU-MIMO-OFDMA transmissions.

T1G16, respectively.

6.7.5 Overall Performance of DeepMux

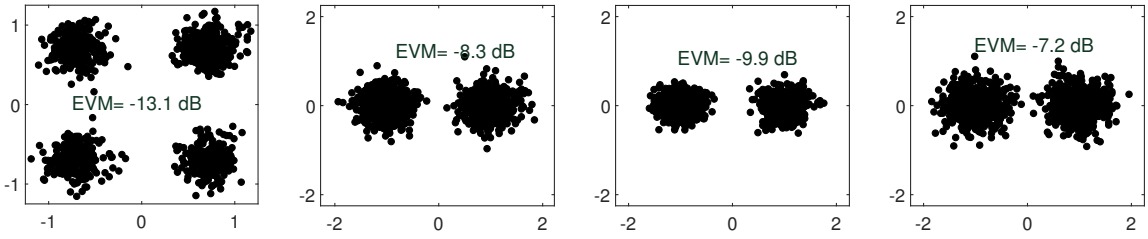
Methodology. The full evaluation of DeepMux requires a large-scale wireless testbed with many STAs to mimic real 802.11ax networks in MU-MIMO-OFDMA transmissions. However, we do not have such a luxury. We therefore use a hybrid approach that combines emulation and experimentation to evaluate DeepMux. Fig. 6.17 shows our testbed setting, where the AP serves 4 real STAs and 16 virtual STAs. The 4 real STAs perform over-the-air transmissions, while the 16 virtual STAs are created by the AP based on the pre-stored CSI from other locations. The virtual STAs are used for DLRA. In the downlink transmission, the AP sends precoded signals to all (real and virtual) STAs, and the performance is measured at STAs.



(a) Original SNRs. (b) Preprocessed SNRs.

(c) RU assignment. (d) Allocated power.

Figure 6.18: A case study on resource allocation by DLRA.

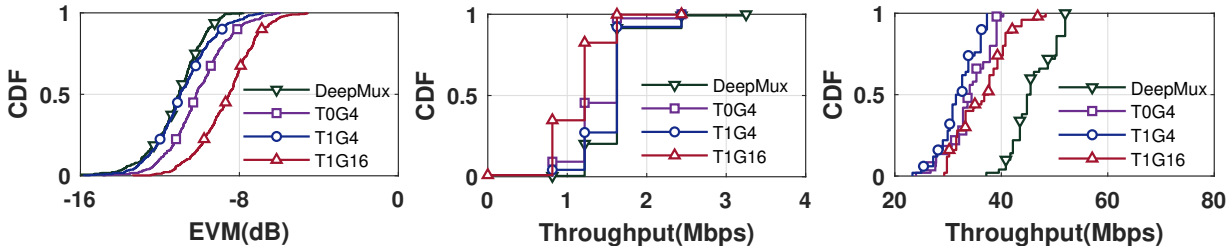


(a) DeepMux (b) 802.11-T0G4 (c) 802.11-T1G4 (d) 802.11-T1G16

Figure 6.19: EVM of decoded signal on first STA over first RU.

A Close Look into DLRA. As a case study, we place one of real STAs at the locations marked by triangle 1 in Fig. 6.11(c). Fig. 6.18(a) shows the SNR values from the real and virtual STAs. The reported SNR values are first preprocessed for normalization, as shown in Fig. 6.18(b). The normalized values are then fed into a DNN for RU assignment. Fig. 6.18(c) shows the RU assignment results from the DNN. With the RU assignment results from DNN, the optimization problem in (6.7) degrades to an LP problem. The LP problem is then solved to obtain the power allocation results, which are shown in Fig. 6.18(d).

Referring to Fig. 6.18(d), the rightmost column denotes RU assignment and allocated power to the STA of interest. Fig. 6.19 shows the constellation of received signal by the mentioned STA on



(a) EVM. (b) Gross throughput. (c) Net throughput.

Figure 6.20: Comparison of DeepMux and 802.11 protocols when an 8-antenna AP serves 20 stations on 20 MHz bandwidth.

the first RU with the aid of DeepMux and existing protocols in 802.11ax. The results reveal superior performance of DeepMux in terms of EVM. For this STA, the gross throughput achieved on the first RU is 2.4 Mbps, 1.2 Mbps, 1.2 Mbps, and 0.8 Mbps with DeepMux, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16, respectively. The net throughput achieved by this user on the first RU is 9.1 Mbps, 4.9 Mbps, 6.5 Mbps, and 4.7 Mbps with DeepMux, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16, respectively. Over all RUs, DeepMux obtains 43.5 Mbps net throughput, while 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 respectively achieve 31.7 Mbps, 29.9 Mbps, and 37.8 Mbps.

Extensive Results. To obtain more comprehensive results, we place the four real STAs at different locations marked with filled circles in Fig. 6.11(c). The experimental results are summarized as follows.

- *EVM:* Fig. 6.20(a) presents the measured EVM at STAs. On average, DeepMux achieves -11.2 dB EVM for STAs, while 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 reach -10.1 dB, -10.9 dB, and -8.6 dB EVM, respectively.

- *Gross Throughput per RU:* Fig. 6.20(b) presents gross throughput per RU. DeepMux achieves 1.6 Mbps gross throughput per 26-tone RU. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 1.4 Mbps, 1.6 Mbps, and 1.1 Mbps, respectively.

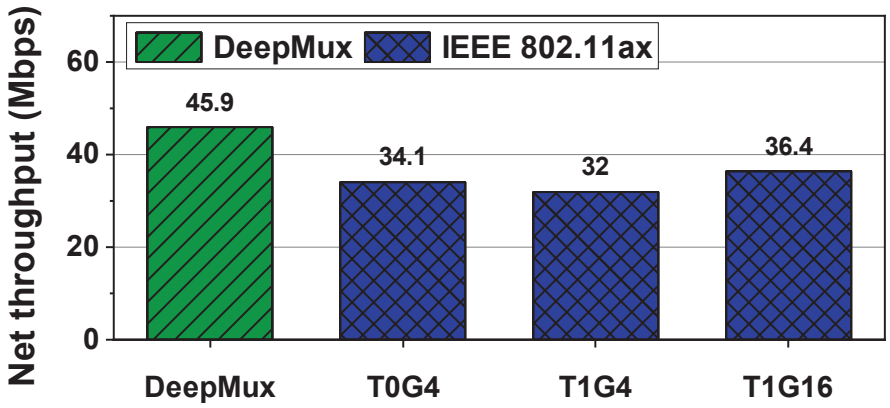


Figure 6.21: Average net throughput achieved by DeepMux and 802.11 protocols.

- *Net Throughput:* Fig. 6.20(c) shows the net throughput achieved by different protocols, and Fig. 6.21 shows the average net throughput at the AP. Specifically, DeepMux achieves 45.9 Mbps net throughput on average. In contrast, 802.11-T0G4, 802.11-T1G4, and 802.11-T1G16 achieve 35.7 Mbps, 32.0 Mbps, and 36.4 Mbps, respectively. The net throughput gain of DeepMux is 34.9% compared to 802.11-T0G4, 43.6%, compared to 802.11-T1G4, and 26.3% compared to 802.11-T1G16.

6.8 Chapter Summary

In this chapter, we presented DeepMux, a deep-learning-based approach to enhance the efficiency of downlink MU-MIMO-OFDMA transmissions in 802.11ax networks. DeepMux is designed upon two components, namely DLCS and DLRA, both of which reside in APs and impose no computation/communication burden to Wi-Fi clients. DLCS leverages DNNs to reduce overhead of CSI feedback in 802.11 protocols. It uses uplink channels to train the DNNs for downlink channels, making the training process easy to implement. Numerical results show that it can reduce the sounding overhead by $62.0\%\sim90.5\%$ without sacrificing CSI feedback accuracy. DLRA tackles an MILP resource allocation problem by decoupling its integer and continuous optimization

sub-problems and employing a DNN to compute a solution to the integer part. Numerical results show that DLRA can achieve 98.9% optimality in RU assignment while bearing a low computational complexity. We have built a wireless testbed to examine the performance of DeepMux in an indoor environment. Experimental results show that DeepMux increases network throughput by $26.3\%\sim43.6\%$ compared to 802.11 protocols.

Chapter 7

A Communication Framework for FL in Intelligent Transportation Systems

7.1 Introduction

Knowledge about vehicles, drivers, environments, and their mutual interactions is critical for ITS. ML techniques have been extensively studied to extract useful knowledge from massive data collected by vehicles so as to enhance the safety and efficiency of ITS. Conventional ML techniques are propelled by a central server with unconditional access to data collected by vehicles and infrastructure. However, with the advancement of autonomous vehicles, the amount of data from the sensors of vehicles (e.g., ;lidars, radars, cameras, and inertial sensors) can easily reach to gigabit per second, making it impractical to transfer raw data to a server, let alone the privacy issue around sharing raw data.

FL has been introduced as a privacy-preserving and communication-efficient alternative, where individual clients (rather than a central server) carry out the model training process [96]. While FL is a promising training paradigm for vehicular networks, the limited communication capacity of these networks along with the heterogeneous sensing, storage, and processing capabilities of individual vehicles, bring up an important question – how to optimize the design and operation of wireless vehicular networks to facilitate FL.

Different strategies have been proposed for FL to address its communication cost, such as decreasing the communication frequency [114], reporting local models using a sparse representation [5, 103, 147, 157], and quantization of model parameters [140, 168, 180]. The main idea of these strategies is to reduce the communication overhead of FL by tuning learning parameters and structure, which will likely cause FL performance degradation. Recently, pioneering work [28, 29, 40, 152, 171, 189, 195] has been conducted to address FL's communication overhead problem from a networking perspective by efficient resource allocation and scheduling schemes. To the best of our knowledge, existing works mainly employ cross-layer optimization techniques to enhance learning efficiency. They assume that global CSI is available at the server. They also assume that CSI remains valid for the time period of an FL iteration (a.k.a. global iteration). Given the small channel coherence time caused by high mobility of vehicles, these two assumptions may not be valid in practical vehicular networks.

In this paper, we present a Communication Framework for FL (CF4FL) for ITS, with the aim of accelerating the FL training process. We consider a vehicular network that comprises a server (for model aggregation and dissemination) and many distributed vehicles (for data collection and local model training). Each vehicle *continuously* collects data samples from its surrounding environment using on-board sensors such as camera, radar, and lidar; and it uses its collected data samples for local model training when scheduled by the server. To embrace topology dynamicity and hardware heterogeneity of vehicular networks, a deadline is defined for each vehicle as the maximum number of global iterations during which the vehicle can keep/store its collected data samples. Once the deadline is reached, the newly collected data samples will be partially or entirely lost due to the limited storage or other limiting factors. CF4FL considers the case where each vehicle has a specific deadline for its data collection. CF4FL mainly comprises two complementary components: DDVS and CVPS.

DDVS is an online scheduler equipped with two scheduling schemes: *a general but complex scheduler* and *a lightweight heuristic scheduler*. DDVS selects a subset of vehicles in each global iteration. The selected vehicles will perform local model training (using their collected data samples) and send their resultant local models to the server in the current FL iteration, while the vehicles that are not selected will continue to collect data samples. Given the deadline of data collection at vehicles, DDVS must meticulously and systematically select the vehicles in each FL iteration to maximize the amount of data samples for local model training and therefore minimize the data loss at vehicles. CVPS, on the other hand, focuses on enhancing the communication capacity between vehicles and server to reduce the duration of a global iteration. CVPS allows the server to concurrently poll multiple vehicles in a global iteration. The key challenge is the time misalignment of multiple concurrent packets caused by the signal propagation delay, packet processing delay, and clock imperfections. CVPS addresses this challenge by a novel spatial signal detection algorithm, which decodes asynchronous data packets from multiple vehicles. CVPS needs neither inter-vehicle synchronization nor instantaneous CSI for asynchronous concurrent vehicle transmissions.

We have evaluated CF4FL through a blend of experimentation and simulation. We implemented CVPS on an SDR vehicular testbed where the server has four antennas and each vehicle has one antenna, and evaluated its performance in three typical scenarios: parking lots, local streets, and highways. Our experimental results shows four vehicles can send their local models to the server simultaneously with 98% success rate. The experimental results are utilized to conduct trace-driven simulation for the performance evaluation of CF4FL. Our results show that, DDVS reduces data loss by 76%, 54%, and 59% compared to Random, Round-Robbin, Earliest-Deadline-First schedulers, respectively. Overall, CF4FL reduces the training convergence time of FL by 39%.

7.2 Related Work

In the literature, there are two research lines involving both FL and networking: *FL for networking* and *networking for FL*. This work belongs to the latter category.

FL in Wireless Vehicular Networks. While many works studied FL applications for transportation systems [63, 145, 164], few investigated the unique challenges of FL in vehicular networks. [198] considered the heterogeneity of local data samples and designed an approach to selectively collect and aggregate local models for fast convergence. [86] proposed a privacy-preserving aggregation for FL in navigation systems. In [26, 136, 137], blockchain-based FL frameworks were proposed to protect the privacy of vehicles when sharing local models. [159] proposed a new clustered architecture for FL in vehicular networks which leverages vehicle-to-vehicle communications to conserve communication resources. [185] proposed a greedy algorithm to accelerate FL by assigning resources to the vehicles with high-quality data samples. [171] and [105] are the most relevant works to this paper. [171] proposed an algorithm for vehicle selection and wireless resource allocation in cellular systems based on dataset content. This work took into account both limited bandwidth and packet error rate in its resource allocation strategy to maximize learning efficiency. The proposed resource allocation is reliant on the exact realization of links capacity and the availability of CSI. While CF4FL pursues a similar objective, it differs from [171] in the problem settings, including the lack of instantaneous CSI and concurrent vehicle polling. [105] accelerates FL in vehicular networks by selecting vehicles with massive local datasets and dropping those with few data samples. It neither considers vehicle-specific deadlines nor focuses on minimizing data loss.

Resource Allocation and Scheduling for FL. Resource allocation and participant scheduling in each global iteration are important for FL convergence. Several research efforts have been made

to study participant scheduling and resource allocation toward different objectives, such as minimizing FL loss [28,29,171], minimizing latency [152], improving energy efficiency [40,189,195], and enhancing learning efficiency [90]. However, these works are limited to stationary or semi-stationary networks where instantaneous CSI with relatively large coherence time can be estimated at the server in each global iteration. In practice, such an luxury is barely available, especially in vehicular networks. CF4FL differs from these efforts in terms of requirements, objective, and network settings.

Communication Airtime Overhead of FL. The limited communication capacity is a realistic barrier for the FL deployment in wireless networks, which throttles the learning process and slows down the learning convergence. Pioneering works have been done to resolve the communication problem for FL using different approaches, such as decreasing the communication frequency (i.e., the number of global iterations) [114], reporting local models using their sparsified representations [5,103,147,157], and quantization of model parameters [140,168,180]. Apparently, CF4FL is orthogonal to these efforts as it does not optimize FL but innovates the networking design to improve the convergence speed of FL training.

7.3 Federated Learning in Vehicular Networks

The deployment of FL in vehicular networks is a complex task due to the unique features of vehicular networks and the stringent requirements of data collection. CF4FL assumes that vehicles can label their collected data for local training. It also assumes that vehicles have sufficient computational power for local training [127]. In what follows, we first describe the system model and then formulate the problem. Finally, we point out the challenges in the design of an efficient solution.

7.3.1 System Model

Practical realization of ITS requires to collect an immense amount of data in vehicular environments, such as information of other vehicles, the condition of road surface, the probability of accidents, and the existence of local objects. The collected data by different vehicles is a valuable source of information to train ML models for the applications such as pothole detection, collision avoidance, object/pedestrian identification, and curb avoidance. For vehicles, sharing their raw data samples with a central server for training a unified model may not be a good idea due to the concerns about data privacy, the limited network bandwidth, and the huge size of data samples. FL alleviates these issues and offers a decentralized framework to train a *global model* through the *local model* training at individual vehicles using their privately-owned data.

We consider a vehicular network that comprises a central server and many vehicles as shown in Fig. 7.1 , where each vehicle is equipped with a single antenna and the server is equipped with multiple antennas. The server is responsible for vehicle scheduling, receiving local models from scheduled vehicles, aggregating models, and broadcasting the aggregated model to all vehicles. Each vehicle performs *continuous* data collection from its surrounding environment using its onboard sensors. If it is scheduled in the current iteration, it first uses its collected data for local model training and then reports the updated local model to the server; otherwise (not scheduled), it continues to collect data until its deadline is reached. To describe FL training, let us consider a network at global iteration t. Denote $\mathcal{N}(t)$ as the set of vehicles associated to a server, with $|\mathcal{N}(t)| = N(t)$. Denote M as the number of antennas at the server. Denote $\mathcal{I}_i(t)$ as the dataset at vehicle i in global iteration t. Assume that the data collection and transmissions at vehicles are done in parallel. Also, assume that a unique frequency band (e.g., a channel in 802.11p or a resource block in C-V2X) is assigned to the FL task under consideration, and the frequency band

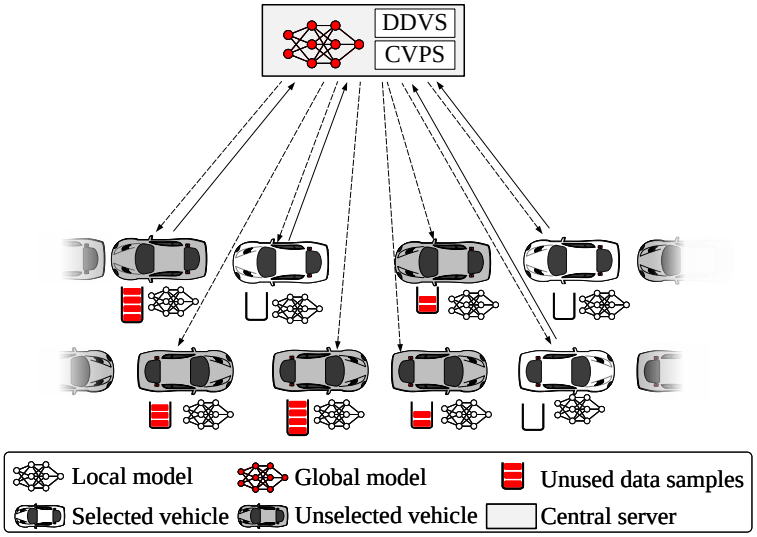


Figure 7.1: FL in a vehicular network.

is used via TDMA for communications between vehicles and the server.

