PUSHING THE ENVELOPE OF WI-FI NETWORKS USING DISTRIBUTED MULTI-USER MIMO

by

NEELAKANTAN NURANI KRISHNAN

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Narayan B. Mandayam

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

JANUARY, 2020

ABSTRACT OF THE DISSERTATION

Pushing the Envelope of Wi-Fi Networks Using Distributed Multi-User MIMO

By NEELAKANTAN NURANI KRISHNAN

Dissertation Director:

Narayan B. Mandayam

This dissertation presents a distributed multi-user MIMO Wi-Fi architecture referred to as D-MIMO that boosts network throughput performance compared to state-of-the-art Wi-Fi access points with co-located antennas. D-MIMO, at a high level, is a technique by which a set of wireless access points are synchronized and grouped together to jointly serve multiple users simultaneously. The cooperation between the access points reduces intra-network interference and hence improves spatial reuse of channels. We study D-MIMO Wi-Fi networks in four broad sections: (i) by prescribing lightweight and effective solutions to the problems of channel access and multi-user MIMO user selection in D-MIMO Wi-Fi, (ii) through experimental evaluations of the proposed solutions on a D-MIMO Wi-Fi network implemented in an indoor testbed using software defined radio platforms, (iii) by constructing a deep reinforcement learning framework to address dynamic resource management in D-MIMO Wi-Fi networks, and (iv) by investigating the benefits that the D-MIMO architecture brings to dense Wi-Fi networks operating in mmWave (60 GHz) bands. These components form the original contributions of this dissertation to knowledge.

Designing a D-MIMO Wi-Fi network invites us to revisit fundamental Wi-Fi concepts such as carrier sensing multiple access that governs medium/channel access among Wi-Fi access points. We propose a medium access protocol for D-MIMO that assimilates channel sensing observations from different access points to resolve channel contention

among D-MIMO groups. We also propose a novel way of using channel reciprocity and the network topology to select downlink multi-user (MU) MIMO recipients without requesting any form of channel state information feedback from the users during the selection phase. The proposed solutions are lightweight, do not require modifications at the user equipment, and hence will work with legacy 802.11ac devices. We compare the performance of the D-MIMO configuration to that of baseline dense Wi-Fi deployments (access points with co-located antennas), operating in 5 GHz bands, through extensive network simulations. We observe an improvement of 3.5$\times$ in median and 191% in mean user throughput, as well as a reduction of 61% in channel access delay with D-MIMO.

Next, we present an implementation of a distributed MIMO Wi-Fi group—using software defined radio platforms—in an indoor experimental testbed. The implemented setup consists of four two-antenna Wi-Fi access points (synchronized in time and phase using a GPS-disciplined clock reference system) and twenty two-antenna users, and is compliant with the 802.11ac very high throughput framework. We use this setup to serve as a proof-of-concept of the proposed lightweight MU-MIMO user selection algorithm. Through extensive experimental evaluations, we demonstrate that the proposed algorithm outperforms a simple random user selection strategy by achieving an improvement of up to 60% in median and 43% in mean group throughput performance. Furthermore, the proposed user selection algorithm performs close to optimality—the difference in performance between the proposed user selection algorithm and optimal user selection is a mere 13%.

As the third installment of this dissertation, we address two dynamic resource management problems germane to D-MIMO Wi-Fi networks: (i) channel assignment of D-MIMO groups, and (ii) deciding how to cluster access points to form D-MIMO groups, in order to maximize user throughput performance. These problems are known to be NP-Hard for which only heuristic solutions exist in literature and we explore the potential of harnessing principles from deep reinforcement learning (DRL) to address these challenges. We construct a DRL framework through which a learning agent interacts with a D-MIMO Wi-Fi network, learns about the network environment, and successfully converges to policies that effectively address the aforementioned

challenges. Through extensive simulations and on-line training based on D-MIMO Wi-Fi networks, we demonstrate the efficacy of DRL agents in achieving an improvement of 20% in user throughput performance compared to heuristic solutions, particularly when network conditions are dynamic. This work also showcases the effectiveness of DRL agents in meeting multiple network objectives simultaneously, for instance, maximizing throughput of users as well as fairness of throughput distribution among them.

In the final part of this dissertation, we consider dense Wi-Fi networks operating in mmWave (60 GHz) bands and use the D-MIMO architecture to improve user throughput performance in these networks compared to baseline arrangements. Rigorous network simulation results reveal an enhancement of 395% in average user throughput and a reduction of 75% in channel access delay with D-MIMO compared to baseline. We observe an interesting behavior wherein a user achieves very high modulation and coding scheme indices more number of times with the baseline configuration compared to D-MIMO, especially when the user is located close to an access point (AP). This behavior can be ascribed to two causes: i) a higher probability of line-of-sight of the short distance AP-user link (that favors baseline), and ii) a ramification of the use of zero-forcing precoding to cancel inter-user interference in D-MIMO. This observation motivates the design of future networks as amalgams of both baseline and D-MIMO arrangements.

Acknowledgments

First, I am highly indebted to my advisor Prof. Narayan B. Mandayam, whose ability to balance the contrasting aspects of seeing the big picture as well as focusing attention on details has always inspired me. Prof. Mandayam provided me with the liberty to pursue my research interests right from the get-go and he was always available whenever I needed guidance, even amidst his highly booked schedule as the Chair of the department. I am highly grateful for the time and effort that he dedicated to provide constructive suggestions to improve my research, both in its technical aspects as well as in its presentation. He has always been an advocate of my efforts and success in graduate school.

Being a part of WINLAB comes with several perks—one of them is the ready availability of Professors, who are experts in their respective fields, for discussions. I have had the good fortune to work with Prof. Dipankar Raychaudhuri, Prof. Roy Yates, and Prof. Emina Soljanin, who have always taken take time out of their busy schedules to provide valuable insights into my research. I feel blessed to be in their company and for having had the opportunity to discuss several of my ideas with them. I would like to express my gratitude to Prof. Raychaudhuri, Prof. Soljanin, and Prof. Zoran Gajic for participating in my dissertation proposal and defense committees. I would also like to acknowledge Dr. Gokul Sridharan, who was working as a post-doctoral researcher with Prof. Mandayam when I started at WINLAB. Dr. Sridharan spent several hours to clarify the basics of wireless communications and the associated mathematical framework, and inculcated in me the ability to comprehensively address the research problem in hand.

I would be remiss if I did not mention the contributions of Ivan Seskar—Chief Technology Officer, WINLAB—who honestly is an encyclopedia in anything and everything wireless. Over the past five years, I would have bothered Ivan umpteen number of

times but never did he hesitate to discuss my questions in detail and provide instantaneous yet insightful solutions. Ivan taught me how to ask the right questions at the right time and to always bear in mind the practicality of research. I am astounded at the level of knowledge he possesses and he has been/will consistently be a tremendous source of inspiration.