7.3.2 Problem Formulation

In the conventional training process of FL as shown in Fig. 7.1 and Fig. 7.2, the server selects a subset of vehicles for local model training. Denote $\mathcal{S}(t)$ as the set of selected vehicles in global iteration t. Each selected vehicle, say i, trains its local model to minimize a loss function. Denote $\Theta_i(t)$ as the local model parameters. Then, the loss function can be written as: $L\left(\Theta_i(t), \mathcal{I}_i(t) | \Theta_i(t-1)\right)$, where $\Theta_i(t-1)$ is the initial parameters of local model. In the rest of this paper, we drop the condition of $\Theta_i(t-1)$ for notation simplicity. Vehicle i sends $\Theta_i(t)$ to the server and discards $\mathcal{I}_i(t)$ in its buffer to collect future data. An unselected vehicle, on the other hand, piles up the collected data samples during the current iteration on top of what it already has in its buffer, until it has been selected. Piling up data samples can adversely affect the entire training process, especially in a dense vehicular network where some vehicles would not be selected for many global iterations. These vehicles face several issues. *First*, the size of unused data samples (i.e., data samples collected since the last participation in FL) may exceed the storage limit. *Sec-*

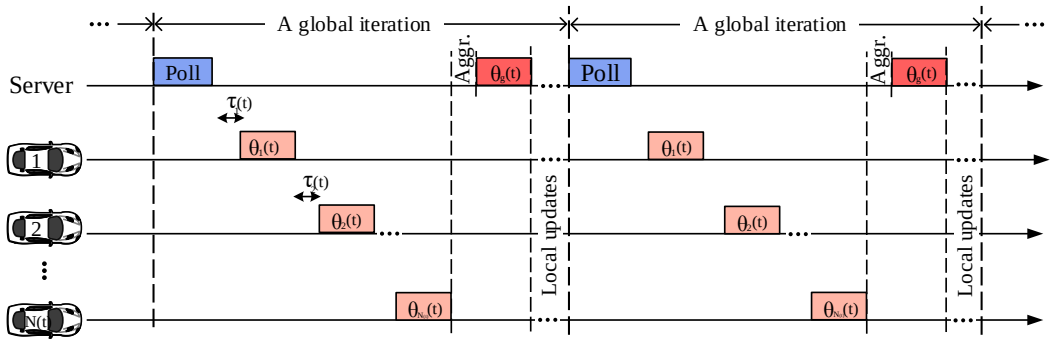


Figure 7.2: Sequential polling for the iterative training of FL in vehicular networks.

ond, vehicle stragglers (i.e., computation-slow vehicles) drastically increase the time duration of a global iteration. The local processing time can contribute to a time limit (i.e., deadline) for each vehicle or equivalently a virtual cap on an allowable amount of data that the vehicle can collect. We denote this cap as F_i (in bits) for vehicle i. When vehicle i is not selected for a long time and its unused data samples exceed F_i bits, the data samples will be lost from that point on.

At the end of a global iteration, the server receives local models from the selected vehicles and aggregates them to obtain a new global model. The contribution of each local model to the global one is proportional to the amount of data that is used in training that local model [29]. Simply put, the aggregation is a weighted average of the polled local models. Denote $\Theta_g(t)$ as the parameters of the aggregated global model. Then, we have

$$\Theta_g(t) = \frac{\sum_{i\in\mathcal{S}(t)} |\mathcal{I}_i(t)| \cdot \Theta_i(t)}{\sum_{i\in\mathcal{S}(t)} |\mathcal{I}_i(t)|}. \tag{7.1}$$

Similarly, the global loss is evaluated as a weighted average on loss of local models, i.e.,

$$L(\Theta_g(t)) = \frac{\sum_{i\in\mathcal{S}(t)} |\mathcal{I}_i(t)| \cdot L\left(\Theta_i(t), \mathcal{I}_i(t)\right)}{\sum_{i\in\mathcal{S}(t)} |\mathcal{I}_i(t)|}. \tag{7.2}$$

As shown in Fig. 7.2, the server then broadcasts the global model to all the vehicles, includ-

ing those who were not selected for local model training in the current global iteration. At all vehicles, their local models are replaced with the global one to initialize the next iteration of local model training. The network continues global iterations until the global model converges, e.g., $|L(\Theta_g(t)) - L(\Theta_g(t-1))| \leq \epsilon$, where ϵ is a pre-defined threshold. We define the learning efficiency of global iteration t as $|L(\Theta_g(t)) - L(\Theta_g(t-1))| \cdot \frac{1}{\Delta t}$, where Δt is the time duration of global iteration t.

Now, the question to ask is how to increase learning efficiency in each global iteration while avoiding data loss due to the deadline and storage constraints. To answer this question, we first formulate the learning efficiency in its general form for conventional FL setting. As shown in Fig. 7.2, Δt is dominated by the uplink and downlink data transmissions as well as local processing time. Then, it can be calculated as:

$$\Delta t = \sum_{i \in \mathcal{S}(t)} \left(\tau_i(t) + \frac{Z(\Theta_i(t))}{C_{ui}(t)} \right) + \frac{Z(\Theta_g(t))}{\min\limits_{i \in \mathcal{N}(t)} C_{di}(t)}, \tag{7.3}$$

where $\tau_i(t)$, $C_{ui}(t)$, and $C_{di}(t)$ are vehicle i's local processing time, uplink data rate, and downlink data rate, respectively. $Z(\cdot)$ returns the size of its input in bits. Here, we have $Z(\Theta_i(t)) = Z(\Theta_g(t)), \forall i \in \mathcal{S}(t)$. The uplink transmissions also take place sequentially in a TDMA manner. In practice, the server typically limits the maximum time duration of a global iteration, i.e., $\Delta t \leq T$, where T is a pre-defined constant.

In each global iteration, the server selects a subset of vehicles that maximize learning efficiency without losing collected data at vehicles. Hence, the problem can be formulated as:

$$\underset{\mathcal{S}(t)}{\text{maximize}} \quad \frac{1}{\Delta t} \cdot \big(L(\Theta_g(t)) - L(\Theta_g(t-1)) \big) \tag{7.4a}$$

$$\text{s.t.} \quad \Delta t \leq T, \tag{7.4b}$$

$$|\mathcal{I}_i(t+1)| \leq F_i, \quad \forall i \in \mathcal{N}(t)/\mathcal{S}(t), \tag{7.4c}$$

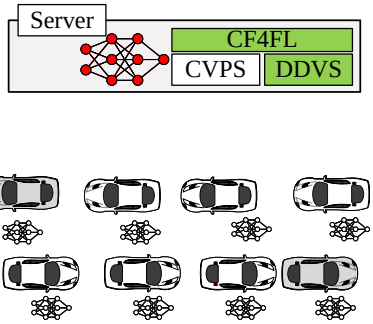
where (7.4b) is the maximum allowable time for a global iteration, and (7.4c) attempts to avoid data loss for the vehicles that are not selected for local model training in the current global iteration.

7.3.3 Challenges

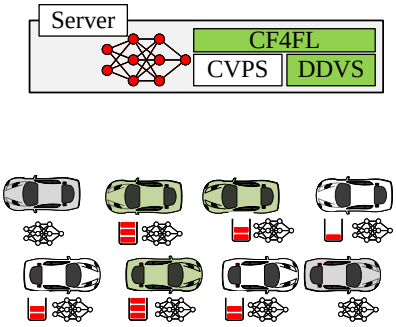
There are two challenges to solve the problem in (7.4).

Challenge 1. Solving (7.4) requires perfect global CSI knowledge for the server to select vehicles. However, obtaining fresh CSI for the server is extremely difficult, if not impossible. This is because the channel coherence time in vehicular networks is too short for channel acquisition. For example, consider a relatively small neural network model with $8,778$ parameters, which we later use for digit classification in Section 7.7. If each parameter is represented by 32 bits, it takes at least 10.1 ms to transmit the entire model in the uplink using the most aggressive MCS of IEEE 802.11p, 64QAM and 3/4 coding rate. However, 10.1 ms is very likely beyond the channel coherence time of many vehicular networks. To address this challenge, we propose a deadline-driven vehicle scheduler, which allows the server to poll vehicles in the absence of CSI.

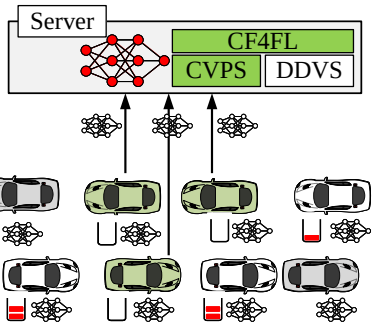
Challenge 2. Another challenge is to reduce the airtime consumption of a global iteration. A natural approach is uplink MU-MIMO transmission, which allows the server to communicate with multiple vehicles at the same time. However, existing uplink MU-MIMO schemes require the packets from vehicles to be aligned in time. This is extremely hard in vehicular networks due to the high mobility (e.g., 60 mph) and the dynamic network topology. Pursuing network-wide timing synchronization, even if possible in practice, inevitably entails a large amount of airtime overhead. To combat this challenge, we propose an asynchronous uplink MU-MIMO scheme, which allows



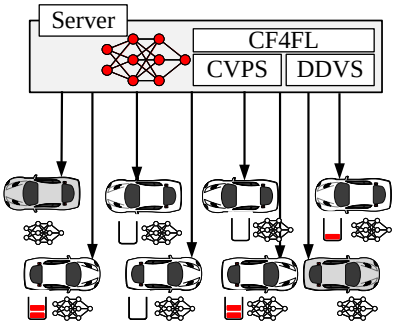
(a) Step 1: If necessary, DDVS removes some of the vehicles (gray colored) and considers a set of vehicles which are schedulable (white colored).



(b) Step 2: Without knowing CSI, DDVS designs a scheduler merely based on vehicles deadlines and buffer status and selects a subset vehicles (green colored).



(c) Step 3: CVPS recovers local models which are concurrently sent by selected vehicles.



(d) Step 4: Server aggregates the local models and sends back the global model to all vehicles.

Figure 7.3: An overview of CF4FL and its underlying components: DDVS schedules the vehicles without requiring global CSI and CVPS recovers concurrent, but asynchronous, packets transmitted by vehicles.

the server to receive packets from multiple vehicles at the same time.

7.4 CF4FL: Overview

CF4FL is a heuristic vehicular communication framework to accelerate FL in general. To maximize the learning efficiency in (7.4), CF4FL focuses on two tasks. *First*, CF4FL endeavors to maximize the numerator of the objective function in (7.4a) (i.e., learning accuracy). CF4FL pursues the the same objective as the schedulers proposed in [29,94,152,171], in which the analysis shows that learning efficiency will be improved by using (consuming) more data samples for training local

models. CF4FL assumes that data at all vehicles are independent and identically distributed (iid) and bear the same quality. It also assumes a direct relation between the data consumption of local training and the convergence of global model. CF4FL considers the vehicle-specific deadlines and avoids data sample loss at vehicles. Leveraging the maximum number of local data samples for training the global model, CF4FL improves the objective function in (7.4a). *Second*, given the set of selected vehicles (i.e., $\mathcal{S}(t)$), concurrent polling minimizes the denominator of the objective function in (7.4a) (i.e., Δt). CF4FL strives to solve (7.4) and meet constraints (7.4b) and (7.4c), provided that the original problem has a feasible solution. These two tasks will be carried out by DDVS and CVPS, respectively, as shown in Fig. 7.3. In what follows, we highlight the key components of CF4FL.

Server. As the central controller, the server is responsible for three tasks: i) passing appropriate information (i.e., deadlines, which are translation of F_i in time domain) to DDVS; ii) aggregating local models; and iii) broadcasting the global model at the end of each global iteration. The calculation of the deadlines is detailed in Section 7.5.1.

DDVS. At the beginning of each global iteration, DDVS receives the deadlines from server and designs a scheduler to poll at most M vehicles by CVPS, where M is the number of antennas at the server. It is the available spatial DoF for polling. If the scheduling problem is feasible, it guarantees that a vehicle is polled before reaching its deadline. Unfortunately, the scheduling problem is not always feasible. As such, DDVS first determines the feasibility of the scheduling problem. If infeasible, DDVS removes some vehicles with the shortest deadlines to make the scheduling problem feasible. This process is illustrated in Fig. 7.3(a). Once the scheduling feasibility (termed schedulability) is secured, DDVS finds a scheduler to poll the remaining vehicles within a finite number of iterations.

CVPS. Upon receiving a poll frame from DDVS, the selected vehicles prepare their local mod-

els and send them to the server. CVPS leverages multiple antennas at the server to decode the uplink data packets. As the vehicles are asynchronous in nature, CVPS first compensates the time and frequency offsets of the collided frames. Then, it constructs a spatial detection filter to recover the data packets, from which local models are extracted by the server.

7.5 Deadline-Driven Vehicle Scheduler (DDVS)

DDVS is responsible for examining the feasibility of (7.4) and designing a scheduler based on the deadlines specified by the server. We first propose a general scheduler to find a cyclic scheduler that guarantees zero data loss if the *network deadline* (deadlines of all vehicles in the network) is schedulable. The general scheduler comes with a high computational complexity as it needs to find a cycle on a large graph called *steady state graph*, making it hard to implement for large vehicular networks. We therefore propose a lightweight scheduler to handle vehicle selection problem in large vehicular networks.

7.5.1 Network Deadline and State

DDVS determines vehicles' polling order based on their deadline and state, which we describe below.

Deadline. For a vehicle, say vehicle i, we denote its deadline as d_i. It indeed translates F_i into time domain based on three parameters: the worst-case duration of a global iteration, the sensing rate of vehicle i, and processing delay of vehicle i, which are denoted by t_w, b_i, and τ_i, respectively. t_w is conservatively defined with respect to the case where the lowest MCS in 802.11p is used for all transmissions in a global iteration. b_i and τ_i root in hardware capabilities of vehicle i. We also assume the vehicles persistently collect data in time domain, and the number of

their collected data samples linearly increases over time. Then, the deadline for vehicle i is defined as $d_i = \lfloor F_i/(t_w.b_i) + \tau_i/t_w \rfloor$, which is reflected in (7.4c).

Zero data loss is guaranteed for vehicle i if it collects data no more than d_i subsequent global iterations before being polled. With respect to the individual deadline of vehicles, we further define network deadline as $\vec{d}(t) \triangleq \left(d_1, d_2, \cdots, d_{N(t)}\right)$ for global iteration t. The individual and network deadlines are calculated at the server. Vehicle i, reports F_i and b_i once to the server as a part of its association process. On the other hand, the server is aware of M, $Z(\theta_g)$, and MCS; therefore, it easily obtains t_w which is fixed during the whole training cycle. Then, the server calculates d_i.

State. To indicate the number of global iterations elapsed from the last time a vehicle has been polled, we define a counter, i.e., buffer state. For vehicle i, the buffer state is denoted by $p_i(t)$ and can be written as:

$$p_i\left(t\right) = \begin{cases} 1, & \text{if } i \in \mathcal{S}(t); \\ p_i(t-1)+1, & \text{if } i \notin \mathcal{S}(t). \end{cases} \tag{7.5}$$

We further define *network state* as $\vec{p}(t) \triangleq \left(p_1(t), p_2(t), \cdots, p_{N(t)}(t)\right)$ for global iteration t.

7.5.2 Schedulability of Network Deadline and General Scheduler

The objective of DDVS is to design a scheduler to honor the constraint of $\vec{p}(t) \preccurlyeq \vec{d}(t)$ for $\forall t > 0$. To design such a scheduler, we need to answer two fundamental questions: i) Is the network deadline schedulable (i.e., the existence of a scheduler that satisfies the constraints in (7.4))? ii) If a network deadline is schedulable, how to find a scheduler for it? To answer these two questions, we have the following remarks. First, not every network deadline is schedulable. If the network deadline is not scheduable, DDVS first removes some of vehicles to secure the scheduability. Second, for a schedulable network deadline, DDVS designs a cyclic scheduler, which turns out to be optimal but with high computational complexity. Subsequently, a low-complexity heuristic is

designed for large-scale vehicular networks.

The two tasks, examining the schedulibility of a network deadline and designing a scheduler, are tightly intertwined. Apparently, the network deadline $\vec{d}(t)$ is schedulable if there exists a scheduler $\mathcal{S}$ (with $\mathcal{S}(t)$ being the set of selected vehicles at global iteration t) such that $\vec{p}(t) \preccurlyeq \vec{d}(t)$ for $\forall t > 0$. Such a scheduler should select vehicle i at least once per d_i global iterations. In addition, as we will show, CVPS will allow the server to poll M vehicles in a global iteration. Define$l(\vec{d}(t)) \triangleq \sum_{i=1}^{N(t)} \frac{1}{d_i}$ as the *network load*. Then, we have the following necessary condition for the schedulibility of $\vec{d}(t)$.

Lemma 5. *If $\vec{d}(t)$ is schedulable, then $l(\vec{d}(t)) \leq M$.*

Proof. For vehicle i, zero data loss is guaranteed if it is polled at least once in every d_i subsequent global iterations. Let us assume the network's deadline does not change within $t \in [0,\ T]$. Polling rate r_i for vehicle i then satisfies $r_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} I_i(\mathcal{S}(t)) \geq \frac{1}{d_i}$, where $I_i(\mathcal{S}(t)) = 1$ if $i \in \mathcal{S}(t)$; otherwise, $I_i(\mathcal{S}(t)) = 0$. For all vehicles, we have $M \overset{(a)}{\geq} \sum_{i=1}^{N(t)} r_i \overset{(b)}{\geq} \sum_{i=1}^{N(t)} \frac{1}{d_i}$, where (a) holds as M is the maximum number of vehicles that can be polled by CVPS in global iteration, and (b) is directly concluded from the constraint on polling rate. ■

Lemma 5 implies that the network load supported by the server should not exceed M. DDVS does not set a limit on the number of vehicles for scheduling. Instead, it sets a limit on the network load for schedulability. DDVS can work for a small-size network with as less as M vehicles or a large-scale network with many vehicles. DDVS first determines if the network deadline meets the necessary condition. If not, DDVS removes vehicles with the smallest deadlines one by one until the condition is met. This treatment follows two reasons. First, a vehicle with the shortest deadline is the bottleneck of scheduling as it has the highest contribution to the network load. Second, a vehicle with the shortest deadline has the lowest contribution in improving global model per poll.