During my time in grad school, I was fortunate enough to be given the opportunity to intern with leading research institutions, one of them being Nokia Bell Labs in Sunnyvale, CA. I spent two productive summers there—under the guidance of Dr. Klaus Doppler, Head of Connectivity Labs—during which the general premise of this dissertation took shape. I am highly grateful to Dr. Doppler, Enrico-Henrik (Henkka) Rantala, and Dr. Eric Torkildson for being exceptional mentors and great colleagues. I would like to specially thank Eric for being a member of my dissertation defense committee.

Next, I would like to give a shout-out to my family away from home—Mrs. and Mr. Lakshman Easwaran, Mrs. and Mr. Vinod Venkattaraman, and their children in New Jersey. They ensured that I never missed my family, were always warm and welcoming, and provided me with delectable home-cooked food almost every weekend. I would like to next acknowledge Mrs. Rajeswari Satish, my guru in vocal training in Indian classical music in New Jersey. I am glad to have had the opportunity to continue my training even during my time in grad school. I am also grateful to my friends in New Jersey and in Palakkad for always being great company; I am refraining from mentioning names lest I forget someone.

Finally, I am tremendously grateful to my family in India—my parents, uncles and aunts, cousins, and grandparents. I consider myself to be highly fortunate to be part of a joint family that has constantly provided me with emotional support, has been understanding of the demands of grad school, and has always picked me up when times were difficult. Thank you very much for the unconditional love, unwavering care, and steadfast support!

Acknowledgment of funding: U.S. Office of Naval Research under grant number N00014-15-1-2168, National Science Foundation (NSF) "COSMOS" Project under grant number CNS-187923, and NSF "CRISP" Project under grant number 1541069.

Dedication

அம்மா , அப்பா , அம்மூம்மா , தாத்தா , குஞ்சம்மை

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emerging data-intensive applications such as augmented and virtual reality (AR/VR) and 8K video will drive the throughput requirements of Wi-Fi networks of the next generation. To meet rising throughput demands over time, Wi-Fi has steadily added support for wider channel bandwidths, including 40 MHz in 802.11n (in 2.4 GHz), 80/160 MHz in 802.11ac/ax (in 2.4/5 GHz), 320 MHz now under consideration [1], and 2160 MHz in 802.11ad (in 60 GHz). Another approach to achieving higher throughput is reducing the distance between neighboring Wi-Fi access points (APs). This allows each AP to serve a smaller area and provide its users with higher average SNR. In practice, however, these two approaches of wider channels and denser networks are at odds. Interference between closely-spaced co-channel APs limits availability of the larger bandwidth channels, which are accessed on a best-effort basis, causing devices fall back to narrower channels.

Distributed MIMO, also referred to as Network MIMO, has long been studied in theory because of its ability to dramatically increase the throughput of wireless networks. Distributed MIMO is widely regarded as an important technology to meet the performance objectives established for next-generation Wi-Fi networks (IEEE 802.11be) [2–5]. We envision a distributed MIMO system that consists of *several time and phase-synchronized APs that jointly transmit and receive signals, thereby acting as a single spatially-distributed virtual antenna array to simultaneously serve multiple users*. This is the reason why this technology is called distributed multi-user MIMO (we address it as D-MIMO hereon). The cooperation between APs reduces intra-network interference and hence improves spatial reuse of channels.

The overarching objective of this dissertation is to explore the use of the D-MIMO

architecture to improve user throughput performance of next-generation Wi-Fi networks with dense AP deployments. Specifically, we study D-MIMO Wi-Fi networks through the following four means:

1. by proposing effective and lightweight solutions for the problems of channel access and multi-user (MU) MIMO user selection in D-MIMO Wi-Fi, and by comparing the performance of D-MIMO Wi-Fi with baseline deployments through extensive network simulations,

2. by implementing a D-MIMO Wi-Fi network using software defined radios in an indoor testbed, and by performing experimental evaluations on the deployed setup to demonstrate the efficacy of the proposed solutions,

3. by constructing a deep reinforcement learning framework to address dynamic resource management in D-MIMO Wi-Fi networks, and

4. by extending the architecture of D-MIMO to dense Wi-Fi networks operating in mmWave (60 GHz) bands with high channel bandwidths and studying the benefits that D-MIMO brings compared to baseline configurations.

This chapter is organized as follows. Section 1.1 describes the high-level architecture of D-MIMO Wi-Fi. Section 1.2 illustrates the motivation for implementing D-MIMO in Wi-Fi networks with dense deployment of APs. Section 1.3 discusses several challenges associated with realizing D-MIMO Wi-Fi networks; addressing these challenges forms the foundation of this dissertation. Section 1.4 lists the major contributions of this dissertation along with its organization.

1.1 Architecture of D-MIMO Wi-Fi

The basic idea of D-MIMO is to divide the functionality of a Wi-Fi access point (AP) into two entities, radiohead (RH) and processing unit (PU), as seen in Figure 1.1. A RH is an external radio front end unit with one or more antennas providing a baseband signal interface for the PU. The PU encompasses all the functionalities of an AP that are not in the RH. The PU also maintains time and phase synchronization in RHs that

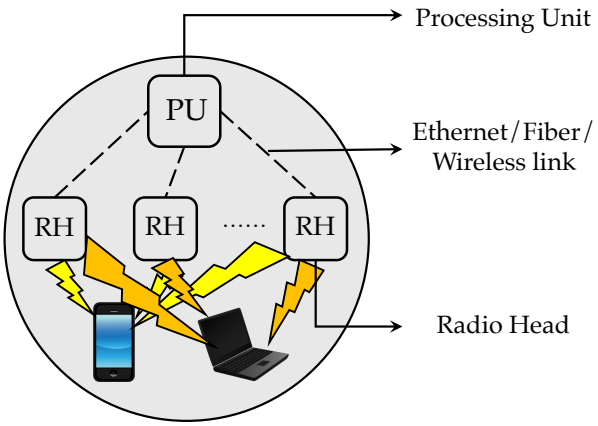


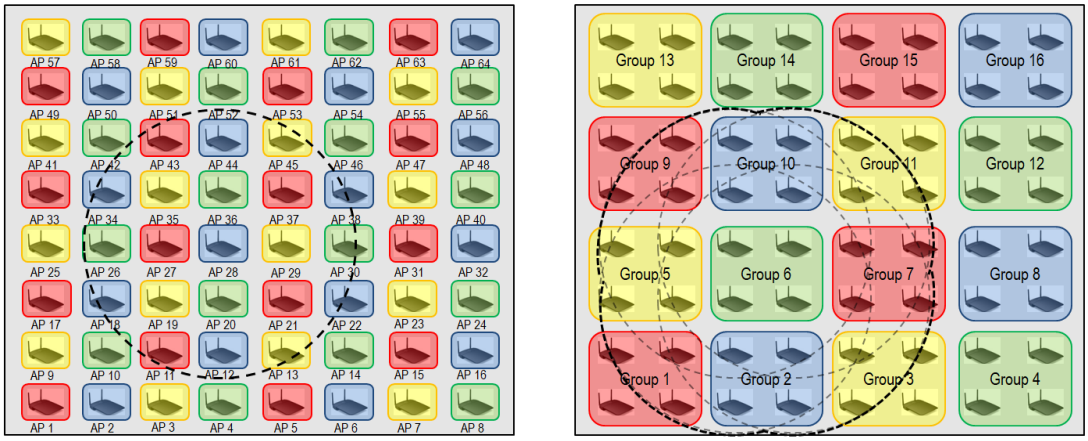
Figure 1.1: Representative architecture of a D-MIMO group. A group with M RHs with N antennas each can support M × N simultaneous downlink streams.

could be connected to the PU by using wired or wireless links. The core idea behind D-MIMO is that wireless RHs—which are synchronized—cooperatively form a single virtual antenna array that facilitates joint transmission of RHs to serve multiple users simultaneously (hence the name distributed MU-MIMO).

1.2 Motivation for the Use of D-MIMO in Wi-Fi Networks

The trend in wireless network design recently has been dense access point deployments to increase system capacity. Keeping this in mind, the following two aspects motivate D-MIMO implementations. We use cartoon examples of Wi-Fi networks with dense deployments of Wi-Fi APs to make the case for D-MIMO, as shown in Figure 1.2a (baseline configuration) and Figure 1.2b (D-MIMO configuration). The color assigned to a Wi-Fi AP or a D-MIMO group identifies the channel assigned to it.

1. **Wi-Fi densification**: 802.11ac/ax networks operate in the unlicensed 5 GHz frequency range, which is divided into a limited number of channels typically shared by multiple APs. As the inter-AP distance is reduced to improve SNR, more APs will share the same channel, resulting in less channel accesses per AP, and more control and management frames in total. Observe from Figure 1.2a that *hearing range* of a baseline AP (the distance till which its transmissions reach above the



(a) Baseline configuration (b) D-MIMO configuration

Figure 1.2: Cartoon example of baseline (dense deployment of Wi-Fi APs) and D-MIMO (groups of four RHs each) arrangements. Frequency reuse factor is four with each channel represented by a unique color. Dotted circle represents the hearing range of an AP/RH. The bold circle indicates the composite hearing range of a D-MIMO group.

clear channel assessment threshold) includes multiple other APs and their associated users in the same channel. In fact, in the scenario considered in Figure 1.2a, there are up to five other APs that are assigned the same blue channel as AP 28. This results in increased contention for the channel resources that will result in reduced transmission opportunities for all these APs as well as increased levels of co-channel interference in their downlink transmissions. Deploying D-MIMO (as described in Figure 1.2b), however, employs coordination among APs and results in fewer neighboring networks on the same channel while still preserving the desired SNR. The bold circle in Figure 1.2b denotes the composite hearing range of a D-MIMO group, which is obtained by combining the envelope of hearing ranges of each of the constituent RHs.

2. **Improved MIMO channel conditioning**: An alternative to densification of Wi-Fi APs is to increase the number of antennas per AP to support several simultaneous downlink transmissions. In fact, in the market currently, Wi-Fi APs equipped with up to eight antennas are available. However, adding more antennas to the AP does not lead to a corresponding increase in downlink rates. This is because of increased correlation of channels due to existence of few dominant scatterers,

small angle spread, and insufficient antenna spacing [6]. As described by [7, 8], co-located antenna systems can suffer from low channel rank that results in fewer spatial degrees of freedom and hence lower multiplexing gains. On the other hand, due to separation of transmitters in D-MIMO, spatial correlation of channels is reduced and hence the channel matrices have better conditioning. D-MIMO systems also achieve macro-diversity protection from all links having similar deep large scale fading. This is, however, not the case with the baseline setup in which multiple antennas sited in the same locale experience the same shadowing and thus cannot improve the situation.

1.3 Challenges with Realization of D-MIMO Wi-Fi Networks

Some of the challenges associated with realizing a D-MIMO Wi-Fi network in practice are described below. Providing solutions to address these challenges constitutes the core of this dissertation.

1. **Synchronization of** RHs: To achieve joint transmissions and receptions in D-MIMO, the RHs of a D-MIMO group must be tightly synchronized in time as well as phase. Various methods for synchronization of RHs have been proposed in literature: MegaMIMO [9], AirSync [10, 11], MegaMIMO 2.0 [12], DCAP [13], and Chorus [14]. Synchronization among RHs may be established using any of the aforementioned methods or by simply connecting them using wired links to a common external clock (as described in NEMOx [15]). This dissertation does not consider the synchronization problem; it assumes synchronization among RHs is established one way or the other.

2. **Channel access**: In Wi-Fi networks, devices share the channel using the carrier sense multiple access with collision avoidance (CSMA/CA) protocol, in which devices must sense the channel to be idle prior to channel access. With a conventional AP in the baseline setup, its co-located antennas are likely to share a common view of a channel's idle state. With distributed MIMO, however, the separation between radio heads will lead to different views of the channel state

making idle state decision ambiguous. Hence, it is important that the PU of a D-MIMO group assimilates the channel sensing observations of all the constituent RHs in order to resolve channel contention.

3. **Multi-antenna user scheduling**: A user associated with a D-MIMO group is at different ranges with respect to the RHs that are part of the group. Considering user and RH distributions, along with the knowledge of channel characteristics into account, we can improve downlink performance in terms of long term average throughput by choosing the optimal set of users to serve in order to maximize sum network throughput. However, the selection of users in the downlink has to be performed without incurring a high channel sounding/estimation overhead. Furthermore, once the users have been selected, optimal power allocation has to be performed across streams to maximize D-MIMO group throughput in every transmission opportunity while canceling interference between the simultaneous downlink streams.

4. **Dynamic resource management**: There are two major dynamic resource management problems associated with D-MIMO Wi-Fi networks, which are listed below:

   (a) CHANNEL ASSIGNMENT PROBLEM: If there are N D-MIMO groups and K available channels ($K < N$) (for instance, 16 groups and 4 channels in Figure 1.2b), what is the optimal channel assignment policy to maximize user throughput performance?