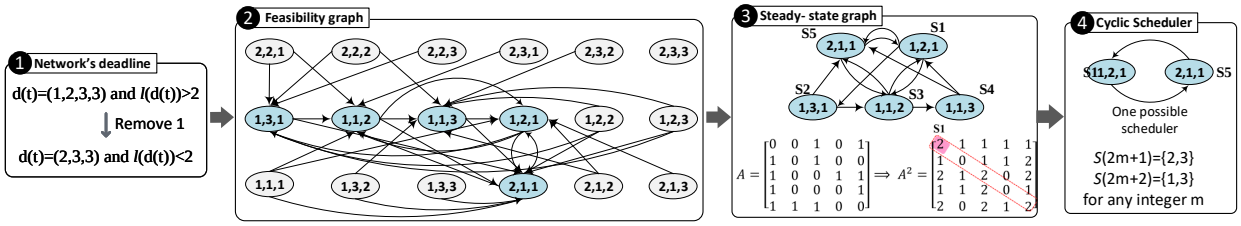


Figure 7.4: An example of checking the necessary condition of schedulibility, constructing feasibility and steady-state graphs, finding the shortest cycle, and obtaining a cyclic scheduler.

We use a small example shown in Fig. 7.4 to illustrate this process. In this example, four vehicles with network deadline $\vec{d}(t) = (1, 2, 3, 3)$ are associated to a server with two antennas, i.e., $M = 2$. Referring to step 1 in the example, the initial network load is $l(\vec{d}(t)) = 2.16$. As the network load does not meet the necessary condition, DDVS removes first vehicle with $d_1 = 1$ and updates the network deadline to $\vec{d}(t) = (2, 3, 3)$. Then, we have $l(\vec{d}(t)) < 2$, which meets the necessary condition of schedulability.

Feasibility Graph. Once $\vec{d}(t)$ meets the necessary condition of schedulability, DDVS examines the schedulability of $\vec{d}(t)$. To do so, DDVS constructs a feasible scheduling space including feasible network states and possible transitions between the network states. The feasible scheduling space is constructed using a directed graph called *feasibility graph*. The feasibility graph G is constructed as follows.

$$G = (V, E), \tag{7.6a}$$

$$V = \{\vec{p}(t) : \vec{p}(t) \preccurlyeq \vec{d}(t)\}, \tag{7.6b}$$

$$E = \{\vec{p}(t-1) \to \vec{p}(t) : \exists\, \mathcal{S}(t) \text{ and } \vec{p}(t) \in V\}. \tag{7.6c}$$

Referring to Fig. 7.4, the constructed feasibility graph for a given network deadline $\vec{d}(t) = (2, 3, 3)$ is shown in Step 2. To further clarify state and transitions in this graph, let us focus on an example where the network state is $\vec{p}(t) = (1, 3, 1)$ and $\mathcal{S}(t) = \{1, 2\}$. Since vehicles 1 and 2 are selected,

their buffer will be cleared and the network state transits to $\vec{p}(t+1) = (1,1,2)$ according to (7.5). Since $\vec{p}(t) \preccurlyeq \vec{d}(t)$, $\vec{p}(t+1) \preccurlyeq \vec{d}(t+1)$, and $|\mathcal{S}(t)| = M$, both states and corresponding transition belong to the feasibility graph. As an another example, let us consider the case that, for the same initial state, i.e., $\vec{p}(t) = (1,3,1)$, first and third vehicles are selected for polling. Then, $\vec{p}(t+1) = (1,4,1)$, which is not a feasible state. Therefore, $\vec{p}(t+1)$ and the transition from $\vec{p}(t)$ are not in the feasibility graph.

Per (7.6c), an edge in the feasibility graph corresponds to a scheduling decision, and a cycle in the graph corresponds to a cycle of decisions that can be followed for infinite time. Therefore, a unique correspondence exists between a cycle on feasibility graph G and a cyclic scheduler $\mathcal{S}$. A cycle with length c in the feasibility graph is equivalent to a *cyclic scheduler* having $\mathcal{S}(t+c) = \mathcal{S}(t)$ for $t > 0$. It is easy to see that, under such a cyclic scheduler, we also have $\vec{p}(t+c) = \vec{p}(t)$ for $t > c$. This correspondence is exploited to determine the schedulibility of a network deadline.

Lemma 6. *$\vec{d}(t)$ is schedulable if and only if there is a cycle on the feasibility graph G in* (7.6)*, and the repetition of the cycle represents a feasible cyclic scheduler for* (7.4)*.*

Proof. If the deadlines and number of vehicles are finite, the number of vertices in G is also finite as:

$$|V| = \prod_{i=1}^{N(t)} d_i < \infty, \text{ if } N(t) < \infty \text{ and } d_i < \infty \ \forall i \in \mathcal{N}(t). \tag{7.7}$$

Therefore, there exists two distinct FL rounds t' and t'' with $1 \leq t' < t'' \leq |V| + 1$ for which $\vec{p}(t') = \vec{p}(t'')$. Any path with a length larger than $|V|$ includes a cycle. In other words, we cannot infinitely move on the feasibility graph without forming any cycle. If $\vec{p}(t') = \vec{p}(t'')$ and a cycle is formed, all vehicles are polled at least once within $[t', t'']$. This statement itself can be proved

based on contradiction. If vehicle i is not polled, we have:

$$\nexists\, t \in [t', t''] \text{ s.t. } i \in \mathcal{S}(t), \tag{7.8a}$$

$$p_i(t') < p_i(t''), \tag{7.8b}$$

$$\vec{p}(t') \neq \vec{p}(t''). \tag{7.8c}$$

where (7.8b) follows from the fact that buffer state is a monotonic increasing function as long as the vehicle is not polled. (7.8b) directly results (7.8c) which is in contradiction to our assumption $\vec{p}(t') = \vec{p}(t'')$. Therefore, all vehicles will be polled at least once within a cycle. The repetition of the cycle on feasibility graph G results in a cyclic scheduler which results $p_i(t) \leq d_i$ for $t > 0$ and $\forall i \in \mathcal{N}(t)$. ■

.

Lemma 6 implies that finding a scheduler for $\vec{d}(t)$ is equivalent to finding a cycle on graph G. The complexity of such a search is $\mathcal{O}(|V| + |E|)$ [161], which is likely to be intractable in practice.[1]

Pruning feasibility graph. To reduce the computational complexity, we narrow down the search space by pruning graph G while maintaining its cycles. This is done by removing all the network states and their connected edges that cannot be a part of any cycle in G. We call the pruned graph *steady state graph* $G_s = (V_s, E_s)$. If $\vec{p}(t) \in V$ is also a vertex in V_s, it has the following features: i) no more than M elements of $\vec{p}(t)$ have hit their deadline; ii) no more than M elements of $\vec{p}(t)$ that have not hit the deadline carry the same values; and iii) one or more elements of $\vec{p}(t)$

[1] The number of states in a feasibility graph is $|V| = \prod_{i=1}^{N(t)} d_i$, and the number of outgoing edges from one state can reach up to $C_M^{N(t)}$, which is the number of M-combinations from $N(t)$ vehicles. For a network with tens of vehicles and a few antennas at the server, $|V| + |E|$ can easily reach to millions, making the search for a cycle on G intractable.

equal to 1. The latter feature can be relaxed to the case where exactly M elements are equal to 1, if the vehicular network is very dense and DDVS intends to use all available spatial degrees of freedom to poll M vehicles in each global iteration. It is obvious that all cycles in G do exist in G_s and vice versa. Obtaining the steady state graph is illustrated in Step 3 of the example shown in Fig. 7.4, where the feasibility graph is pruned as described above. The steady state graph is much smaller than the feasibility graph. For the example in Fig. 7.4, it has only 5 states (marked as S1 to S5).

Shortest Cycle in Steady State Graph G_s. The shortest cycle in G_s is critical as it keeps $p_i(t)$ for all $i \in \mathcal{N}(t)$ at small values. Hence, if a vehicle suddenly leaves the network or stops participating in FL, then a small number of data samples will be lost. To find a cycle with the shortest length, we sort vertices in V_s and derive an adjacency matrix $\mathbf{A}$ for G_s such that $\mathbf{A}(i,j) = 1$ if there exists an edge from vertex i to vertex j, and $\mathbf{A}(i,j) = 0$ otherwise. If $N(t) > M$, which is the case for a typical vehicular network, all the diagonal entries of $\mathbf{A}$ are zero, i.e, $\text{diag}(\mathbf{A}) = \mathbf{0}$ where $\mathbf{0}$ denotes an all-zero vector with length $|V_s|$. For such an adjacency matrix, the shortest cycles has length n if n is the smallest integer number for which $\text{diag}(\mathbf{A}^n) \neq \mathbf{0}$. A state corresponding to the position of a non-zero element on diameter of $\mathbf{A}^n$ is located on the shortest cycle(s). Then, we can leverage Floyd–Warshall algorithm [47] to find the shortest cycle for that state. If $N(t) \leq M$, even with the most pressing deadlines, i.e., $d_i = 1 \ \forall i \in \mathcal{N}(t)$, the network load meets the necessary condition of schedulibility. All vehicles will be scheduled to serve in every global iterations. In the steady state graph, this scheduler is a self-loop that starts and ends at the same state in every global iteration.

Referring to the example in Fig. 7.4, the adjacency matrix is calculated. As the first element on the diameter of $\mathbf{A}^2$ is non-zero, the length of the shortest cycle is 2 and it passes through S1. DDVS finds such a cycle on the steady state graph. It chooses one of the two existing cycles with such conditions. In this example, the cycle between S1 and S5 is selected. The cycle corresponds

Algorithm 7.1 A deadline-driven cyclic scheduler.

Input. The network deadline $\vec{d}(t)$
if $\vec{d}(t) = \vec{d}(t-1)$ **then**
    Keep current scheduler
else
    $\mathcal{Q} = \mathcal{N}(t)$
    $\mathcal{S} = \emptyset$
    while $\sum_{i\in\mathcal{Q}} 1/d_i > M$ or $\mathcal{S} = \emptyset$ **do**
        $\mathcal{Q} \longleftarrow \mathcal{Q}\backslash\{i\}$ such that $d_i = \min\{\vec{d}(t)\}$
        Update $\vec{d}(t)$
        if $\sum_{i\in\mathcal{Q}} 1/d_i < M$ **then**
            Construct feasibility graph G
            Obtain G_s from G with adjacency matrix $\mathbf{A}$
            if $\exists n \in \mathbb{N}$ such that $\text{diag}(\mathbf{A}^n) \neq \vec{\mathbf{0}}$ **then**
                Find smallest n
                Find cyclic scheduler $\mathcal{S}$ with length n using Floyd–Warshall algorithm [47]
Return $\mathcal{S}$

to a cyclic scheduler which selects vehicles 2 and 3 on odd global iterations and selects vehicle 1 and 3 on even ones.

A General Scheduler. Alg. 7.1 presents an algorithm to check the schedulability of $\vec{d}(t)$ and construct a cyclic scheduler. It comprises four main steps: preparing network deadline, constructing and pruning feasibility graph, constructing steady state graph, and finding the shortest cycle.

Computational Complexity of General Scheduler. The computational complexity of finding the shortest cycles on the steady state graph $G_s = (V_s, E_s)$ using Floyed-Warshall algorithm is $\mathcal{O}(|V_s|^3)$. The number of vertices in G_s can be approximated by: $|V_s| \approx \sum_{j=1}^{C_M^{N(t)}} \prod_{k\in\mathcal{C}_{Mj}}(d_k) - \sum_{i=M+1}^{N(t)} C_i^{N(t)} - \sum_{i=M+1}^{N(t)} \sum_{j=1}^{C_i^{N(t)}} \min_{k\in\mathcal{C}_{ij}}(d_k - 1)$ where where $\mathcal{C}_{ij}$ is the jth realization of i-combinations from $N(t)$ vehicles. In the worst case, the computational complexity of the general scheduler is $\mathcal{O}(d_{max}^{3(N(t)-M)})$, where $d_{max} = \max_{i\in\mathcal{N}(t)}\{d_i\}$. It can be seen that the computational complexity grows polynomially w.r.t. deadlines and exponentially w.r.t. the number of vehicles. Due to its high complexity, this scheme is intractable in dense vehicular networks.

7.5.3 A Lightweight Scheduler

While Alg. 7.1 is capable of constructing a cyclic scheduler for a given network deadline, it is of high computational complexity and thus only suited for small networks. For large-scale networks, we propose a heuristic called Extended Polynomial Scheduler Extended Polynomial Scheduler (EPS), which is of a low computational complexity.

Main Idea. EPS was inspired by the transformation of "Fictitious Polynomial Mapping" in [95]. The main idea behind EPS is to map a network deadline $\vec{d}(t)$ to a Fictitious Polynomial Deadline (FPD) $\vec{\tilde{d}}(t)$ that satisfies $\vec{\tilde{d}}(t) \preccurlyeq \vec{d}(t)$. Based on FPD, we propose EPS for the polynomial deadline $\vec{\tilde{d}}(t)$. Given $\vec{\tilde{d}}(t) \preccurlyeq \vec{d}(t)$, the proposed scheduler by EPS will also meet the original deadline $\vec{d}(t)$.

Transformation. EPS is designed based on a special structure of FPD. A vector $\vec{\tilde{d}}(t) = (\tilde{d}_1, \tilde{d}_2, \cdots, \tilde{d}_{N(t)})$ is FPD if $\tilde{d}_i = b \cdot 2^{m_i} i$ for $\forall i \in \mathcal{N}(t), b \in \mathbb{N}$, and $m_i \in \mathbb{Z}$ [95]. Now, a question is that for a given $\vec{d}(t)$, how can we find an FPD $\vec{\tilde{d}}(t)$ such that $d_i \geq \tilde{d}_i$ for all i's and $l(\vec{\tilde{d}}(t)) \leq M$? To find such an FPD, it is sufficient to check $N(t)$ different realizations of FPD, i.e., $d_i = \tilde{d}_i$ for $i \in \{1, 2, \cdots, N(t)\}$. Specifically, for each $i \in \{1, 2, \cdots, N(t)\}$, we construct $\vec{\tilde{d}}(t) = \left(d_i 2^{\lfloor \log_2(d_1/d_i) \rfloor}, d_i 2^{\lfloor \log_2(d_2/d_i) \rfloor}, \cdots, d_i 2^{\lfloor \log_2(d_{N(t)}/d_i) \rfloor}\right)$. We can find a mapping for $\vec{d}(t)$ if and only if we have $l(\vec{\tilde{d}}(t)) \leq M$ for one of these realizations. If such an FPD is not found from all the realizations, we remove vehicles with the shortest deadlines one by one until an FPD is found. It is worth noting that for a single network deadline, multiple FPDs with suitable load may exist. In such a case, we pick the FPD with the lowest load.

We illustrate the mapping procedure through the example as shown in Fig 7.5. In this example, the network deadline is $\vec{d}(t) = (2, 2, 3, 3, 3, 4, 5, 6, 7, 9, 9, 9, 10)$, yielding $l(\vec{d}(t)) = 3.19$. We pick $d_1 = 2$ for mapping. Then, we have $\vec{\tilde{d}}(t) = (2 \times 2^{\lfloor \log_2(2/2) \rfloor},\ 2 \times 2^{\lfloor \log_2(2/2) \rfloor},\ \cdots,\ 2 \times$

$2^{\lfloor \log_2(10/2) \rfloor}) = (2,2,2,2,2,4,4,4,4,8,8,8,8)$, yielding $l(\vec{\tilde{d}}(t)) = 4$. Since the network load meets $l(\vec{\tilde{d}}(t)) \leq M$ condition in Lemma 5, we use the polynomial deadline $\vec{\tilde{d}}(t)$ for scheduling.

Grouping. Once an FPD with a proper load is found, we then construct a feasible scheduler. For the special case where $M = 1$, the Fictitious Scheduler Construction (FSC) algorithm in [95] can provide a feasible scheduler when $\vec{d}(t)$ can be mapped to $\vec{\tilde{d}}(t)$ whose load is no more than one, i.e., $l(\vec{\tilde{d}}(t)) \leq 1$. Therefore, for the general case with $M > 1$, if we can divide $\mathcal{N}(t)$ into M separate groups and each group can be mapped to an FPD with load no greater than 1, then a feasible scheduler can be constructed. We now present a procedure to divide $\mathcal{N}(t)$ into such M separate groups. Recall that $\vec{d}(t)$ can be mapped to an FPD $\vec{\tilde{d}}(t)$ with $l(\vec{\tilde{d}}(t)) \leq M$. Without loss of generality, we assume $\tilde{d}_1 \leq \tilde{d}_2 \leq \cdots \leq \tilde{d}_{N(t)}$. Then, we pick the first k elements with $\sum_{i=1}^{k-1} 1/\tilde{d}_i < 1$ and $\sum_{i=1}^{k} 1/\tilde{d}_i \geq 1$ as one group. Again, referring to the example shown in Fig. 7.5, the FPD is divided into four groups, each of which holds a load no less than 1. The deadlines of the four groups are $\{2,2\}$, $\{2,2\}$, $\{2,4,4\}$, and $\{4,4,8,8,8,8\}$.

Construction of feasible scheduler. To design the feasible scheduler, we apply FSC on each group. The schedulers designed for all groups will be aggregated toward a final scheduler, which makes a decision for polling a subset of vehicles in each global iteration. For the final scheduler, we have the following lemma:

Lemma 7. *For any $\vec{d}(t)$ that can be mapped to an FPD $\vec{\tilde{d}}(t)$ with $\vec{\tilde{d}}(t) \preccurlyeq \vec{d}(t)$ and $l(\vec{\tilde{d}}(t)) \leq M$, EPS can find a feasible scheduler.*

Proof. $\tilde{d}_i$ can be written as $\tilde{d}_i = b/2^{m_i}i$, where $m_i \in \mathbb{N}$, m_i is non-increasing with i, and b is an integer. Equivalently, $1/\tilde{d}_i = 2^{m_i}i/b$. For the first group including vehicle 1 to vehicle k, we have:

$$s \triangleq \sum_{i=1}^{k} 2^{m_i} \tag{7.9a}$$

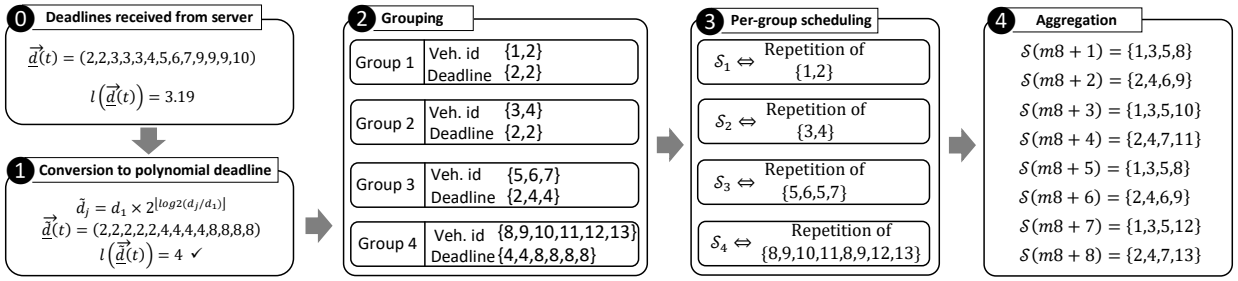


Figure 7.5: An example of using EPS for scheduling. The network includes 13 vehicles with network deadline $\vec{d}(t) = (2, 2, 3, 3, 3, 4, 5, 6, 7, 9, 9, 9, 10)$ and $l(\vec{d}(t)) = 3.19$. Final scheduler is the output of step 4 and it is a cyclic scheduler with cycle length 8.

$$(s - 2^m k)/b \leq 1, \tag{7.9b}$$

$$s/b > 1. \tag{7.9c}$$

(7.9b) and (7.9c) follow from (7.9a) and the definition of a group. Based on (7.9b) and (7.9c), s can be expressed as $s = \lceil b/2^m k \rceil \cdot 2^m k = \lceil \tilde{d}_k \rceil \cdot 2^m k$. Given that $m_i \geq m_k$ for all $i < k$, we have $s \leq \lceil b/2^{m_i} \rceil \cdot 2^{m_i} i = \lceil \tilde{d}_i \rceil \cdot 2^{m_i} i$ for $i < k$. Therefore, we can obtain an FPD $\vec{\tilde{g}}$ with $l(\vec{\tilde{g}}) = 1$ as $\vec{\tilde{g}} = (\tilde{d}_1, \tilde{d}_2, \cdots, \tilde{d}_k) \cdot s/b$. For $\tilde{g}_i$, $i = 1, 2, \cdots, k$, we have $\tilde{g}_i = \tilde{d}_i \cdot s/b \leq \tilde{d}_i \cdot \lceil \tilde{d}_i \rceil \cdot 2^{m_i} i/b = \lceil \tilde{d}_i \rceil \leq d_i$. Therefore, $(d_1, d_2, \cdots, d_k)$ can be mapped to $\vec{\tilde{g}}$ and we can use FSC to find a scheduler for it [95]. We repeat the above procedure up to $(M-1)$ times for remaining groups, and use FSC to find a feasible scheduler for each of them. Then we combine the M different schedulers and get feasible scheduler $\mathcal{S}$.