   (b) AP CLUSTERING PROBLEM: How should APs be clustered together to form D-MIMO groups to maximize throughput performance of users in the network? For instance, in the arrangement in Figure 1.2b, four neighboring RHs have been grouped to form one D-MIMO group. It is not clear, however, if this is the optimal clustering policy particularly when users are distributed non-uniformly in the network space.

   These problems are known to be NP-Hard and only heuristic solutions exist in literature. Additionally, it is desired to empower D-MIMO Wi-Fi networks with

an autonomous adaptation to dynamic network conditions, for instance, resilience to mobility of users and presence of random external Wi-Fi interference.

1.4 Contributions of this Dissertation

Devising solutions to effectively address the aforementioned challenges forms the crux of this dissertation. The specific contributions of the dissertation are listed below, along with the publications that it has generated.

1. **Chapter 2** describes novel, lightweight, and effective solutions for the problems of channel access and multi-user MIMO user selection in D-MIMO Wi-Fi networks. These solutions work with off-the-shelf Wi-Fi user devices without necessitating any modifications in them. Chapter 2 also provides an elaborate discussion based on the results obtained from extensive simulations of Wi-Fi networks in baseline and D-MIMO configurations. The metrics of comparison between the baseline and D-MIMO arrangements include channel access delay, user throughput, and modulation-and-coding-scheme (MCS) indices achieved by the users. The proposed solutions along with the general architectural change enable D-MIMO to achieve an improvement of 3.5$\times$ in median and 191% in average user throughput compared to baseline, as well as a reduction of 61% in channel access delay. The proposed solutions and the obtained results have been published as [16].

2. **Chapter 3** discusses the implementation of a D-MIMO Wi-Fi group in an indoor testbed using software defined radio platforms. Specifically, we build a D-MIMO network consisting of four RHs, synchronized in frequency and time, as well as twenty users, and use this setup as a proof-of-concept for the user selection algorithm proposed in Chapter 2. Rigorous experimental evaluations on this setup highlight that: (i) the proposed user selection algorithm achieves an increase of up to 60% in group throughput performance compared to a random user selection strategy, and (ii) the proposed algorithm performs close to optimal user selection.

3. **Chapter 4** explores the potential of harnessing concepts and algorithms from deep reinforcement learning (DRL) to address dynamic resource management in

D-MIMO Wi-Fi networks. In particular, this chapter considers the two dynamic resource management problems described in Section 1.3, item 4. We construct a DRL framework through which an agent interacts with a D-MIMO Wi-Fi network, learns about the network environment, and successfully converges to policies that address the aforementioned problems and hence improve the performance of the network. We consider practical network scenarios like non-uniform spatial distribution of users and user mobility. This chapter provides motivating scenarios for the use of DRL in D-MIMO Wi-Fi networks, the specifics of the DRL agents used for on-line training, and demonstrates the efficacy of the DRL agents in achieving an improvement of up to 20% in network throughput performance compared to popular heuristic solutions. This chapter has been published as [17].

4. **Chapter 5** studies dense Wi-Fi networks operating in mmWave (60 GHz) bands and extends the architecture of D-MIMO to improve user throughput performance compared to state-of-the-art wireless access points (AP) with co-located antennas (baseline). Rigorous network simulation results reveal an enhancement of up to 395% in average user throughput and a reduction of 75% in channel access delay with D-MIMO compared to baseline. We observe an interesting behavior wherein a user achieves very high modulation and coding scheme (MCS) indices more number of times with the baseline configuration compared to D-MIMO, especially when the user is located close to an AP. We substantiate this behavior based on the histogram of distances (between AP/RH and user) at which different MCS indices are achieved and provide a guideline to design future Wi-Fi networks. The results from this chapter have been compiled in [18].

Analysis and results from our previous publications in [19–22] have not been included in this dissertation as they do not fit the scope of the topic.

Chapter 2

Channel Access and Multi-User MIMO User Selection in D-MIMO Wi-Fi Networks[§]

2.1 Summary and Organization

This chapter proposes light-weight solutions to the problems of channel access as well as multi-user (MU) MIMO user selection in D-MIMO. The proposed solutions do not require any modifications on the user side and hence will work with legacy 802.11ac devices. This chapter first reviews existing literature in the field of distributed MIMO and its extension to Wi-Fi networks in Section 2.2. The subsequent technical content of this chapter is divided into the following sections: (i) proposal of a channel access procedure for D-MIMO Wi-Fi in Section 2.3, (ii) proposal of a lightweight user selection algorithm for MU-MIMO transmission in Section 2.4, (iii) optimal power allocation among concurrent streams to maximize D-MIMO group throughput in Section 2.5, and (iv) a description of the network simulation scenarios, the implemented channel model, and a discussion of the results obtained from extensive network simulations in Section 2.6.

2.2 Literature Review

An extensive survey of D-MIMO techniques and developments was given in [23]. Works in [24–28] showed how the capacity of a single D-MIMO group scaled with the number of transmitters through analysis and simulations, along with precoding algorithms for single and multiple D-MIMO groups. However, these works were under the assumption of a cellular network model in which, owing to centralized spectrum

[§]Parts of this chapter have been published as [16]

allocation, each D-MIMO group was assumed to act independently of others. This assumption is hence invalid in Wi-Fi networks in which channel access is mediated by distributed listen-before-talk-like protocols. Authors in [29–31] facilitated cooperation among wireless access points (AP) belonging to an enterprise Wi-Fi network by using successive interference cancellation to recover/decode collided packets. This however necessitated that at least one AP reliably decoded a packet in the presence of interference.

Practical system designs and implementations of D-MIMO are described in [9, 10, 12, 14, 32], which describe the merits of using distributed MU-MIMO over deploying independent APs with co-located antennas. However, these works do not propose algorithms to choose users to maximize group throughput. Furthermore, they describe the implementation in case of a single D-MIMO group and do not consider inter-group interference or how medium access will be resolved when multiple D-MIMO groups are involved. NEMOx [15] facilitated efficient spatial reuse in distributed MIMO networks, by introducing a scalable random access medium access control (MAC) architecture that connects a small number of neighboring APs for clustered D-MIMO. Its client selection algorithm involved solving an opportunistic scheduling problem iteratively, with time-averaged channel state information (CSI) feedback from all the users, to select the user(s) which maximized the sum rate. This algorithm, however, was computationally intensive. Furthermore, the proposed modifications to the channel access mechanism were not directly compliant with the 802.11 standards. The authors in [13] proposed a user selection algorithm for D-MIMO that randomly selected fewer users than what the D-MIMO group was capable of in order to achieve better channel conditioning. Although the algorithm in [13] chose users without requesting CSI feedback from the users, it is desirable to propose a user selection algorithm that serves as many number of users in a transmission opportunity as possible.