■

In our example shown in Fig. 7.5, a cyclic scheduler is designed for each group using FSC in [95]. As an instance, for the first group which includes vehicles 1 and 2, the scheduler polls vehicle 1 on every even global iteration and polls vehicle 2 on every odd global iteration.

Aggregation. Once a cyclic scheduler is constructed for each group. In each global iteration, we poll the vehicles specified by each group. Referring to the example in Fig. 7.5, at $t = 3$, the

vehicle selected in the third global iteration are $\mathcal{S}(3) = \{1, 3, 5, 10\}$. For the general case, if the number of groups are less than M, more than one vehicle specified by a scheduler will be selected. For example, if there are two groups and $M = 4$, on each global iteration, two subsequent vehicles will be selected by the scheduler of each group.

Computational Complexity of EPS. The computational complexity of EPS can be attributed to finding FPD and FSC. FSC is called only once when an appropriate FPD is found. Finding an appropriate FPD may go through an iterative removal of vehicles with the shortest deadline to relax the network load. EPS finds FPD over the entire set of vehicles. Based on the computational complexity analysis in [95], the computational complexity of EPS is $\mathcal{O}(N^3(t)) + \mathcal{O}(M d_{max} \log d_{max})$. In a dense vehicular network, the computational complexity of EPS grows polynomially w.r.t. $N(t)$, which is much lower compared to the general scheduler.

When to Invoke EPS. Alg. 7.1 presents a generic scheduler, which can find a feasible cyclic scheduler that may not be realized by EPS. This issue can be intuitively inferred by considering the gap between necessary condition of schedulability (i.e., $\vec{d}(t) \leq M$) and a network load threshold that guarantees existence of at least one FPD (i.e., $\vec{d}(t) \leq M \ln(2)$) [95]. However, in moderate-size or large-size vehicular networks, steady state graphs become too large to store and process. Based on the available computational resources at the server, a threshold is needed to be set on the network size to efficiently switch between EPS and the general scheduler, and to gain a suitable trade-off between performance and complexity.

How DDVS (EPS and General Schedulers) Mitigate Stragglers Effect. After incorporating processing delays into the deadlines, if a straggler exists in the network, the network load drastically increases. When such an increase pushes the network load beyond the threshold of schedulability, DDVS removes the vehicles with the shortest deadline. These vehicles are less-capable vehicles, and likely pose high processing delays. Therefore, DDVS treats the schedulability of the

Algorithm 7.2 Extended polynomial scheduler (EPS).

1: **Input**: The network deadline $\vec{d}(t)$
2: **if** $\vec{d}(t) = \vec{d}(t-1)$ **then**
3: Keep current scheduler
4: **else**
5: $\mathcal{Q} = \mathcal{N}(t)$
6: Sort $\vec{d}(t)$ in increasing order
7: **while** 1 **do**
8: **for** $i = 1, 2, \cdots, |\mathcal{Q}|$ **do**
9: Set $\vec{\tilde{d}}(t) = [d_i 2^{\lfloor \log_2(d_1/d_i) \rfloor}, \cdots, d_i 2^{\lfloor \log_2(d_{|\mathcal{Q}|}/d_i) \rfloor}]$
10: **if** $l(\vec{\tilde{d}}(t)) \leq M$ **then**
11: **goto** line 14
12: $\mathcal{Q} \longleftarrow \mathcal{Q} \backslash \{1\}$
13: Update $\vec{d}(t)$
14: **while** $|\mathcal{Q}| > 0$ **do**
15: **if** $\sum_{i=1}^{|\mathcal{Q}|} 1/\tilde{d}_i > 1$ **then**
16: Find the smallest k such that $\sum_{i=1}^{k} 1/\tilde{d}_i \geq 1$
17: **else**
18: Set $k = |\mathcal{Q}|$
19: For $[d_1, d_2, \cdots, d_k]$, use FSC to find a feasible scheduler and aggregate it into $\mathcal{S}$
20: $\mathcal{Q} \longleftarrow \mathcal{Q} \backslash \{1, 2, \cdots, k\}$
21: Update $\vec{d}(t)$ and $\vec{\tilde{d}}(t)$
22: **Return** $\mathcal{S}$

network by ignoring stragglers with high processing delays.

Summary of EPS. Alg. 7.2 summarizes EPS. It first sorts the vehicles based on their deadline in a non-decreasing order and then maps it to multiple FPDs. If EPS finds an FPD with load less than M, it uses that FPD for scheduling; otherwise, it removes the vehicle with the shortest deadline and repeats this procedure. Once an FPD with a load less than M is found, it partitions the FPD into multiple groups and constructs a scheduler for each of them. The aggregation of schedulers for different groups leads to a desired scheduler.

7.6 Concurrent Vehicle Polling Scheme (CVPS)

Concurrent vehicle polling will significantly improve the FL convergence, and uplink MU-MIMO is an approach to achieving concurrent vehicle polling. While uplink MU-MIMO has been well studied in WiFi and cellular networks, existing techniques are limited to stationary or semi-stationary networks as they assume the perfect time alignment of uplink transmissions. This assumption, however, is not valid in vehicular networks. This is because, while the frequency synchronization can be achieved using GPS or other techniques, the time misalignment of uplink transmissions (caused by signal propagation delay, packet processing delay, clock jitters, etc.) is hard to eliminated in dynamic vehicular networks. To address this issue, we propose an asynchronous uplink MU-MIMO transmission scheme to enable concurrent vehicle polling. It should be noted that the asynchronism CVPS deals with is different from that in asynchronous FL. CVPS deals with the signal-level asynchrony, while FL deals with the message-level asynchrony. In Asynchronous FL, the server receives delayed local models even after the termination of a global iteration. The asynchronism in asynchronous FL is in the order of packets or frames. In contrast, CF4FL deals with the synchronous FL where all local models from the selected vehicles will be received by the server within the corresponding global iteration. The PHY-layer asynchronism in CF4FL is in the order of signal samples.

When DDVS initiates a global iteration, the selected vehicles simultaneously send their local models to the server as shown in Fig. 7.6. The vehicles transmit their local models through multiple frames within a stream. This is because a frame must lie within channel coherence time, which is relatively short in vehicular networks. Per 802.11p standard, each frame comprises a preamble including an L-STF and an L-LTF, an L-SIG, and payload (frame body).

As shown in Fig. 7.7, CVPS employs M antennas of the server to mitigate inter-vehicle in-

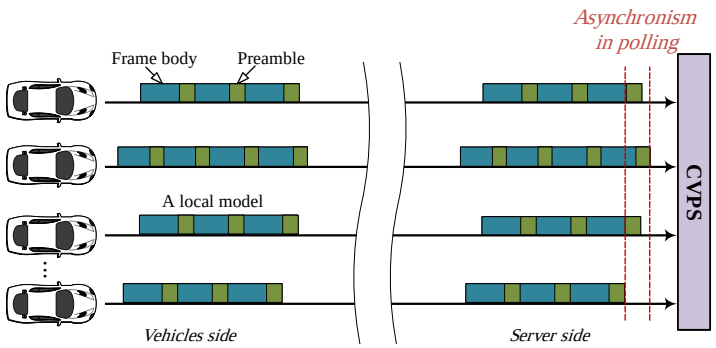


Figure 7.6: Asyncronism in local model transmissions of the selected vehicles.

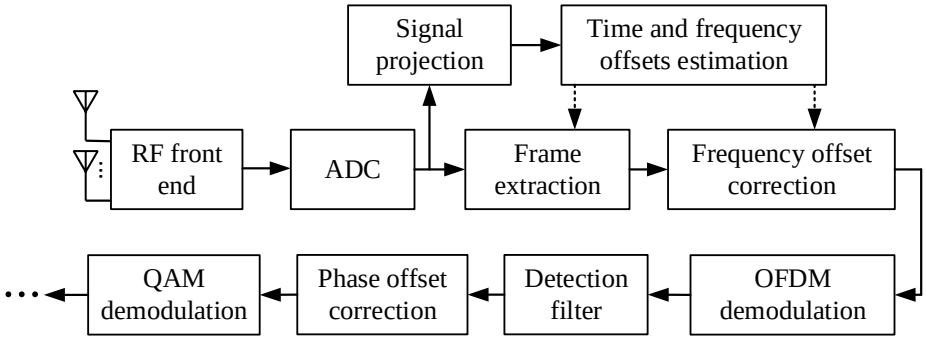


Figure 7.7: PHY-layer structure of CVPS.

terference and recovers all the transmitted frames within streams. To do so, the received signal samples from M antennas first go through the signal projection module, which decomposes the signaling space into M subspaces in the time domain. The projection of signal in each subspace is used for time and frequency synchronization. Once time and frequency offsets are compensated, the signals will be converted to frequency domain using OFDM demodulation. A spatial detection filter is then designed for each interfered frame. The spatial detection filter not only suppresses inter-vehicle interference, but it also equalizes the unknown channel. The recovered frame is demodulated after phase offset compensation. In what follows, we describe the key components of CVPS.

Synchronization via Signal Projection. In the vehicular scenario shown in Fig. 7.6, synchronization of streams is challenging as each stream is polluted by inter-vehicle interference.

To alleviate the interference, we project the time-domain signal samples into M orthogonal subspaces. Let us denote the nth received samples from all antennas by $\mathbf{y}(n) \in \mathbb{C}^{M\times 1}$. The basis of signal subspaces at sampling index n, $\mathbf{B}(n)$, can be calculated through eigenvalue decomposition as follows:

$$[\mathbf{B}(n), \mathbf{\Lambda}(n)] = \text{EVD}\left(\frac{1}{2L_s+1}\sum_{i=n-L_s}^{n+L_s}\left(\mathbf{y}(i)\mathbf{y}(i)^{\mathsf{H}}\right)\right), \tag{7.10}$$

where EVD$(\cdot)$ denotes eigenvalue decomposition, $(\cdot)^{\mathsf{H}}$ is conjugate transpose operation, L_s is an integer number that defines a window length in calculation, $\mathbf{\Lambda}(n) \in \mathbb{C}^{M\times M}$ is a diagonal matrix containing eigenvalues, and $\mathbf{B}(n) \in \mathbb{C}^{M\times M}$ has corresponding eigenvectors with $\mathbf{B}_j(n)$ being its jth columns. $\mathbf{B}_j(n)$ is the base for the jth subspace and can be used to project received signal samples at sampling index n onto subspace j. The projected signals are then used for synchronization. To find the appropriate subspace for a certain stream, we try all subspaces and choose the one with highest cross-correlation peak in time synchronization. That said, if the jth subspace is chosen for stream i, $\tilde{y}_i(n) \triangleq \mathbf{B}_j(n)^{\mathsf{H}}\mathbf{y}(n)$ is employed for time and frequency offset compensations of stream i. The beginning of a frame in stream i is then calculated w.r.t. the peak of correlation between LTF waveform used by vehicle i and $\tilde{y}_i$. Also, the carrier frequency offset is computed by $\theta_i = 1/K \cdot \angle(\sum_{n=n_0}^{n=n_0+K-1} \tilde{y}_i(n)\tilde{y}_i(n+K)^{\mathsf{H}})$, where $\angle(\cdot)$ is the angle of a complex number, K is the FFT size, and n_0 is the position of the first LTF sample in a frame. The calculated offset is corrected before further processing.

Spatial Detection Filter. The synchronized signals are first translated into the frequency domain by OFDM demodulation. Let us focus on the first frame of stream i coming from vehicle i.

In the frequency domain, the received signal can be written as:

$$\mathbf{Y}(l,k) = \mathbf{h}_{ui}(k)x_i(l,k) + \sum_{j\in\mathcal{S}(t),j\neq i} (\mathbf{h}_{uj}(k)x_j(l,k)), \tag{7.11}$$

where $\mathbf{Y}(l,k) \in \mathbb{C}^{M\times 1}$ and $x_i(l,k) \in \mathbb{C}$ are the received signal at the server and the transmitted signal from vehicle i on subcarrier k and sample l, respectively. Also, $\mathbf{h}_{ui}(k) \in \mathbb{C}^{M\times 1}$ denotes the channel from vehicle i to the server on subcarrier k. Although the channel gain may vary over the stream, it is assumed to be unchanged over one frame. For recovering a frame in stream i, we particularly look for filter $\mathbf{P}(k) \in \mathbb{C}^{M\times 1}$ that nullifies $\sum_{j\in\mathcal{S}(t),j\neq i} \mathbf{h}_{uj}(k)x_j(l,k)$ and equalizes the effect of channel $\mathbf{h}_{ui}(k)$. The filter can be constructed as:

$$\mathbf{P}(k) = \left[\sum_{(l,k')\in\mathcal{R}_{ik}} \mathbf{Y}(l,k')\mathbf{Y}(l,k')^{\mathsf{H}}\right]^{-1}\left[\sum_{(l,k')\in\mathcal{R}_{ik}} \mathbf{Y}(l,k')R_i(l,k')^{\mathsf{H}}\right], \tag{7.12}$$

where $R_i(l,k')$ is the reference signal on sample l and subcarrier k' of the preamble used by vehicle i and $\mathbf{Y}(l,k')$ denotes the corresponding received signal samples over all the antennas. $\mathcal{R}_{ik}$ is the set of reference signal samples located within a pre-defined sliding window around subcarrier k. With this filter, interference mitigation and channel equalization can be achieved through $\hat{x}_i(l,k) = \mathbf{P}(k)^{\mathsf{H}}\mathbf{Y}(l,k)$, where $\hat{x}_i(l,k)$ is the estimated signal symbol. (7.12) suggests that the design of $\mathbf{P}(k)$ is not reliant on CSI and it only needs pre-known reference signal samples in the preamble of desired frame, which is the case for L-LTF and L-STF samples in IEEE 802.11p.

Computational complexity of Filter Design. The computational complexity of designing a spatial filter is independent of the size of vehicular networks since at most M vehicles will be polled in each global iteration. The design of such a filter requires matrix multiplication, addition, and inversion. The overall computational complexity of designing a spatial filter is $\mathcal{O}(N_{sc}M^3)$,

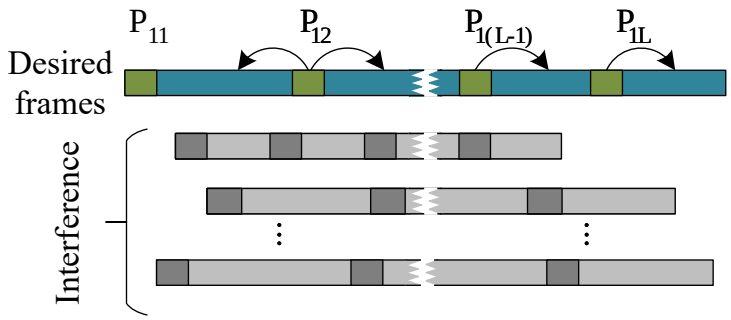


Figure 7.8: Illustrating the idea of CVPS.

where N_{sc} is the number of subcarriers.

Mitigating Preamble Misalignment. The detection filter in (7.12) can remove unintended streams if those streams interfere with the preamble of the desired frame. This requirement cannot be met at the first frame of each stream due to the lack of network-wide synchronization. As an example, consider the transmitted streams shown in Fig. 7.8, where the preamble of the first frame from stream 1 is not collided with stream 2. If the reference signal samples in the preamble are leveraged to design a detection filter like P_{11}, this filter cannot mitigate the interference caused by stream 2. To address this issue, we do not use the preamble of the first frame for filter design. Instead, once the filter P_{12} is designed for the second frame, it is used for both first and second frames. Here, we have assumed that time misalignment does not exceed the length of a frame. It is worth noting that time misalignment is not a challenge for the frames located on the tail of streams. This is because a non-interfering stream at the preamble will not interfere with the rest of the frame. Therefore, starting from the second frame, the detection filter leverages the reference signal samples to recover the frame's body within the same frame as shown in Fig. 7.8 for frames 2 to L.

7.7 Performance Evaluation

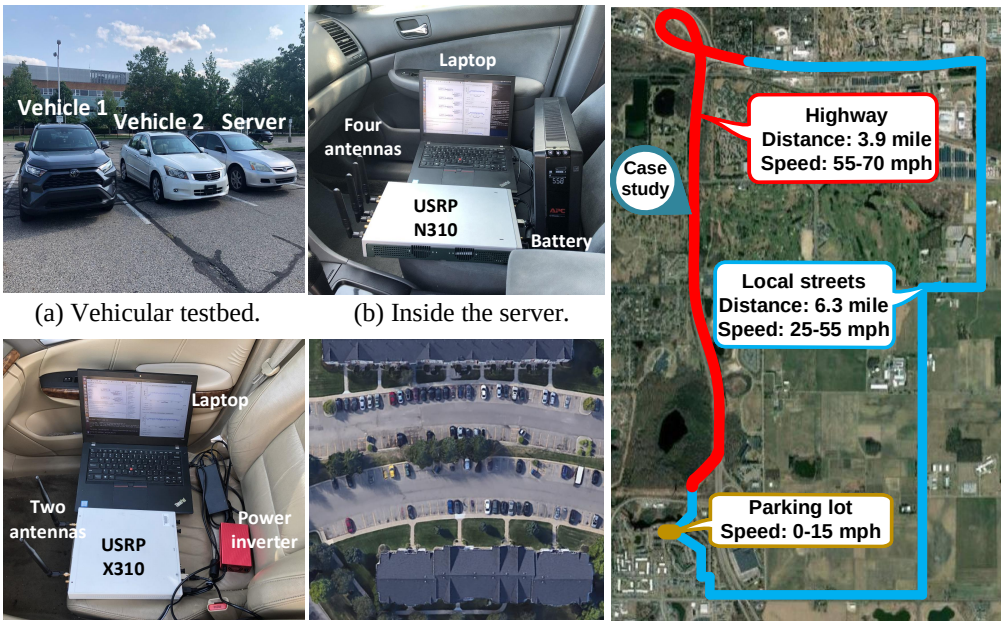
In this section, we evaluate the performance of CF4FL and its two components (DDVS and CVPS) using experiments and trace-driven simulation.

7.7.1 Evaluation Methodology

We first implement CVPS on a wireless vehicular testbed and investigate its performance on parking lots, local streets, and highways. The measurement results will be used to simulate vehicle polling in large vehicular networks with $N(t) = 5 \sim 25$ vehicles. In our simulation, DDVS uses the general scheduler if $N(t) \leq 8$ and EPS otherwise. Through trace-driven simulation, we then evaluate CF4FL in dynamic vehicular networks of different sizes.

Vehicular Testbed. Fig. 7.9 shows our small-size vehicular testbed used to evaluate CVPS. It has three vehicles: one acts as the server, and the other two act as four virtual vehicles. The server is implemented using a USRP N310 radio with four antennas ($M = 4$) for the transmission/reception of RF signals, a ThinkPad T480 with Quad-Core i5-8250U CPU for baseband signal processing, and an APC 1500VA UPS battery as shown in Fig. 7.9(b). Each of the two client vehicles carries a USRP X310 device, a ThinkPad T480 with Quad-Core i5-8250U CPU, and a BESTEK 300W power inverter as shown in Fig. 7.9(c). Since USRP X310 has two independent RF chains, we use the two client vehicles to emulate four client vehicles, each of which has one antenna for radio signal transmission and reception.

Experimental Route. We evaluate CVPS using sequential polling (i.e., single-user MIMO) as the comparison baseline in three scenarios: a parking lot as shown in Fig. 7.9(d) at $0 \sim 15$ mph speed, local streets at $25 \sim 55$ mph speed on 6.3 miles, and a highway at $55 \sim 70$ mph speed on



(a) Vehicular testbed. (b) Inside the server. (c) Inside a vehicle. (d) Parking lot. (e) Experimental routes

Figure 7.9: Experimental scenarios and our vehicular testbed for evaluating polling approaches.

3.9 miles, as shown in Fig. 7.9(e). The two client vehicles keep staying within $50 \sim 300$ ft distance from the server during several laps on the experimental route.

Trace-Driven Simulation. We simulate CF4FL for large networks with different numbers of vehicles based on our collected experimental results. Specifically, we assume a network with size (number of vehicles) $N(0)$ at the beginning, where a vehicle can join/leave the network based on the arrival/leave global iterations drawn from Poisson distribution with parameter λ. In our simulation, we let $N(0) \in \{10, 15, 20\}$ and $\lambda \in \{0.02, 0.04\}$, with the same probability for a vehicle joining and leaving the network. In a simulated network, each vehicle has an integer deadline drawn from uniform distribution between 2 to 10. Also, we assume a vehicle collects a batch of data during each global iteration.

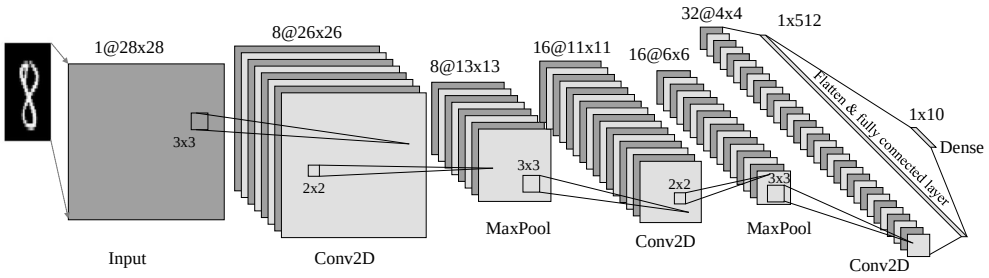


Figure 7.10: CNN-based FL application for digit classification.

7.7.2 FL Task

As a case study, we use FL to classify images of digits 0 to 9 in our evaluation. The digit classification is a useful tool in vehicular environments for different purposes, such as recognizing road sign of speed limit, identifying the information on traffic sign, recognizing clearance limits, weight limits, etc.

Dataset. We use MNIST dataset. It includes 70,000 images of handwritten digits, where 60,000 are used for training and 10,000 for test. Each image has 28×28 pixels and labeled with a number from 0 to 9. In our experiments, the dataset is partitioned among vehicles in an iid manner.

Neural Network Architecture. We use a Convolutional Neural Network (CNN) as shown in Fig. 7.10 to perform the desired FL task. The input is 28×28 pixels. The first 2D convolutional layers are followed by batch normalization, ReLu, and max pooling layers. The outputs of the last 2D convolutional layers are flattened and then flowed into a dense layer. A softmax layer is applied to the output of dense layer to represent the predicted digit.

Training and Convergence. For training digit classifier, the learning rate is set to 0.001 and it is not decayed as the data samples in vehicles will be discarded after consumption (past, current, and future data samples are equally valuable). The number of global iterations is also not preset. We assume that the global model converges when the classification accuracy change of two consecutive global iterations is less than 0.1%.

Benchmarks. For DDVS, we employ the following three schedulers as the performance benchmark.

- *Random Scheduler (RND)*: At each global iteration, RND scheduler selects M vehicles among $N(t)$ vehicles with equal probabilities. The selection is performed regardless of vehicles status (i.e., $\vec{d}(t)$ and $\vec{p}(t)$).

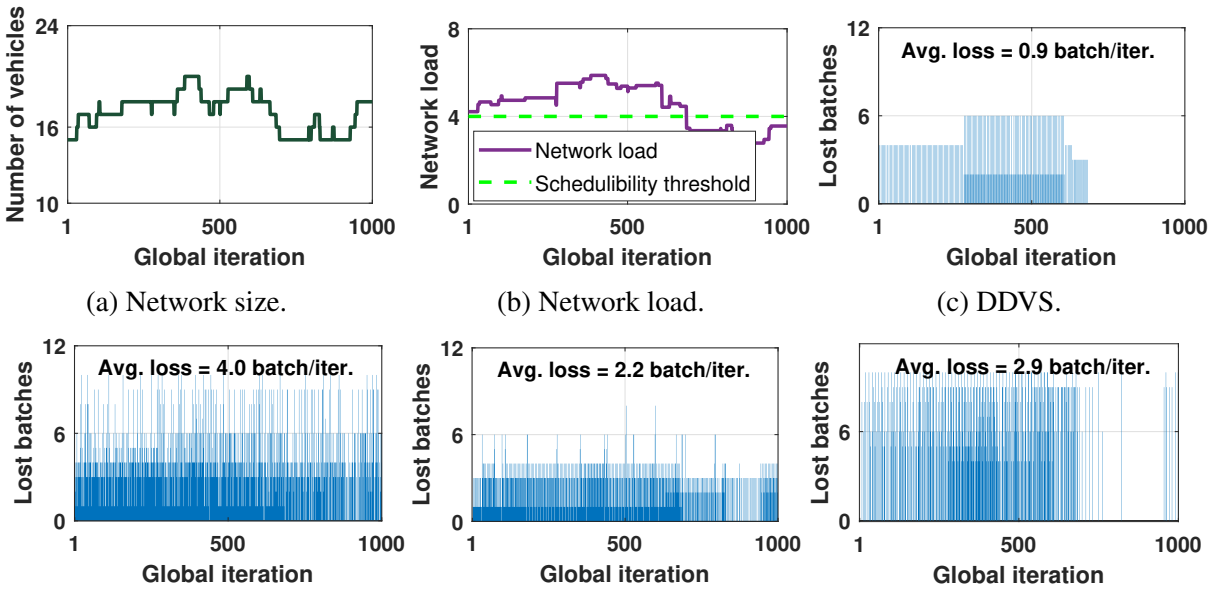
- *Round-Robin Scheduler (RR)*: The RR scheduler is essentially a cyclic time-sharing scheduler. It selects M vehicles in each global iteration such that in a large number of global iterations, all the vehicles are polled with equal probability.

- *Earliest Deadline First Scheduler (EDF)*: In each global iteration, EDF scheduler selects M vehicles that are closest to their corresponding deadline (i.e., M vehicles corresponding to M smallest elements in $\vec{d}(t) - \vec{p}(t)$). If more than M vehicles are found with the same closeness, the ones with more data samples are selected.

For CVPS, we employ Sequential Polling (SP) as the performance baseline. While CVPS polls M vehicles in a global iteration concurrently, SP polls M vehicles sequentially in a global iteration. Finally, for CF4FL, we combine the benchmark schedulers with SP to provide three benchmarks: RND+SP, RR+SP, EDF+SP.

7.7.3 Performance of DDVS

A Case Study. We simulate a vehicular network starting with $N(0) = 15$ vehicles at the first global iteration. Vehicles join or leave the network at the beginning of each global iteration, following a Poisson distribution with $\lambda = 0.02$. An instance of such a network is shown in Fig. 7.11(a). The network size varies between 15 to 20 vehicles during $1,000$ global iterations. The shortest and



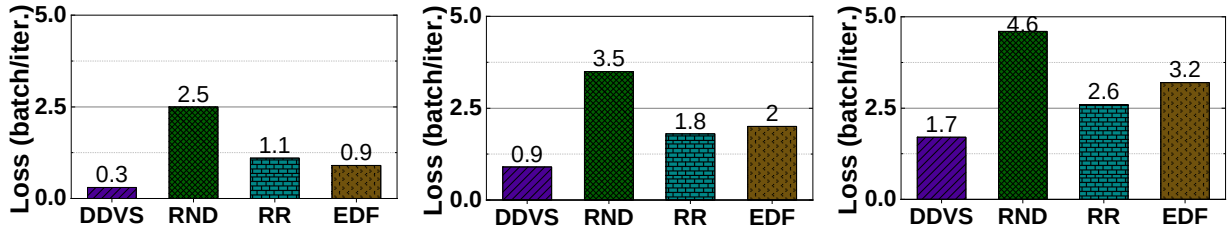
(a) Network size. (b) Network load. (c) DDVS.

(d) RND scheduler. (e) RR scheduler. (f) EDF scheduler.

Figure 7.11: Data loss of different schedulers for a vehicular network with $N(0) = 15$ and $\lambda = 0.02$.

longest interval between two subsequent changes are 1 and 97 global iterations. The network load also varies between 2.8 to 5.9 as shown in Fig. 7.11(b). Also, we assume $M = 4$ and, therefore, the necessary condition for the schedulibilty of network load is $l(\vec{d}(t)) \leq 4$. We leverage all benchmark schedulers along with DDVS on this network and measure the lost batches of data in each iteration. The maximum data loss is 6, 10, 8, and 10 batches in an iteration for DDVS, RND, RR, and EDF schedulers, respectively. The average data loss of DDVS, RND, RR, and EDF schedulers is 0.9, 4.0, 2.2, and 2.9 batch per global iteration, respectively.

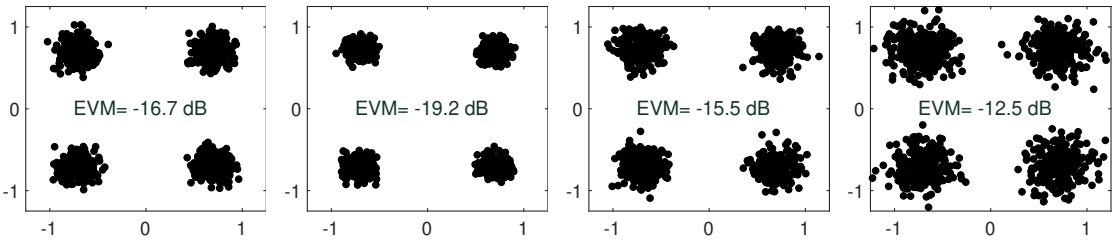
Extensive Simulation. By the same token, we repeat the above study through extensive simulation to measure the gain of DDVS scheduler in different network sizes and different network dynamics. Fig. 7.12 presents the loss of data, from which we have following observations: i) compared to RND scheduler over all cases, DDVS reduces the data loss by 76.1% on average; ii) compared to RR scheduler, DDVS reduces the data loss by 53.9% on average; and iii) compared to EDF scheduler, DDVS reduces the data loss by 59.0% on average.



(a) $N(0) = 10$ and $\lambda = 0.02$. (b) $N(0) = 15$ and $\lambda = 0.02$. (c) $N(0) = 20$ and $\lambda = 0.02$.

(d) $N(0) = 10$ and $\lambda = 0.04$. (e) $N(0) = 15$ and $\lambda = 0.04$. (f) $N(0) = 20$ and $\lambda = 0.04$.

Figure 7.12: Data loss of DDVS, RND, RR, and EDF schedulers in different vehicular networks.



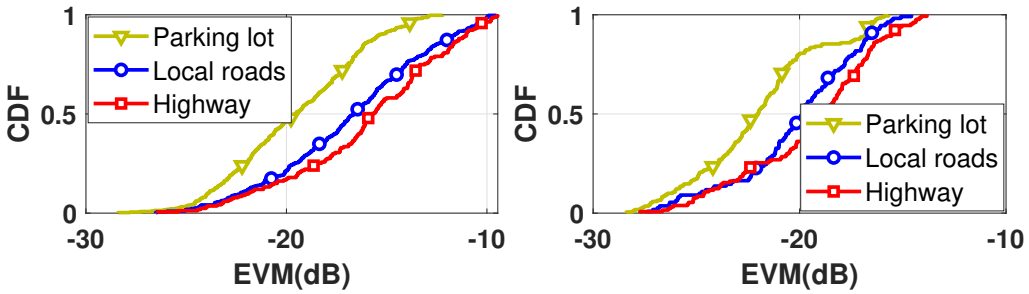
(a) Stream 1. (b) Stream 2. (c) Stream 3. (d) Stream 4.

Figure 7.13: The EVM performance of CVPS when decoding four concurrent data packets.

7.7.4 Performance of CVPS

A Case Study. The case study is conducted at the location marked in Fig. 7.9(a). We first conduct sequential polling and send a stream from each antenna of vehicles 1 and 2 one by one. The error vector magnitude (EVM)[2] of decoded signals are -19.8 dB, -20.0 dB, -19.1 dB, and -17.3 dB. The average data rate achieved by sequential polling is interpolated to 11 Mbps. We then conduct CVPS, which concurrently polls four streams from vehicles 1 and 2. The constellations of first decoded frame in all streams are shown in Fig. 7.13. The EVM of decoded frames is

[2]EVM is calcuated by $\text{EVM} = 10\log_{10}(\frac{\mathbb{E}[|\hat{S}(l,k)-S(l,k)|^2]}{\mathbb{E}[|S(l,k)|^2]})$, where $\hat{S}(l,k)$ and $S(l,k)$ are the lth estimated and original modulated symbols on the kth subcarrier, respectively.



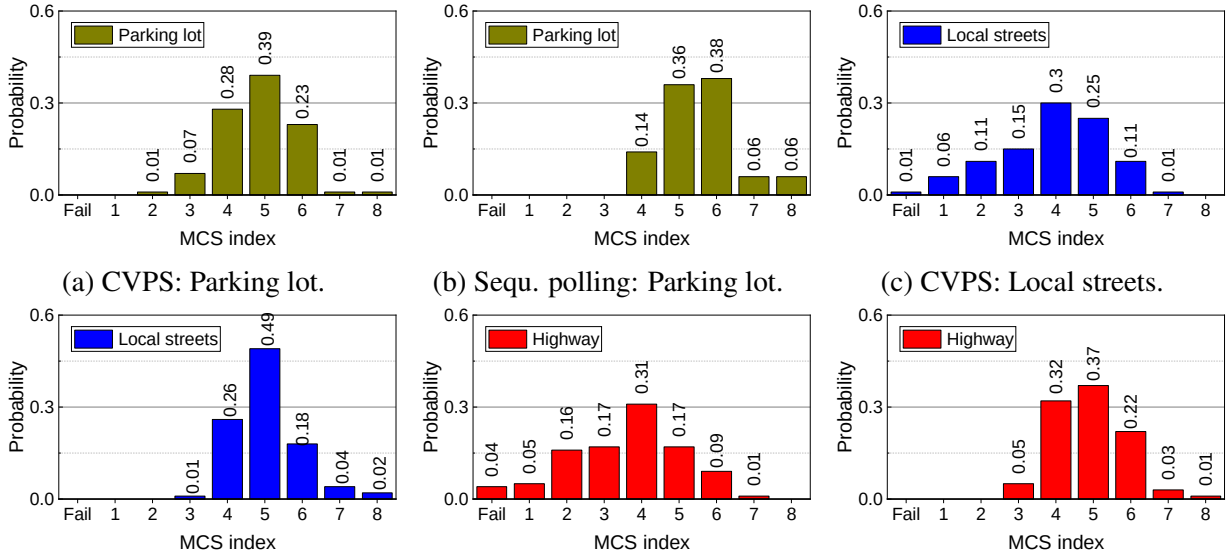
(a) CVPS. (b) Sequential polling.

Figure 7.14: EVM of decoded frames via CVPS and sequential polling in parking lot, local streets, and highway.

-16.7 dB, -19.2 dB, -15.5 dB, and 12.5 dB. Collectively, CVPS yields 34 Mbps data rate. As a global iteration includes the polling of $M = 4$ vehicles in uplink and a broadcast in downlink, CVPS reduces the time consumption of a global iteration by $2.2\times$ compared to sequential polling.