Recent works in MU-MIMO user selection [33, 34] (and references therein) demonstrated how to choose users in a MU-MIMO setting to maximize group throughput. These approaches assumed either explicit (albeit compressed) or implicit CSI feedback from the users. In contrast, our objective is to propose a lightweight user selection

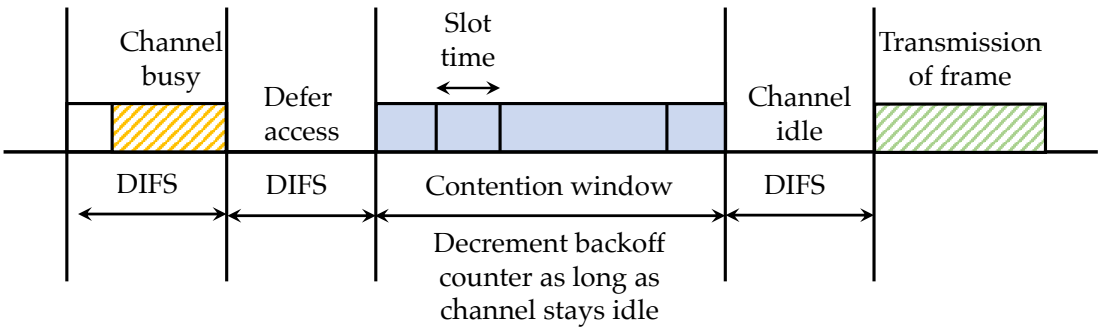


Figure 2.1: Timeline of distributed coordination function (DCF) for a baseline Wi-Fi AP

algorithm, when transmitters are spatially separate, which does not require explicit nor compressed/uncompressed CSI feedback from the users.

2.3 Channel Access

This section first briefly introduces the channel access mechanism in baseline Wi-Fi networks, explains why such a channel access procedure is not directly applicable to D-MIMO Wi-Fi networks, and proposes the necessary changes to the protocol for it to work in a D-MIMO configuration.

2.3.1 Channel Access in Baseline Wi-Fi

Wi-Fi uses distributed coordination function (DCF) as a channel access protocol to mediate channel access among wireless stations [35]. In short, DCF is based on physical and virtual carrier sensing using input from one or more co-located antennas to gain a transmission opportunity (TXOP). DCF randomizes channel access in a distributed and adaptive fashion to avoid collisions.

The timeline of DCF is described in Figure 2.1. When a frame (or a MAC service data unit) arrives at the head of the transmission queue, if the channel is busy, the MAC waits until the medium becomes idle, then defers for an extra time interval called the DCF Interframe Space (DIFS). The medium is declared as busy when the energy in the medium exceeds a certain threshold known as the clear channel assessment (CCA) threshold. Each Wi-Fi AP maintains a contention window (CW), which is used to select a random backoff counter. The backoff counter value is determined as a random integer

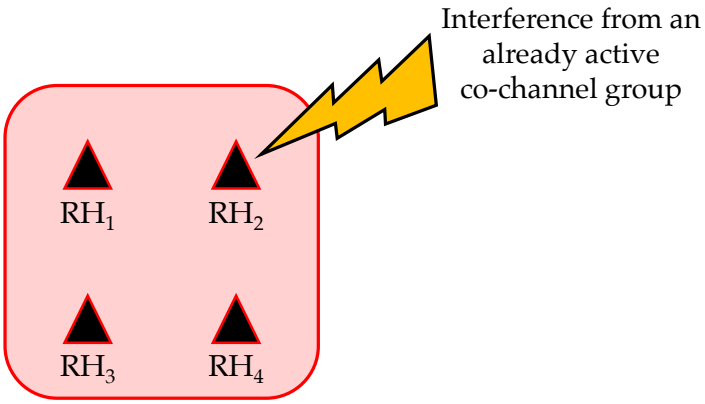


Figure 2.2: Interference seen by a D-MIMO group operating in the red channel. Notice how RH_1, RH_3, and RH_4 do not sense the interference from the active co-channel group but RH_2 does sense it.

drawn from a uniform distribution over the interval [0, CW]. If the channel stays idle during the DIFS deference, the MAC then starts the backoff process by selecting a random backoff counter. For each slot time interval, during which the medium stays idle, the backoff counter is decremented. If during the count down process, the medium is sensed to be busy, then the count down is suspended until the medium stays idle for the duration of a DIFS period. When the backoff counter value reaches zero, the frame is transmitted. On the other hand, if the medium is busy when a frame arrives at the head of the queue, the MAC waits till the medium stays idle for the duration of a DIFS interval and continues the process as described previously.

2.3.2 Channel Access in D-MIMO Wi-Fi

Since, in case of a D-MIMO group, sensing is performed with RHs (more precisely, antennas) separated geographically, DCF needs to be revisited. This is because one RH may sense a channel state busy while another RH (belonging to the same D-MIMO group as the former) may sense the same channel state as idle. For instance, consider the scenario in Figure 2.2 that describes a D-MIMO group, which is assigned the red channel, contending for channel access. Consider a co-channel group operating in the network space that is already active. Notice how RH_2 perceives the co-channel interference owing to its geographical proximity to the co-channel group, and hence declares the channel to be 'busy', but RH_1, RH_3, and RH_4 do not. This behavior is not

seen in a baseline AP with co-located antennas since the antennas are located close to each other and hence are likely to share a common view of the channel. To resolve the ambiguity in case of a D-MIMO group and to incorporate awareness of location of RHs, we will determine the channel state by centrally combining inputs from the constituent RHs. This requires that each RH report its channel state to the processing unit (PU).

We propose to generalize DCF (we will consider only one access category below) for D-MIMO as follows:

1. The processing unit (PU) maintains independent groups of one or more radio-heads (RHs) called *sensing groups* (SG). A RH can be part of multiple sensing groups. A backoff timer is associated with each sensing group.

2. All SGs' backoff timers are initialized to the same random value uniformly selected from integers in the range [0, CW], where CW is the contention window parameter.

3. A SG's channel state is determined by combining the channel states (0 if channel is idle and 1 if channel is busy) of its constituent RHs using `OR` or `AND` operators.

4. A SG's backoff timer counts down while its composite channel state is idle.

5. When a SG counts down to zero, or multiple groups count down simultaneously, they gain a TXOP. The backoff timer of all SGs shall be reset.

6. RHs associated with the winning sensing group(s) form the *transmission group* and can be used for D-MIMO transmission. We propose two concepts to facilitate enhanced channel access by D-MIMO groups: (i) combination of winning SGs, and (ii) extension of transmission group (these concepts are elucidated in more detail at the end of Section 2.3.3).

2.3.3 Strategies to Form Sensing Groups

In this section, we describe and compare three example sensing group strategies tested in simulation.

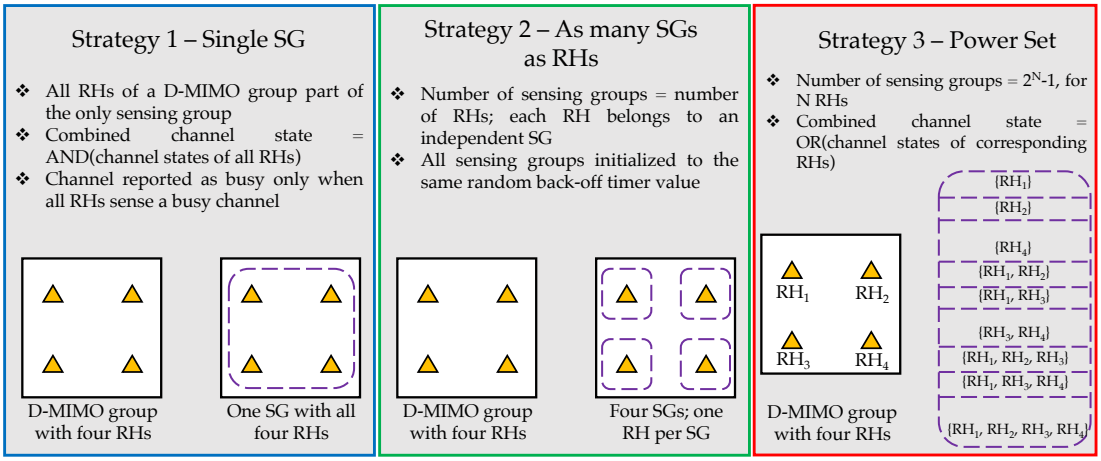


Figure 2.3: The three strategies to form sensing groups in case of a D-MIMO group with four RHs

1. STRATEGY 1: There is a single SG composed of all RHs in the D-MIMO group. Channel states of all RHs are combined using the `AND` operator, that is, the channel is reported as busy and the backoff timer is paused only when all RHs sense a busy channel. At the end of the counting-down process, the RHs that sense an idle channel will get the opportunity to transmit. It is easy to see that using the `OR` operator (instead of `AND`) makes the strategy very conservative; the back-off process is paused even if one of the RHs sense a busy channel.

2. STRATEGY 2: Each RH forms a separate SG with its own backoff timer. The backoff timers are initialized to an identical random value and are synchronized after a TXOP is obtained. When a RH senses a busy channel, the corresponding backoff timer is paused.

3. STRATEGY 3: $2^N - 1$ independent SGs are created, corresponding to the powerset of N RHs, excluding the empty set. For example, a D-MIMO group with 4 RHs has 15 SGs: $SG_1 = \{RH_1\}$, $SG_2 = \{RH_2\}$, ..., $SG_5 = \{RH_1, RH_2\}$, $SG_6 = \{RH_1, RH_3\}$, ..., $SG_{11} = \{RH_1, RH_2, RH_3\}$, $SG_{12} = \{RH_1, RH_2, RH_4\}$, ..., $SG_{15} = \{RH_1, RH_2, RH_3, RH_4\}$. A backoff timer is paused while any RH in the SG detects a busy channel, that is, channel state of a SG is determined by the `OR` operation of the channel states of its constituent RHs.

A summary of the three strategies is given in Figure 2.3 in the context of a D-MIMO group with four RHs.

For strategies 2 and 3, one might advocate use of independent random backoff timer values for SGs instead of the current choice of using the same random value. However, this will result in these strategies being aggressive while contending for a channel. For instance, strategy 3 sets $2^N - 1$ independent backoff timers, and there is a relatively high probability that at least one timer will be initialized to a small backoff time. This backoff timer dominates the behavior of the D-MIMO group, since it tends to count to zero first. With $2^N - 1$ backoff timers uniformly selected from integers in the range [0, CW], the expected minimum backoff time (in slots) is given by:

$$\mathbb{E}[T_{min}] = \sum_{k=0}^{CW} k \sum_{n=0}^{N-1} \binom{N}{n} \left(\frac{1}{CW}\right)^{N-n} \left(\frac{CW-k-1}{CW}\right)^{n}.$$

For a D-MIMO group with four RHs (15 SGs) and CW $= 15$, the expected value of the smallest backoff initialization time is only $0.52 \times 9\,\mu s = 4.68\,\mu s$. In other words, the D-MIMO group would frequently try to reclaim the channel immediately after transmission or after waiting a single idle slot. By comparison, a conventional AP with CW $= 15$ waits $67.5\,\mu s$ on average before attempting to reclaiming the channel. Hence, we chose to initialize backoff timers of all SGs to the same random value.

We introduce two additional concepts to facilitate better channel access in D-MIMO Wi-Fi, which are explained below:

1. COMBINE WINNERS: If multiple SGs in a D-MIMO group counted down at the same time and observed the channel to be idle, then the RHs belonging to all the winner SGs will be added to the transmission group. For instance, in case of strategy 3, if SG_5 and SG_7 counted down together, then the RHs belonging to both these SGs (which are RH_1, RH_2, and RH_4) will be added to the transmission group.

2. EXTENSION OF TRANSMISSION GROUP: For all strategies, a transmission group is formed when any backoff timer reaches zero, and the transmission group is extended to any additional RHs in the D-MIMO group that sense the channel as

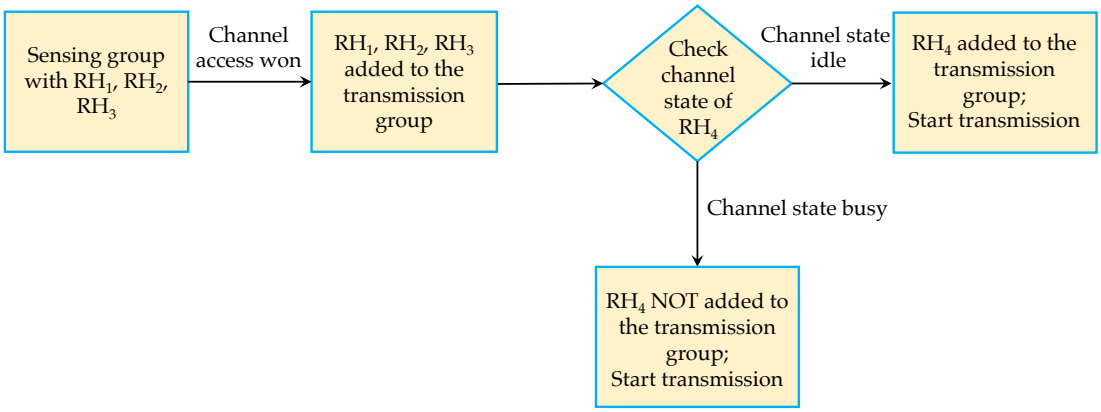


Figure 2.4: Flowchart describing the extension of a transmission group

idle. The procedure of extension is described in the flowchart shown in Figure 2.4. Assume, for instance, that SG_{11} (in case of strategy 3) completed counting down to zero first among all other SGs and hence RH_1, RH_2 and RH_3 become part of the transmission group. At this time, we observe the channel state of RH_4 (not part of SG_{11}). If the channel state is observed to be idle, we extend the transmission group by adding RH_4 to it, else the original transmission group stands. Following a transmission, we reset the backoff timers of all SGs.