Extensive Experiments. We perform extensive experiments to measure the EVM of decoded signals polled by CVPS and sequential approaches at parking lot, local streets, and highway. The cumulative distribution function (CDF) of measured EVMs is illustrated in Fig. 7.14. The average EVM of decoded signals with CVPS is -19.5 dB, -16.8 dB, and -15.9 dB at the parking lot, local streets, and highway, respectively. The average EVM of decoded signals with sequential polling is -22.0 dB, -20.2 dB, and -19.4 dB at parking lot, local streets, and highway, respectively. Apparently, CVPS has a slight EVM degradation compared to sequential polling. This degradation is caused by the residual inter-vehicle interference of concurrent transmissions.

Fig. 7.15 presents probability of MCS selection for uplink transmissions in both CVPS and sequential polling. It shows that CVPS causes 1% and 4% data packet loss in local streets and highway, respectively. In contrast, no loss is observed for sequential polling. This is because sequential polling selects a single vehicle for uplink transmissions+, while CVPS selects four vehicles for concurrent uplink transmissions. The data packet loss is caused by the poor uplink channel. Fig. 7.16 shows the data rate achieved by two polling strategies. It is proportional to the data rate



(a) CVPS: Parking lot. (b) Sequ. polling: Parking lot. (c) CVPS: Local streets.

(d) Sequ. polling: Local streets. (e) CVPS: Highway. (f) Sequ. polling: Highway.

Figure 7.15: Comparison of CVPS and sequential polling in terms of MCS selection probability.

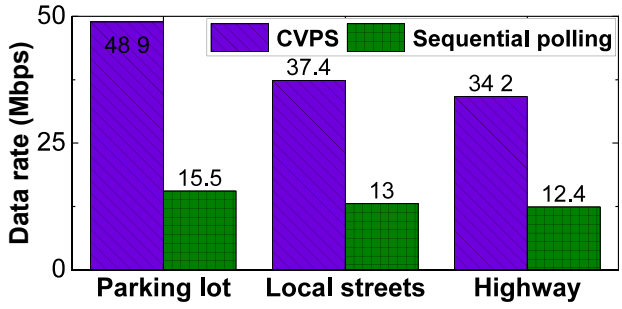
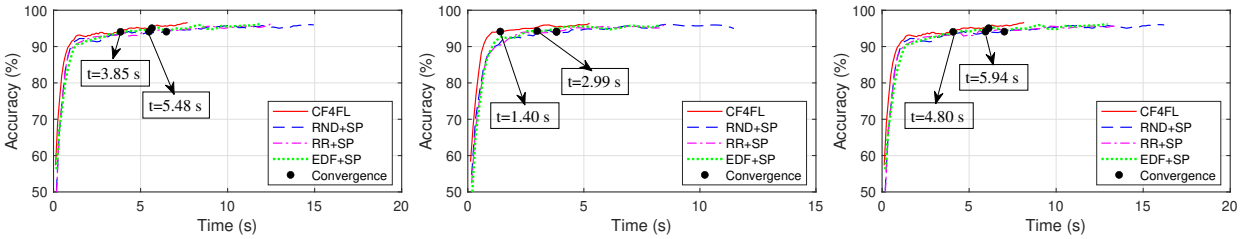


Figure 7.16: Data rate achieved by CVPS and sequential polling.

of local model polling. Evidently, CVPS offers a much higher data rate for local model polling in the uplink, thereby shortening the time consumption of each global iteration. CVPS alone reduces the duration of a global iteration by 58.3%, 52.4%, and 52.4% in a parking lot, local streets, and highways, respectively.

7.7.5 Performance of CF4FL (DDVS + CVPS)

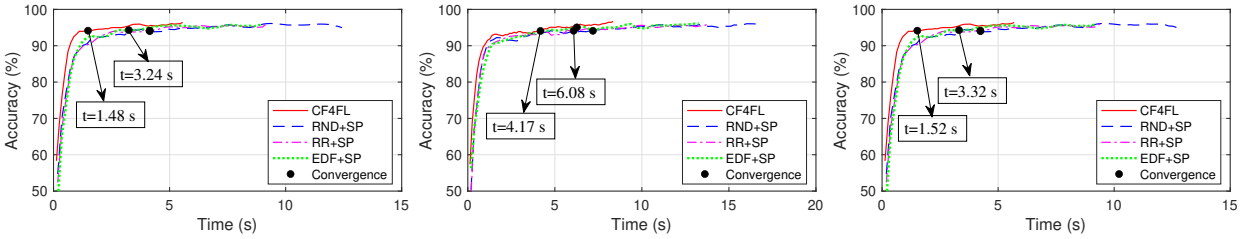
Finally, we evaluate the performance of CF4FL by comparing it with RND+SP, RR+SP, and EDF+SP benchmarks in two cases: i) $N(0) = 10$ and $\lambda = 0.02$, and ii) $N(0) = 20$ and $\lambda = 0.04$.



(a) Parking lot: $N(0) = 10$ and $\lambda = 0.02$.

(b) Parking lot: $N(0) = 20$ and $\lambda = 0.04$.

(c) Local streets: $N(0) = 10$ and $\lambda = 0.02$.



(d) Local streets: $N(0) = 20$ and $\lambda = 0.04$.

(e) Highway: $N(0) = 10$ and $\lambda = 0.02$.

(f) Highway: $N(0) = 20$ and $\lambda = 0.04$.

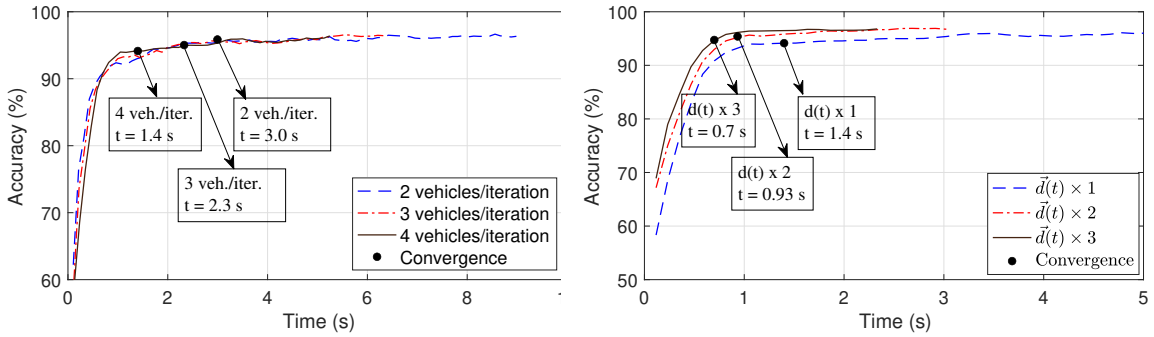
Figure 7.17: Convergence of CF4FL and benchmark approaches in different vehicular scenarios.

Two cases are simulated in the three environments (parking lot, local streets, and highways), and a total of six scenarios are evaluated. The performance of CF4FL and its benchmarks are presented in Fig. 7.17. And we have the following observations.

- **FL Convergence Speed.** On average over all scenarios, CF4FL reduces the convergence time by 48.2%, 34.9%, and 35.3% compared to RND+SP, RR+SP, and EDF+SP, respectively.
- **Data Collection Speed.** As shown in Fig. 7.17(a)-(f), CF4FL obtained 60,000 data samples in a shorter period of time compared to the benchmarks. On average, data collection speed of CF4FL is $2.2\times$, $1.8\times$, and $1.7\times$ faster than RND+SP, RR+SP, and EDF+SP, respectively.

7.7.6 Effect of Vehicle Selection and Deadlines on Learning

To determine how the number of selected vehicles per global iteration and vehicles' deadlines affect FL training, we have considered two additional test scenarios on a vehicular network at a



(a) Impact of number of selected vehicles.

(b) Impact of scaling deadlines

Figure 7.18: Effect of vehicle selection and deadlines on FL training.

parking lot with $N(0) = 20$, $\lambda = 0.04$, and a four-antenna server. First, we investigate the effect of selected vehicles by putting an intentional limit on the number of selected vehicles per global iteration. In the second scenario, while four vehicles are selected per iteration, the deadlines are scaled by a factor of $1, 2$, and 3. As shown in Fig. 7.18(a), increasing the number of selected vehicles (up to $M = 4$) per global iteration will accelerate the learning process. When the server selects 2, 3, and 4 vehicles per iteration, the global model converges within 3.0s, 2.3s, and 1.4s. Referring to Fig. 7.18(b), doubling and tripling the deadlines reduce the convergence time by 23.3% and 53.3%, respectively.

7.8 Chapter Summary

In this paper, we studied vehicle scheduling and polling problems associated with FL in vehicular networks, with the aim of accelerating the convergence speed of FL training process. To tackle the above two problems, we presented CF4FL, a vehicular communication framework for FL training process. CF4FL comprises two complementary components, namely DDVS and CVPS. DDVS is a scheduler for each global iteration of FL, which reduces data loss under deadline constraints. CVPS takes advantage of multiple antennas at the server to enable concurrent local model polling,

thereby significantly reducing the time duration of each global iteration and leading to a faster convergence of FL. We have evaluated CF4FL through a blend of experimentation and trace-driven simulation. Our results show that CF4FL reduces the convergence time by 39.4%, and it collects data samples $1.9\times$ faster than existing solutions in parking lots, local streets, and highways.

Chapter 8

Summary and Outlook

In this thesis proposal, we strode to bring intelligence and efficiency to the next-generation wireless networks. We leveraged communication frameworks, artificial intelligence, and synergies between them to take most out of limited communication resources and improve both learning process and performance of wireless networks. We plan to continue our research to develop more practical solutions for wireless communication networks. In the following, we first summarize our past efforts and then explain open problems in intelligent networking we intend to purse.

8.1 Summary

Efficiency. In this thesis, we first proposed new communication frameworks targeting improvement of spectral efficiency and throughput. Specifically, we proposed a spectrum sharing scheme to enable two uncoordinated networks concurrently utilize the available spectrum. We designed two blind interference cancellation techniques to mitigate co-channel interfereg in-band underlay D2D communications in cellular networks. nce and establish an underlay spectrum sharing mechanism for the two networks. We further extended our idea for enablin We then shifted our focus to re-visit multiple access in a single WLAN. We proposed a downlink NOMA scheme to enhance connectivity and throughput of WLANs. Our scheme benefits from new pre-coder and SIC techniques to improve per-group throughput and a new user grouping approach to improve network

throughput.

Intelligence. With the aid of recent advances in artificial intelligence, we proposed two techniques for WLANs. We proposed LB-SciFi, which uses DNNs for frequency-domain compression of CSI required for downlink MU-MIMO in WLANs. Reducing the airtime overhead caused by CSI acquisition, LB-SciFi significantly increases the net throughput achieved by MU-MIMO in WLANs. We also proposed DeepMux, which puts a solid step toward implementation of downlink MU-MIMO-OFDMA mode in future WLANs. DeepMux is equipped with two DNNs, one for interpolating CSI reports which are already sparsified by users, another for accelerating resource allocation at AP. We finally presented CF4FL to put a concrete step forward deployment of FL in a challenging environment like vehicular networks. CF4FL benefits from deadline-driven scheduler which aims to reduce data sample loss in vehicle, thereby accelerating convergence of global ML model. CF4FL further enables asynchronous concurrent transmissions for polling vehicles which reduces time period of global iteration. CF4FL, in brief, improves the learning efficiency of FL.

8.2 Future Focus

As our future research efforts mainly target bringing intelligence to wireless communication networks, we describe a number of challenging issue ahead of practical deployment and broadening the scope of our proposed solutions in this thesis as follows.

Proprietary Solutions.: While DeepMux and LB-SciFi have centrally trained DNN-AEs for CSI compression, the trained model is intrinsically a two-sided model which will be deployed partially at both users and the AP. This centralized training and decentralized deployment are not straightforward as users and AP are usually from different vendors, each of which may prefer to use its own proprietary encoder/decoder. Eaither users' vendors or APs' vendors may be unwilling

to transfer a part of their models. How to keep encoder(s) at user(s) and decoder(s) at AP, yet perform a successful training without model transfer, is a challenging task in practice.

Multi-vendor Training Collaboration. In a real wireless ecosystems, users and APs from different vendors form a wireless network. For CSI compression, how to enable an AP to train one decoder in conjunction with multiple encoders is a challenging task.

Model Monitoring. For model monitoring in CSI compression, either users need to send raw CSI to the AP so that AP can calculate CSI reconstruction accuracy or the AP sends its decoder to users so that users can monitor performance of entire DNN-AE. Neither of these approaches are practical. First, AP may refuse to reveal its decoder to the other parties, let alone model exchange overhead. Second, monitoring is a continuous process. If the CSI samples need to be sent for the AP, monitoring inflicts a huge airtime overhead to the wireless network. Non-input-based monitoring is required to isolate the monitoring process at one side (either users or AP).

Quantization-Aware training. LB-SciFi and DeepMux apply post-training quantization for exchanging CSI feedback which is not an efficient solution since ML models have never been exposed to the quantization. The optimal solution is incorporation of quantization in the training process so that the core architecture of ML models learn to mitigate quantization error. Unfortunately, quantization is a non-differentiable function which disrupts the back propagation and model updates during the training process. It is necessary to investigate quantization-aware training to improve the performance of ML models in inference phase.

Quantization misalignment. Quantization misalignment arises when encoders and decoders from different AP and users vendors are trained to work with different quantization methods. As such, in the inference stage, the quantization and dequantization methods are not necessarily matched. How to make the models robust against such a misalignment is a challenging but necessary research direction. Novel solutions are required to enable reducing the burden of data

collection in current wireless technologies.

Low-Overhead Data Collection. ML models for CSI compression usually are trained at the AP/BS; however, users have access to CSI measurements. Due to excessive size of data samples in scale, it is not possible to continuously transfer data from users to the AP/BS. Also, relying on channel reciprocity for collecting high resolution CSI samples is not always possible.

Fairness of FL in heterogeneous ITS. CF4FL has assumed the vehicular environment is rich of data samples, so that the vehicles can continuously collect i.i.d data samples. While CF4FL is the first of its kind focusing on vehicular environment, it will be more practical to extend its scope to heterogeneous environment from data collection perspective. It is challenging to train a fair ML model, equally works well for all vehicles, in such an environment using FL.

BIBLIOGRAPHY

[1] 3GPP TS 38.211. NR; physical channels and modulation. *3rd Generation Partnership Project; Technical Specification Group Radio Access*, 2017.

[2] 3GPP. RP-151114; LTE-WLAN radio level integration and interworking enhancement. Jun. 2015.

[3] Ali Afana, Vahid Asghari, Ali Ghrayeb, and Sofiene Affes. On the performance of cooperative relaying spectrum-sharing systems with collaborative distributed beamforming. *IEEE Transactions on Communications*, 62(3):857–871, 2014.

[4] Ali Afana, Telex MN Ngatched, and Octavia A Dobre. Cooperative AF relaying with beamforming and limited feedback in cognitive radio networks. *IEEE Communications Letters*, 19(3):491–494, 2015.

[5] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 440–445, 2017.

[6] Mohannad H Al-Ali and KC Ho. Transmit precoding in underlay MIMO cognitive radio with unavailable or imperfect knowledge of primary interference channel. *IEEE Transactions on Wireless Communications*, 15(8):5143–5155, 2016.

[7] Haitham Al-Obiedollah, Kanapathippillai Cumanan, Jeyarajan Thiyagalingam, Alister G Burr, Zhiguo Ding, and Octavia A Dobre. Energy efficient beamforming design for MISO non-orthogonal multiple access systems. *IEEE Trans. Commun.*, 67(6):4117–4131, June 2019.

[8] Abdulrahman Alabbasi, Zouheir Rezki, and Basem Shihada. Energy efficiency and SINR maximization beamformers for spectrum sharing with sensing information. *IEEE Transactions on Wireless Communications*, 13(9):5095–5106, 2014.

[9] Ruhallah AliHemmati, Min Dong, Ben Liang, Gary Boudreau, and S Hossein Seyedmehdi. Multi-channel resource allocation toward ergodic rate maximization for underlay device-to-device communications. *IEEE Trans. Wireless Commun.*, 17(2):1011–1025, 2018.

[10] Ruhallah AliHemmati, Ben Liang, Min Dong, Gary Boudreau, and S Hossein Seyedmehdi. Power allocation for underlay device-to-device communication over multiple channels. *IEEE Trans. Signal Inf. Process. Netw.*, 4(3):467–480, 2018.

[11] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Math. program.*, 95(1):3–51, 2003.

[12] Saidhiraj Amuru. Beam learning–using machine learning for finding beam directions. *arXiv preprint arXiv:1906.04368*, 2019.

[13] M Andersen, J Dahl, and L Vandenberghe. CVXOPT: Python software for convex optimization, version 1.1, 2015.

[14] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.

[15] Simin Badri and Mehdi Rasti. Interference management and duplex mode selection in in-band full duplex D2D communications: A stochastic geometry approach. *IEEE Trans. Mobile Comput.*, 2020.

[16] Dmitry Bankov, Andre Didenko, Evgeny Khorov, and Andrey Lyakhov. OFDMA uplink scheduling in ieee 802.11 ax networks. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 1–6, 2018.

[17] Oscar Bejarano, Eugenio Magistretti, Omer Gurewitz, and Edward W Knightly. MUTE: sounding inhibition for MU-MIMO WLANs. In *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 135–143, 2014.

[18] Anass Benjebbour and Yoshihisa Kishiyama. Combination of NOMA and MIMO: Concept and experimental trials. In *Proc. Int. Conf. Telecommun.*, pages 433–438, 2018.

[19] Anass Benjebbour, Yoshihisa Kishiyama, Yukihiko Okumura, Chien-Hwa Hwang, and I-Kang Fu. Outdoor experimental trials of advanced downlink NOMA using smartphone-sized devices. In *Proc. IEEE VTC-Spring*, pages 1–6, 2018.

[20] Anass Benjebbour, Anxin Li, Keisuke Saito, Yuya Saito, Yoshihisa Kishiyama, and Takehiro Nakamura. NOMA: From concept to standardization. In *Proc. IEEE Conf. Stand. Commun. Netw.*, pages 18–23, 2015.

[21] Anass Benjebbour, Keisuke Saito, Anxin Li, Yoshihisa Kishiyama, and Takehiro Nakamura. Non-orthogonal multiple access (NOMA): Concept, performance evaluation and experimental trials. In *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, pages 1–6, 2015.

[22] Panagiotis Botsinis, Dimitrios Alanis, Zunaira Babar, Soon Xin Ng, and Lajos Hanzo. Iterative quantum-assisted multi-user detection for multi-carrier interleave division multiple access systems. *IEEE Transactions on Communications*, 63(10):3713–3727, 2015.

[23] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge Univ. Press, Cambridge, U.K., 2004.

[24] Sheng-Ming Cai and Yi Gong. Cognitive beamforming for throughput maximization with statistical cross channel state information. *IEEE Communications Letters*, 18(11):2031–2034, 2014.