Strategy 3 is the most location-aware in terms of sensing a channel's state but it necessitates maintaining a higher number of backoff timers that scales exponentially with the number of RHs N. A comparison between the three strategies to form sensing groups as well as a discussion on the benefits of implementing the aforesaid concepts of combination and extension are provided in Section 2.6.2.

2.4 Multi-User MIMO User Selection

This section discusses the motivation behind the need for a lightweight MU-MIMO user selection algorithm in D-MIMO Wi-Fi networks, and proposes a novel user selection strategy that does not request channel state information (CSI) feedback from the users during the selection phase.

2.4.1 Motivation

Once the set of RHs that won channel access in a transmission opportunity (TXOP) has been determined, the PU then has to choose the users to be served with MU-MIMO transmission in that TXOP. An 802.11ax frame can support up to eight streams to eight users. When the number of users associated with a D-MIMO group exceeds the number of streams that the group can support, it must select users to serve with MU-MIMO transmission in every TXOP in order to maximize group throughput. The choice of which users to serve is influenced by the quality of channel between the users and the RHs, the number of streams each user can receive, and the data demands of each user. For instance, if users have two antennas, a D-MIMO group may choose to serve between four users (with two streams per user) and eight users (with one stream per user) with a 802.11ax frame in a single MU-MIMO transmission.

To determine the optimal set of users to serve in a MU-MIMO transmission, which will maximize the group throughput, the PU would require accurate estimates of the downlink channels between the RHs and all the users associated with the PU/D-MIMO group. In 802.11 ac and ax, channel estimate is obtained using a channel sounding protocol with explicit compressed beamforming feedback.

The sum throughput of a D-MIMO group, after factoring in the channel sounding overhead, is given by:

$$R = \frac{D}{T_D + T_{Overhead}} \tag{2.1}$$

where D is the total number of data bits in the transmission, T_D is the duration of a TXOP, and $T_{Overhead}$ is the channel feedback overhead associated with sounding for all users computed (as described in the 802.11ac standard). Note that the duration of a TXOP is split as $T_{TXOP} = T_D + T_{Overhead}$.

Throughput is negatively impacted by channel sounding overhead and it is impractical to perform channel sounding for all associated users prior to selecting a few for each MU-MIMO transmission. To obtain a sense of how large the channel sounding overhead may be, consider a D-MIMO group with four RHs (with two antennas each) and eight associated users. If the duration of a TXOP is assumed to be 1 ms, then up

to 80% of it will be devoted to perform channel sounding between the RHs and the eight users (each receiving one stream), assuming high-resolution feedback with 122 subcarriers for channel feedback and 64 for SNR feedback. Thus, only 20% of the TXOP will be available for useful data transmission, which will bring down the total number of bits in the transmission (D) and in turn the throughput (R). Hence, it is important to choose users to serve with MU-MIMO transmission in a TXOP without incurring a prohibitively high channel sounding overhead so that group throughput may be maximized.

2.4.2 User Selection Strategies for D-MIMO Wi-Fi

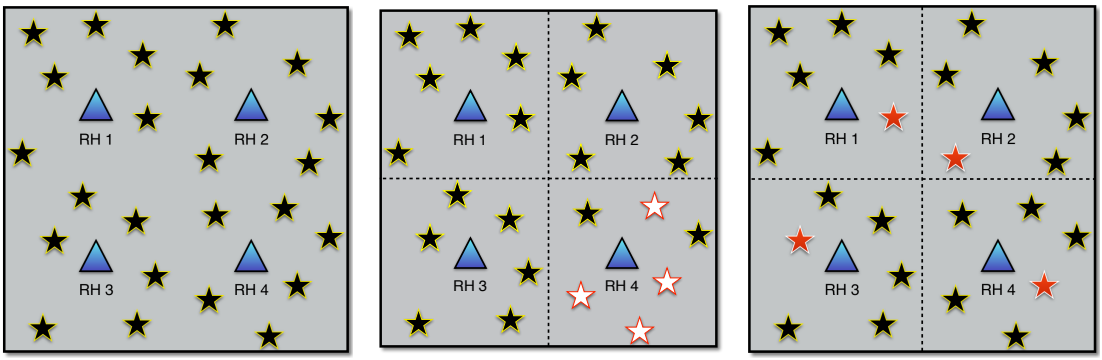
In the following discussion, we explore lightweight strategies to choose users to serve with MU-MIMO transmission in each TXOP, not by requesting explicit CSI feedback from the users, but by exploiting the spatial separation of the RHs/transmitting antennas. Assume a RH has N_t antennas and user k can receive $N_u^{(k)}$ streams. Also assume that the number of streams that can be supported by a D-MIMO group is N_S.

Random User Selection

This is the simplest strategy—randomly choose users among all users associated with the D-MIMO group until the total number of streams that can be supported by the chosen users exceeds N_S. Even though this is very easy to implement, this may not be optimal as it reduces the group throughput for two reasons: i) since users are chosen at random, it may cluster users around a few RHs (Figure 2.5b), and ii) the resulting composite channel matrix between the RHs and selected users may be poorly conditioned due to the dissimilar channel conditions between users.

Norm-based User Selection

Since Wi-Fi is a TDD system, we can exploit channel reciprocity to estimate the downlink signal strength to a user from its uplink transmissions. We claim only *weak channel reciprocity*—we assume proportionality of channel gains in downlink and uplink.



(a) A D-MIMO group (b) Random user selection (c) Proposed user selection

Figure 2.5: An example D-MIMO group with four RHs (represented as triangles). Users (represented as black stars) are distributed uniformly in space. Each image describes the users selected by the corresponding selection strategy. Notice how norm-based strategy groups users (red stars in Figure 2.5c) that are similarly spaced from RHs and how clusters of users are not formed around any RH. This is may not be the case with random selection (white stars in Figure 2.5b).

Denote the uplink channel gain from user 'a' to RH 'R' as $u_{a\to R}$ and the downlink channel gain from RH 'R' to user 'a' as $d_{R\to a}$. Weak channel reciprocity claims that if $u_{a\to R} > u_{b\to R}$, then $d_{a\to R} > d_{b\to R}$.