[25] Y. Cao, N. Zhao, Y. Chen, M. Jin, L. Fan, Z. Ding, and F. R. Yu. Privacy preservation via beamforming for NOMA. *IEEE Trans. Wireless Commun.*, 18(7):3599–3612, July 2019.

[26] Haoye Chai, Supeng Leng, Yijin Chen, and Ke Zhang. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] Chao-Yu Chen, Chi-An Sung, and Hsiao-Hwa Chen. Capacity maximization based on optimal mode selection in multi-mode and multi-pair D2D communications. *IEEE Trans. Veh. Technol.*, 68(7):6524–6534, 2019.

[28] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. Performance optimization of federated learning over wireless networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[29] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 2020.

[30] Yali Chen, Bo Ai, Yong Niu, Ke Guan, and Zhu Han. Resource allocation for device-to-device communications underlaying heterogeneous cellular networks using coalitional games. *IEEE Trans. Wireless Commun.*, 17(6):4163–4176, 2018.

[31] Zengmao Chen, Cheng-Xiang Wang, Xuemin Hong, John Thompson, Sergiy A Vorobyov, Feng Zhao, and Xiaohu Ge. Interference mitigation for cognitive radio MIMO systems based on practical precoding. *Physical communication*, 9:308–315, 2013.

[32] Zhiyong Chen, Zhiguo Ding, Peng Xu, and Xuchu Dai. Optimal precoding for a QoS optimization problem in two-user MISO-NOMA downlink. *IEEE Commun. Lett.*, 20(6):1263–1266, 2016.

[33] Junsu Choi, Sunghyun Choi, and Kwang Bok Lee. Sounding node set and sounding interval determination for IEEE 802.11ac MU-MIMO. *IEEE Transactions on Vehicular Technology*, 65(12):10069–10074, 2016.

[34] Suren Dadallage, Changyan Yi, and Jun Cai. Joint beamforming, power, and channel allocation in multiuser and multichannel underlay MISO cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 65(5):3349–3359, 2016.

[35] DataReportal. Digital Around the World. *DataReportal*. [Online]. Available: https://datareportal.com/global-digital-overview, [Accessed: 17-Sep-2021].

[36] Soumendra Nath Datta and Suresh Kalyanasundaram. Optimal power allocation and user selection in non-orthogonal multiple access systems. In *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, pages 1–6, 2016.

[37] Zhiguo Ding, Pingzhi Fan, and H Vincent Poor. Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions. *IEEE Trans. Veh. Technol.*, 65(8):6010–6023, 2015.

[38] Zhiguo Ding, Robert Schober, and H Vincent Poor. A general MIMO framework for NOMA downlink and uplink transmission based on signal alignment. *IEEE Trans. Wireless Commun.*, 15(6):4438–4454, 2016.

[39] Zhiguo Ding, Zheng Yang, Pingzhi Fan, and H Vincent Poor. On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users. *IEEE signal process. lett.*, 21(12):1501–1505, 2014.

[40] Canh T Dinh, Nguyen H Tran, Minh NH Nguyen, Choong Seon Hong, Wei Bao, Albert Y Zomaya, and Vincent Gramoli. Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Transactions on Networking*, 29(1):398–409, 2021.

[41] Konstantinos Dovelos and Boris Bellalta. A scheduling policy for downlink OFDMA in ieee 802.11 ax with throughput constraints. *arXiv preprint arXiv:2009.00413*, 2020.

[42] Yang Du, Binhong Dong, Zhi Chen, Xiaodong Wang, Zeyuan Liu, Pengyu Gao, and Shaoqian Li. Efficient multi-user detection for uplink grant-free NOMA: Prior-information aided adaptive compressive sensing perspective. *IEEE Journal on Selected Areas in Communications*, 2017.

[43] TS ETSI. 136 213 v12. 3.0 technical specification LTE. *Evolved Universal Terrestrial Radio Access (E-UTRA)*, 2014.

[44] Fang Fang, Haijun Zhang, Julian Cheng, and Victor CM Leung. Energy-efficient resource allocation for downlink non-orthogonal multiple access network. *IEEE Trans. Commun.*, 64(9):3722–3732, 2016.

[45] Guillem Femenias and Felip Riera-Palou. Scheduling and resource allocation in downlink multiuser MIMO-OFDMA systems. *IEEE Trans. Commun.*, 64(5):2019–2034, 2016.

[46] Miltiades C Filippou, Paul De Kerret, David Gesbert, Tharmalingam Ratnarajah, Adriano Pastore, and George A Ropokis. Coordinated shared spectrum precoding with distributed CSIT. *IEEE Transactions on Wireless Communications*, 15(8):5182–5192, 2016.

[47] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[48] Yaru Fu, Lou Salaün, Chi Wan Sung, and Chung Shue Chen. Subcarrier and power allocation for the downlink of multicarrier NOMA systems. *IEEE Trans. Veh. Technol.*, 67(12):11833–11847, 2018.

[49] Ebrahim A Gharavol, Ying-Chang Liang, and Koen Mouthaan. Robust downlink beamforming in multiuser MISO cognitive radio networks with imperfect channel-state information. *IEEE Transactions on Vehicular Technology*, 59(6):2852–2860, 2010.

[50] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int. Conf. Artif. Intell. Stat.*, pages 249–256, 2010.

[51] Andrea Goldsmith, Syed Ali Jafar, Ivana Maric, and Sudhir Srinivasa. Breaking spectrum gridlock with cognitive radios: An information theoretic perspective. *Proceedings of the IEEE*, 97(5):894–914, 2009.

[52] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the RF smog: Making 802.11n robust to cross-technology interference. In *Proceedings of ACM SIGCOMM*, volume 41, pages 170–181, 2011.

[53] Ryan E Guerra, Narendra Anand, Clayton Shepard, and Edward W Knightly. Opportunistic channel estimation for implicit 802.11 af MU-MIMO. In *Proc. 28th Int. Teletraffic Congr. (ITC)*, volume 1, pages 60–68, 2016.

[54] Jiajia Guo, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. Convolutional neural network based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis. *IEEE Trans. Wireless Commun.*, 2020.

[55] Muhammad Fainan Hanif, Zhiguo Ding, Tharmalingam Ratnarajah, and George K Karagiannidis. A minorization-maximization method for optimizing sum rate in the downlink of non-orthogonal multiple access systems. *IEEE Trans. Signal Process.*, 64(1):76–88, 2016.

[56] Shiwen He, Yongming Huang, Haiming Wang, Shi Jin, and Luxi Yang. Leakage-aware energy-efficient beamforming for heterogeneous multicell multiuser systems. *IEEE Journal on Selected Areas in Communications*, 32(6):1268–1281, 2014.

[57] Yuan Yuan He and Subhrakanti Dey. Sum rate maximization for cognitive MISO broadcast channels: Beamforming design and large systems analysis. *IEEE Transactions on Wireless Communications*, 13(5):2383–2401, 2014.

[58] Winston WL Ho and Ying-Chang Liang. Optimal resource allocation for multiuser MIMO-OFDM systems with user rate constraints. *IEEE Trans. Veh. Technol.*, 58(3):1190–1203, 2009.

[59] William D Horne. Adaptive spectrum access: Using the full spectrum space. In *Proceedings of Telecommunications Policy Research Conference (TPRC)*, 2003.

[60] Jun Huang, Jingjing Cui, Cong-Cong Xing, and Hamid Gharavi. Energy-efficient SWIPT-empowered D2D mode selection. *IEEE Trans. Veh. Technol.*, 2020.

[61] Jun Huang, Shuai Huang, Cong-Cong Xing, and Yi Qian. Game-theoretic power control mechanisms for device-to-device communications underlaying cellular system. *IEEE Trans. Veh. Technol.*, 67(6):4890–4900, 2018.

[62] Jun Huang, Cong-cong Xing, and Mohsen Guizani. Power allocation for D2D communications with SWIPT. *IEEE Trans. Wireless Commun.*, 2020.

[63] Xumin Huang, Peichun Li, Rong Yu, Yuan Wu, Kan Xie, and Shengli Xie. Fedparking: A federated learning based parking space estimation with parked vehicle assisted edge computing. *IEEE Transactions on Vehicular Technology*, 70(9):9355–9368, 2021.

[64] Yongwei Huang, Qiang Li, Wing-Kin Ma, and Shuzhong Zhang. Robust multicast beamforming for spectrum sharing-based cognitive radios. *IEEE Transactions on Signal Processing*, 60(1):527, 2012.

[65] Anne Humeau-Heurtier, Chiu-Wen Wu, Shuen-De Wu, Guillaume Mahé, and Pierre Abraham. Refined multiscale Hilbert–Huang spectral entropy and its application to central and peripheral cardiovascular data. *IEEE Trans. Biomed. Eng.*, 63(11):2405–2415, 2016.

[66] David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *Amer. Statistician*, 58(1):30–37, 2004.

[67] 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.

[68] IEEE 802.11n. IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput. *IEEE Std. 802.11n*, pages 1–565, Oct 2009.

[69] IEEE 802.11p-2010. IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. *IEEE Standards*, June 2010.

[70] IEEE P802.11ax. IEEE draft standard for information technology – telecommunications and information exchange between systems local and metropolitan area networks – specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment enhancements for high efficiency WLAN. *IEEE P802.11ax/D4.0*, pages 1–746, March 2019.

[71] Muhammad Inamullah, Bhaskaran Raman, and Nadeem Akhtar. Will my packet reach on time? deadline-based uplink OFDMA scheduling in 802.11 ax WLANs. In *Proc. ACM MSWiM*, pages 181–189, 2020.

[72] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[73] Youngrok Jang, Gyuyeol Kong, Minchae Jung, Sooyong Choi, and Il-Min Kim. Deep autoencoder based CSI feedback with feedback errors and feedback delay in FDD massive MIMO systems. *IEEE Wireless Commun. Lett.*, 8(3):833–836, 2019.

[74] Xiwen Jiang, Alexis Decurninge, Kalyana Gopala, Florian Kaltenberger, Maxime Guillaud, Dirk Slock, and Luc Deneire. A framework for over-the-air reciprocity calibration for TDD massive MIMO systems. *IEEE Trans. Wireless Commun.*, 17(9):5975–5990, 2018.

[75] Xiwen Jiang and Florian Kaltenberger. Channel reciprocity calibration in TDD hybrid beamforming massive MIMO systems. *IEEE J. Sel. Topics Signal Process.*, 12(3):422–431, 2018.

[76] Samy Kambou, Clency Perrine, Meriem Afif, Yannis Pousset, and Christian Olivier. Resource allocation based on cross-layer QoS-guaranteed scheduling for multi-service multi-user MIMO-OFDMA systems. *Wireless Netw.*, 23(3):859–880, 2017.

[77] Mohammad T Kawser, Nafiz Imtiaz Bin Hamid, Md Nayeemul Hasan, M Shah Alam, and M Musfiqur Rahman. Downlink SNR to CQI mapping for different multiple-antenna techniques in LTE. *Int. J. Inf. Electron. Eng.*, 2(5):757, 2012.

[78] Pedram Kheirkhah Sangdeh, Chengzhang Li, Hossein Pirayesh, Shichen Zhang, Huacheng Zeng, and Y Thomas Hou. CF4FL: A communication framework for federated learning in transportation systems. *IEEE Transactions on Wireless Communications*, 2022.

[79] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Aryan Mobiny, and Huacheng Zeng. LB-SciFi: Online learning-based channel feedback for MU-MIMO in wireless LANs. In *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, pages 1–11, 2020.

[80] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Qiben Yan, and Huacheng Zeng. DM-COM: Combining device-to-device and MU-MIMO communications for cellular networks. *IEEE Internet of Things Journal*, 8(17):13516–13527, 2021.

[81] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Qiben Yan, Kai Zeng, Wenjing Lou, and Huacheng Zeng. A practical downlink NOMA scheme for wireless LANs. *IEEE Transactions on Communications*, 68(4):2236–2250, 2020.

[82] Pedram Kheirkhah Sangdeh and Huacheng Zeng. DeepMux: Deep-learning-based channel sounding and resource allocation for IEEE 802.11 ax. *IEEE Journal on Selected Areas in Communications*, 39(8):2333–2346, 2021.

[83] Daehyon Kim. Normalization methods for input and output vectors in backpropagation neural networks. *Int. J. Comput. Math.*, 71(2):161–171, 1999.

[84] Junghoon Kim, Taejoon Kim, Morteza Hashemi, Christopher G Brinton, and David J Love. Joint optimization of signal design and resource allocation in wireless D2D edge computing. In *Proc. IEEE INFOCOM*, pages 2086–2095, 2020.

[85] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Represent.*, pages 1–13, 12 2015.

[86] Qinglei Kong, Feng Yin, Rongxing Lu, Beibei Li, Xiaohong Wang, Shuguang Cui, and Ping Zhang. Privacy-preserving aggregation for federated learning-based navigation in vehicular fog. *IEEE Transactions on Industrial Informatics*, 2021.

[87] Tejashri Kuber, Dola Saha, and Ivan Seskar. Predicting channel transition for MU-MIMO beamforming. In *Proc. IEEE 5G World Forum (5GWF)*, pages 83–88, 2018.

[88] Mehmet Şükrü Kuran, A Dilmac, Ömer Topal, Baris Yamansavascilar, Stefano Avallone, and Tuna Tugcu. Throughput-maximizing OFDMA scheduler for IEEE 802.11 ax networks. In *Proc. IEEE 31st Annu. Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*, pages 1–7, 2020.

[89] Sachitha Kusaladharma and Chinthananda Tellambura. Secondary user interference characterization for spatially random underlay networks with massive MIMO and power control. *IEEE Transactions on Vehicular Technology*, 66(9):7897–7912, 2017.

[90] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, pages 19–35, 2021.

[91] Peng Lan, Chao Zhai, Lizhen Chen, Bin Gao, and Fenggang Sun. Optimal power allocation for bi-directional full duplex underlay cognitive radio networks. *IET Communications*, 12(2):220–227, 2017.

[92] Yann Le Cun, Ido Kanter, and Sara A Solla. Eigenvalues of covariance matrices: Application to neural-network learning. *Phys. Rev. Lett.*, 66(18):2396, 1991.

[93] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[94] Hyun-Suk Lee and Jang-Won Lee. Adaptive transmission scheduling in wireless networks for asynchronous federated learning. *IEEE Journal on Selected Areas in Communications*, 39(12):3673–3687, 2021.

[95] Chengzhang Li, Qingyu Liu, Shaoran Li, Yongce Chen, Y Thomas Hou, Wenjing Lou, and Sastry Kompella. Scheduling with age of information guarantee. *IEEE/ACM Transactions on Networking*, 2022.

[96] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[97] Xiangyi Li and Huaming Wu. Spatio-temporal representation with deep neural recurrent network in MIMO CSI feedback. *IEEE Wireless Commun. Lett.*, 2020.

[98] Xiaoshuai Li, Rajan Shankaran, Mehmet A Orgun, Gengfa Fang, and Yubin Xu. Resource allocation for underlay D2D communication with proportional fairness. *IEEE Trans. Veh. Technol.*, 67(7):6244–6258, 2018.

[99] Xunan Li, Chong Li, and Ye Jin. Dynamic resource allocation for transmit power minimization in OFDM-based NOMA systems. *IEEE Commun. Lett.*, 20(12):2558–2561, 2016.

[100] Wei Liang, Zhiguo Ding, Yonghui Li, and Lingyang Song. User pairing for downlink non-orthogonal multiple access networks using matching algorithm. *IEEE Trans. Commun.*, 65(12):5319–5332, 2017.

[101] Yong Liao, Haimei Yao, Yuanxiao Hua, and Chunguo Li. CSI feedback based on deep learning for massive MIMO systems. *IEEE Access*, 7:86810–86820, 2019.

[102] Chaiman Lim, Taesang Yoo, Bruno Clerckx, Byungju Lee, and Byonghyo Shim. Recent trend of multiuser MIMO in LTE-advanced. *IEEE Communications Magazine*, 51(3):127–135, 2013.

[103] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

[104] Fei Liu, Petri Mähönen, and Marina Petrova. Proportional fairness-based power allocation and user set selection for downlink NOMA systems. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 1–6, 2016.

[105] Su Liu, Jiong Yu, Xiaoheng Deng, and Shaohua Wan. FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[106] Ya-Feng Liu. Complexity analysis of joint subcarrier and power allocation for the cellular downlink OFDMA system. *IEEE Wireless Commun. Lett.*, 3(6):661–664, 2014.

[107] Yuanwei Liu, Zhiguo Ding, Maged Elkashlan, and Jinhong Yuan. Non-orthogonal multiple access in large-scale underlay cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 65(12):10152–10157, 2016.

[108] Yuanwei Liu, Zhijin Qin, Maged Elkashlan, Yue Gao, and Lajos Hanzo. Enhancing the physical layer security of non-orthogonal multiple access in large-scale networks. *IEEE Trans. Wireless Commun.*, 16(3):1656–1672, 2017.

[109] Zhenyu Liu, Lin Zhang, and Zhi Ding. Exploiting bi-directional channel reciprocity in deep learning for low rate massive MIMO CSI feedback. *IEEE Wireless Commun. Lett.*, 8(3):889–892, 2019.

[110] Hanqing Lou, Monisha Ghosh, Pengfei Xia, and Robert Olesen. A comparison of implicit and explicit channel feedback methods for MU-MIMO WLAN systems. In *IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 419–424, 2013.

[111] Chao Lu, Wei Xu, Hong Shen, Jun Zhu, and Kezhi Wang. MIMO channel information feedback using deep recurrent network. *IEEE Commun. Lett.*, 23(1):188–191, 2019.

[112] Jiawei Lyu, Hui-Ming Wang, and Ke-Wen Huang. Physical layer security in D2D underlay cellular networks with poisson cluster process. *IEEE Trans. Commun.*, 68(11):7123–7139, 2020.

[113] Xiaofu Ma, Qinghai Gao, Ji Wang, Vuk Marojevic, and Jeffrey H Reed. Dynamic sounding for multi-user MIMO in wireless LANs. *IEEE Trans. Consum. Electron.*, 63(2):135–144, 2017.

[114] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[115] Jawad Mirza, Gan Zheng, Kai-Kit Wong, and Saqib Saleem. Joint beamforming and power optimization for D2D underlaying cellular networks. *IEEE Trans. Veh. Technol.*, 67(9):8324–8335, 2018.

[116] Marco Moretti, Luca Sanguinetti, and Xiaodong Wang. Resource allocation for power minimization in the downlink of THP-based spatial multiplexing MIMO-OFDMA systems. *IEEE Trans. Veh. Technol.*, 64(1):405–411, 2015.

[117] Toshihisa Nabetani, Narendar Madhavan, Hiroki Mori, and Tsuguhide Aoki. A novel low-overhead channel sounding protocol for downlink multi-user MIMO in IEEE 802.11 ax WLAN. *IEICE Trans. Commun.*, 101(3):924–932, 2018.

[118] Nibedita Nandan, Sudhan Majhi, and Hsiao-Chun Wu. Maximizing secrecy capacity of underlay MIMO-CRN through bi-directional zero-forcing beamforming. *IEEE Transactions on Wireless Communications*, 17(8):5327–5337, 2018.

[119] Nibedita Nandan, Sudhan Majhi, and Hsiao-Chun Wu. Secure beamforming for MIMO-NOMA-based cognitive radio network. *IEEE Communications Letters*, 22(8):1708–1711, 2018.

[120] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, PA, USA, 1994.

[121] Derrick Wing Kwan Ng, Ernest S Lo, and Robert Schober. Dynamic resource allocation in MIMO-OFDMA systems with full-duplex and hybrid relaying. *IEEE Trans. Commun.*, 60(5):1291–1304, 2012.

[122] Van-Dinh Nguyen, Le-Nam Tran, Trung Q Duong, Oh-Soon Shin, and Ronan Farrell. An efficient precoder design for multiuser MIMO cognitive radio networks with interference constraints. *IEEE Transactions on Vehicular Technology*, 66(5):3991–4004, 2017.

[123] Nick Galov. How Many IoT Devices Are There in 2021? *Techjury*, 9-Sep-2021. [Online]. Available: https://techjury.net/blog/how-many-iot-devices-are-there, [Accessed: 17-Sep-2021].

[124] Kentaro Nishimori, Takefumi Hiraguri, Tsutomu Mitsui, and Hiroyoshi Yamada. Effectiveness of implicit beamforming with large number of antennas using calibration technique in multi-user MIMO system. *Electron.*, 6(4):91, 2017.

[125] Yair Noam and Andrea J Goldsmith. Blind null-space learning for MIMO underlay cognitive radio with primary user interference adaptation. *IEEE Transactions on Wireless Communications*, 12(4):1722–1734, 2013.

[126] Yair Noam and Andrea J Goldsmith. The one-bit null space learning algorithm and its convergence. *IEEE Trans. Signal Process.*, 61(24):6135–6149, 2013.

[127] NVIDIA. Hardware for self-driving cars. `https://tinyurl.com/59any9fc`. [Online; accessed 23-Sep-2022].

[128] Jinhyung Oh, Heon-Jin Hong, and Hyung-Do Choi. Performance analysis for channel sounding in IEEE 802.11 ac network. In *Proc. Int. Conf. Inf. Commun. Technol. Convergence*, pages 1240–1242, 2015.

[129] José Armando Oviedo and Hamid R Sadjadpour. A fair power allocation approach to NOMA in multiuser SISO systems. *IEEE Trans. Veh. Technol.*, 66(9):7974–7985, 2017.

[130] Berna Özbek, Mylene Pischella, and Didier Le Ruyet. Energy efficient resource allocation for underlaying multi-D2D enabled multiple-antennas communications. *IEEE Trans. Veh. Technol.*, 2020.

[131] P. Kheirkhah Sangdeh, A. Quadri, and H. Zeng. Demo: A practical spectrum sharing solution. `https://www.cse.msu.edu/~hzeng/spectrum_sharing.html`. Online; accessed 3 September 2021.

[132] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[133] Harri Pennanen, Antti Tölli, and Matti Latva-aho. Multi-cell beamforming with decentralized coordination in cognitive and cellular networks. *IEEE Transactions on Signal Processing*, 62(2):295–308, 2014.

[134] Eldad Perahia and Michelle X Gong. Gigabit wireless LANs: An overview of IEEE 802.11 ac and 802.11 ad. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 15(3):23–33, 2011.

[135] Andre Perez. *Wi-Fi Integration to the 4G Mobile Network*. John Wiley & Sons Inc., 2018.

[136] Shiva Raj Pokhrel and Jinho Choi. A decentralized federated learning approach for connected autonomous vehicles. In *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, 2020.

[137] Shiva Raj Pokhrel and Jinho Choi. Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. *IEEE Transactions on Communications*, 68(8):4734–4746, 2020.

[138] Raghunandan M Rao, Harpeet S Dhillon, Vuk Marojevic, and Jeffrey H Reed. Analysis of worst-case interference in underlay radar-massive MIMO spectrum sharing scenarios. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2019.

[139] Getachew Redieteab, Laurent Cariou, Philippe Christin, and J-F Helard. PHY+ MAC channel sounding interval analysis for IEEE 802.11 ac MU-MIMO. In *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, pages 1054–1058, 2012.

[140] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[141] Felip Riera-Palou and Guillem Femenias. Cluster-based cooperative MIMO-OFDMA cellular networks: Scheduling and resource allocation. *IEEE Trans. Veh. Technol.*, 67(2):1202–1216, 2018.

[142] Olivier Rousseaux, Geert Leus, and Marc Moonen. A blind multi-user MIMO transceiver using code modulation in a multipath context. In *Proceedings of International Conference on Digital Signal Processing*, volume 1, pages 267–270, 2002.

[143] Mohammad Sadegh Safari, Vahid Pourahmadi, and Shabnam Sodagari. Deep UL2DL: Data-driven channel knowledge transfer from uplink to downlink. *IEEE Open J. Veh. Technol.*, 1:29–44, 2019.

[144] 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.

[145] Yuris Mulya Saputra, Diep Nguyen, Hoang Thai Dinh, Thang X Vu, Eryk Dutkiewicz, and Symeon Chatzinotas. Federated learning meets contract theory: Economic-efficiency framework for electric vehicle networks. *IEEE Transactions on Mobile Computing*, 2020.

[146] R. Sarvendranath and N. B. Mehta. Exploiting power adaptation with transmit antenna selection for interference-outage constrained underlay spectrum sharing. *IEEE Transactions on Communications*, 68(1):480–492, 2020.

[147] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2020.

[148] Byoungjin Seok, Jose Costa Sapalo Sicato, Tcydenova Erzhena, Canshou Xuan, Yi Pan, and Jong Hyuk Park. Secure D2D communication for 5G IoT network based on lightweight cryptography. *Applied Sciences*, 10(1):217, 2020.

[149] Wenbo Shen, Peng Ning, Xiaofan He, Huaiyu Dai, and Yao Liu. MCR decoding: A MIMO approach for defending against wireless jamming attacks. In *Proceedings of IEEE Conference on Communications and Network Security (CNS)*, pages 133–138, 2014.

[150] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. Argos: Practical many-antenna base stations. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 53–64, 2012.

[151] Hanif D Sherali and Warren P Adams. Reformulation-linearization techniques for discrete optimization problems. In *Handbook of combinatorial optimization*, pages 479–532. Springer, 1998.

[152] Wenqi Shi, Sheng Zhou, Zhisheng Niu, Miao Jiang, and Lu Geng. Joint device scheduling and resource allocation for latency constrained wireless federated learning. *IEEE Transactions on Wireless Communications*, 20(1):453–467, 2021.

[153] Yi Shi, Jia Liu, Canming Jiang, Cunhao Gao, and Y Thomas Hou. A DoF-based link layer model for multi-hop MIMO networks. *IEEE Trans. Mobile Comput.*, 13(7):1395–1408, 2014.

[154] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[155] Statista. Number of wireless local area network (WLAN) connected devices worldwide from 2016 to 2021 . *Statista*, Dec-2017. [Online]. Available: https://www.statista.com/statistics/802706/world-wlan-connected-device, [Accessed: 17-Sep-2021].

[156] Jung Hoon Suh, Jun Zhu, and Osama Aboul-Magd. System and method for quantization of angles for beamforming feedback, February 6 2018. US Patent 9,887,749.

[157] Haifeng Sun, Shiqi Li, F Richard Yu, Qi Qi, Jingyu Wang, and Jianxin Liao. Toward communication-efficient federated learning in the internet of things with edge computing. *IEEE Internet of Things Journal*, 7(11):11053–11067, 2020.

[158] C Suraci, S Pizzi, D Garompolo, G Araniti, A Molinaro, and A Iera. Trusted and secured D2D-aided communications in 5G networks. *Ad Hoc Networks*, page 102403, 2021.

[159] Afaf Taik, Zoubeir Mlika, and Soumaya Cherkaoui. Clustered vehicular federated learning: Process and optimization. *arXiv preprint arXiv:2201.11271*, 2022.

[160] Junjie Tan, Ying-Chang Liang, Lin Zhang, and Gang Feng. Deep reinforcement learning for joint channel selection and power control in D2D networks. *IEEE Trans. Wireless Commun.*, 2020.

[161] Robert Endre Tarjan. Two streamlined depth-first search algorithms. *Fundamenta Informaticae*, 9(1):85–94, 1986.

[162] Stelios Timotheou and Ioannis Krikidis. Fairness for non-orthogonal multiple access in 5G systems. *IEEE Signal Process. Lett.*, 22(10):1647–1651, 2015.

[163] Le-Nam Tran, Mats Bengtsson, and Björn Ottersten. Iterative precoder design and user scheduling for block-diagonalized systems. *IEEE Transactions on Signal Processing*, 60(7):3726–3739, 2012.

[164] Aashma Uprety, Danda B Rawat, and Jiang Li. Privacy preserving misbehavior detection in IoV using federated machine learning. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6, 2021.

[165] Uyoata Uyoata, Joyce Mwangama, and Mqhele Dlodlo. Robust beamforming for D2D multicast communication. *Physical Communication*, 43:101217, 2020.

[166] Bhukya Venkatesh, Nadella Bala Sai Krishna, and Sonali Chouhan. Distributed optimal power allocation using game theory in underlay cognitive radios. In *Data Communication and Networks*, pages 295–304. Springer, 2020.

[167] Sergio Verdu. *Multiuser detection*. Cambridge university press, 1998.

[168] Hongyi Wang, Scott Sievert, Zachary Charles, Shengchao Liu, Stephen Wright, and Dimitris Papailiopoulos. ATOMO: communication-efficient learning via atomic sparsification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9872–9883, 2018.

[169] Kaidong Wang and Konstantinos Psounis. Scheduling and resource allocation in 802.11 ax. In *Proc. IEEE INFOCOM*, pages 279–287, 2018.

[170] Kaidong Wang and Konstantinos Psounis. Efficient scheduling and resource allocation in 802.11 ax multi-user transmissions. *Comput. Commun.*, 152:171–186, 2020.

[171] Siyu Wang, Fangfang Liu, and Hailun Xia. Content-based vehicle selection and resource allocation for federated learning in IoV. In *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–7, 2021.

[172] Tianqi Wang, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. Deep learning-based CSI feedback approach for time-varying massive MIMO channels. *IEEE Wireless Commun. Lett.*, 8(2):416–419, 2018.

[173] W. Wang, R. Wu, and J. Liang. ADS-B signal separation based on blind adaptive beamforming. *IEEE Trans. Veh. Technol.*, 68(7):6547–6556, July 2019.

[174] Xue Wang, Tao Jin, Liangshuai Hu, and Zhihong Qian. Energy-efficient power allocation and Q-learning-based relay selection for relay-aided D2D communication. *IEEE Trans. on Veh. Technol.*, 69(6):6452–6462, 2020.

[175] Yichen Wang, Pinyi Ren, Qinghe Du, and Li Sun. Optimal power allocation for underlay-based cognitive radio networks with primary user's statistical delay QoS provisioning. *IEEE Transactions on Wireless Communications*, 14(12):6896–6910, 2015.

[176] Yung-Shun Wang, Y-W Peter Hong, and Wen-Tsuen Chen. Dynamic transmission policy for multi-pair cooperative device-to-device communication with block-diagonalization precoding. *IEEE Trans. Wireless Commun.*, 2019.

[177] Muhammad Waqas, Yong Niu, Yong Li, Manzoor Ahmed, Depeng Jin, Sheng Chen, and Zhu Han. A comprehensive survey on mobility-aware D2D communications: Principles, practice and challenges. *IEEE Commun. Surveys Tuts.*, 22(3):1863–1886, 2019.

[178] Daan Weller, Raoul Dijksman Mensenkamp, Arjan van der Vegt, Jan-Willem van Bloem, and Cees de Laat. Wi-Fi 6 performance measurements of 1024-QAM and DL OFDMA. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 1–7, 2020.

[179] Chao-Kai Wen, Wan-Ting Shih, and Shi Jin. Deep learning for massive MIMO CSI feedback. *IEEE Wireless Commun. Lett.*, 7(5):748–751, 2018.

[180] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

[181] Wi-Fi® Alliance. Wi-Fi® in 2019. *Wi-Fi Alliance*, 21-Feb-2019. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-in-2019, [Accessed: 21-Nov-2020].

[182] Jack H Winters. Signal acquisition and tracking with adaptive arrays in the digital mobile radio system IS-54 with flat fading. *IEEE Transactions on Vehicular Technology*, 42(4):377–384, 1993.

[183] Michael Wu, Bei Yin, Guohui Wang, Chris Dick, Joseph R Cavallaro, and Christoph Studer. Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):916–929, 2014.

[184] Mingqing Wu, Jiabing Wang, Yi-Hua Zhu, and Jintong Hong. High throughput resource unit assignment scheme for OFDMA-based WLAN. In *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, pages 1–8, 2019.

[185] Huizi Xiao, Jun Zhao, Qingqi Pei, Jie Feng, Lei Liu, and Weisong Shi. Vehicle selection and resource optimization for federated learning in vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[186] Xiao Xiao, Xiaoming Tao, and Jianhua Lu. Energy-efficient resource allocation in LTE-based MIMO-OFDMA systems with user rate constraints. *IEEE Trans. Veh. Technol.*, 64(1):185–197, 2015.

[187] Xiufeng Xie, Xinyu Zhang, and Karthikeyan Sundaresan. Adaptive feedback compression for MIMO networks. In *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 477–488, 2013.

[188] Hong Xing, Yuanawei Liu, Arumugam Nallanathan, Zhiguo Ding, and H Vincent Poor. Optimal throughput fairness tradeoffs for downlink non-orthogonal multiple access over fading channels. *IEEE Trans. Wireless Commun.*, 17(6):3556–3571, 2018.

[189] Jie Xu and Heqiang Wang. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications*, 20(2):1188–1200, 2021.

[190] Tianyi Xu, Liangping Ma, and Gregory Sternberg. Practical interference alignment and cancellation for MIMO underlay cognitive radio networks with multiple secondary users. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 1009–1014, 2013.

[191] Wei Xu, Yuke Cui, Hua Zhang, Geoffrey Ye Li, and Xiaohu You. Robust beamforming with partial channel state information for energy efficient networks. *IEEE Journal on Selected Areas in Communications*, 33(12):2920–2935, 2015.

[192] 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, 2016.

[193] Qianqian Yang, Mahdi Boloursaz Mashhadi, and Deniz Gündüz. Deep convolutional compression for massive MIMO CSI feedback. In *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, pages 1–6, 2019.

[194] Yuwen Yang, Feifei Gao, Geoffrey Ye Li, and Mengnan Jian. Deep learning-based downlink channel prediction for FDD massive MIMO system. *IEEE Commun. Lett.*, 23(11):1994–1998, 2019.

[195] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications*, 20(3):1935–1949, 2021.

[196] Zheng Yang, Zhiguo Ding, Pingzhi Fan, and Naofal Al-Dhahir. A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems. *IEEE Trans. Wireless Commun.*, 15(11):7244–7257, 2016.

[197] Mohand Yazid and Adlen Ksentini. Modeling and performance analysis of the main MAC and PHY features of the 802.11 ac standard: A-MPDU aggregation vs spatial multiplexing. *IEEE Trans. Veh. Technol.*, 67(11):10243–10257, 2018.

[198] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8:23920–23935, 2020.

[199] Wenjuan Yu, Leila Musavian, and Qiang Ni. Link-layer capacity of NOMA under statistical delay QoS guarantees. *IEEE Trans. Commun.*, 66(10):4907–4922, 2018.

[200] Huacheng Zeng. INFB: A low-overhead downlink MU-MIMO scheme for wireless LANs. In *accepted by IEEE ICNP*, 2018.

[201] Jiankang Zhang, Sheng Chen, Xiaomin Mu, and Lajos Hanzo. Turbo multi-user detection for OFDM/SDMA systems relying on differential evolution aided iterative channel estimation. *IEEE Transactions on Communications*, 60(6):1621–1633, 2012.

[202] Meng Zhang and Yuan Liu. Secure beamforming for untrusted MISO cognitive radio networks. *IEEE Transactions on Wireless Communications*, 17(7):4861–4872, 2018.

[203] Rui Zhang, Feifei Gao, and Ying-Chang Liang. Cognitive beamforming made practical: Effective interference channel and learning-throughput tradeoff. *IEEE Transactions on Communications*, 2(58):706–718, 2010.

[204] Shangwei Zhang, Jiajia Liu, Hongzhi Guo, Mingping Qi, and Nei Kato. Envisioning device-to-device communications in 6G. *IEEE Network*, 34(3):86–91, 2020.

[205] Xinyi Zhang, Jun Wang, Jintao Wang, and Jian Song. A novel user pairing in downlink non-orthogonal multiple access. In *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, pages 1–5, 2018.

[206] Heng Zhao, Gregory Pottie, and Babak Daneshrad. Reciprocity calibration of TDD MIMO channel for interference alignment. *IEEE Trans. Wireless Commun.*, 2020.

[207] Jingjing Zhao, Yuanwei Liu, Kok Keong Chai, Yue Chen, and Maged Elkashlan. Joint subchannel and power allocation for NOMA enhanced D2D communications. *IEEE Trans. Commun.*, 65(11):5081–5094, 2017.

[208] Nan Zhao, Dongdong Li, Mingqian Liu, Yang Cao, Yunfei Chen, Zhiguo Ding, and Xianbin Wang. Secure transmission via joint precoding optimization for downlink MISO NOMA. *IEEE Trans. Veh. Technol.*, 68(8):7603–7615, Aug 2019.

[209] Nan Zhao, Xiaowei Pang, Zan Li, Yunfei Chen, Feng Li, Zhiguo Ding, and Mohamed-Slim Alouini. Joint trajectory and precoding optimization for UAV-assisted NOMA networks. *IEEE Trans. Commun.*, 67(5):3723–3735, 2019.

[210] Nan Zhao, Wei Wang, Jingjing Wang, Yunfei Chen, Yun Lin, Zhiguo Ding, and Norman C Beaulieu. Joint beamforming and jamming optimization for secure transmission in MISO-NOMA networks. *IEEE Trans. Commun.*, 67(3):2294–2305, 2018.