Norm-based user selection entails maintaining two lists: (i) a global *unassigned* list at the PU that initially consists of all the users associated with the D-MIMO group, and (ii) *assigned* list at each RH that is initially empty. The user selection algorithm is split into two phases, which are explicated below:

1. INITIALIZATION PHASE: Each RH computes the Frobenius norm of the $N_t \times N_u^{(k)}$ uplink channel matrix estimate from each user to itself. The norm of the channel matrix estimate is essentially the *channel gain*. We now pick RHs in a round-robin fashion. The selected RH chooses the user with the largest norm and moves the selected user from the unassigned list to add to its 'assigned' users list. This continues until all users are moved from the unassigned list.

2. SELECTION PHASE: The selection algorithm picks a RH in round-robin fashion and the selected RH randomly chooses a user from its 'assigned' users list. Once this user has been selected, other RHs pick users from their corresponding assigned lists that have the closest norm metric/channel gain as the previously selected

user. The selection process terminates when total number of streams that can be supported by the selected users exceeds N_D.

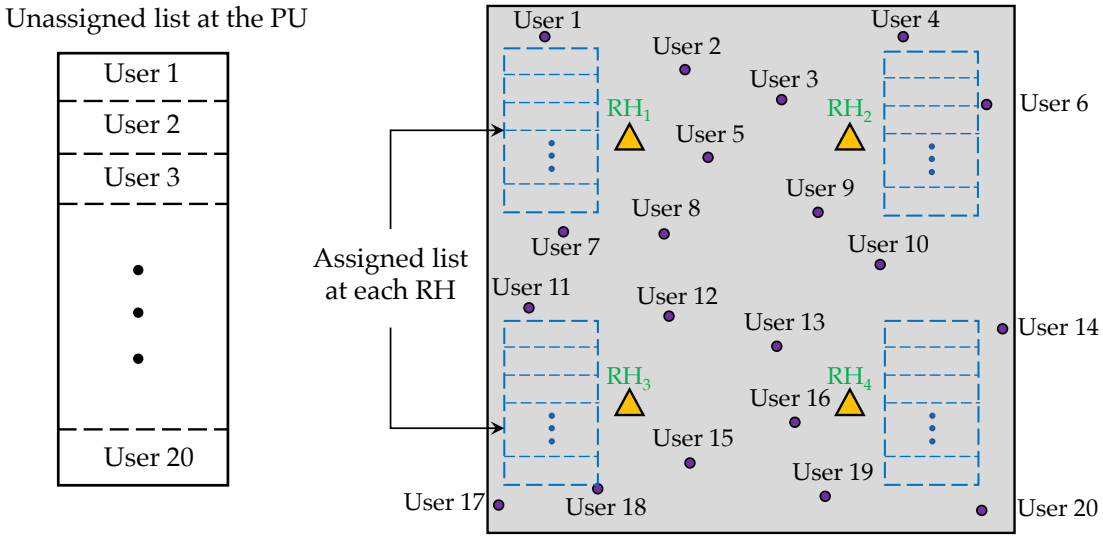
Having two lists (assigned and unassigned) ensures that two RHs do not add the same user to their respective 'assigned' users list. Furthermore, during the selection phase, since users are picked from the assigned list of each RH and the RHs are chosen in a round-robin manner, *users do not cluster around a* RH (see Figure 2.5c).

2.4.3 Explication of the Proposed User Selection Algorithm with an Example

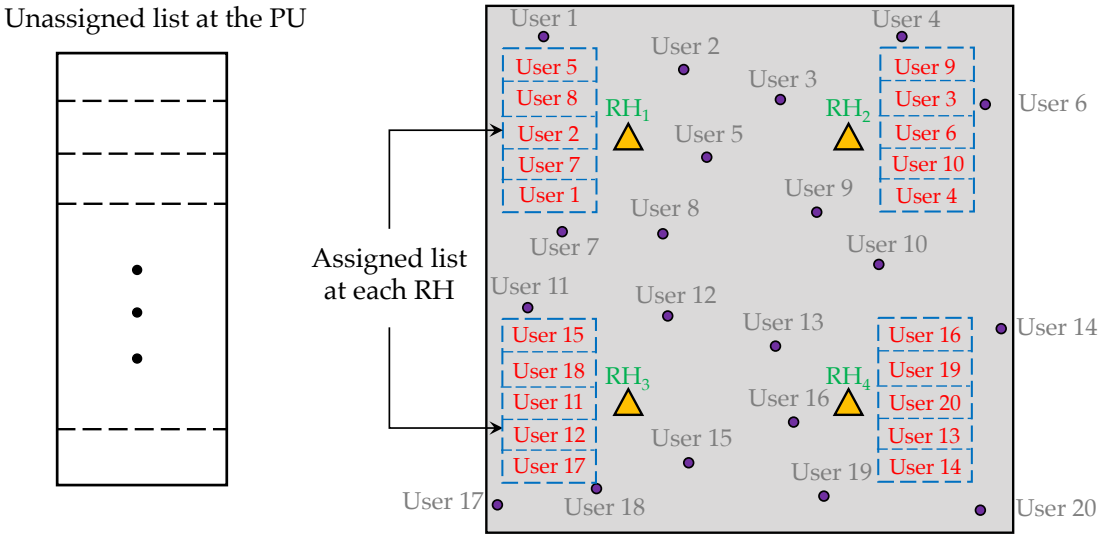
The following discussion elucidates the working of the proposed user selection algorithm. Consider the D-MIMO group shown in Figure 2.6 with four RHs and twenty users distributed uniformly in space. There are two lists maintained as part of the algorithm: a global *unassigned* list at the PU that initially contains all the users associated with the group, and an *assigned* list at each of the RHs that is initially empty.

During the *initialization phase*, each RH computes the channel gains, as discussed before, between itself and all users. Once this is complete, RHs are picked in round robin fashion to populate their respective assigned lists. Assume RH_1 gets picked first. RH_1 will choose the user that has the highest channel gain with itself *and* is also present in the global unassigned list at the PU. Assume that this user is user 2; RH_2 will move user 2 to its assigned list, and remove it from the global unassigned list. Next, let RH_2 be chosen in a round robin manner. It can no longer add user 2 to its assigned list as it is not part of the global unassigned list. RH_2 may pick from the rest of the users—barring user 2—the one with the highest channel gain to itself; for instance, it may pick user 3 to add to its assigned list. This process (round-robin selection of RHs followed by populating the respective assigned lists) continues until all users have been removed from the unassigned list maintained at the PU. This completes the initialization phase.

In the *selection phase* of the algorithm, the users to be served with MU-MIMO transmission in a TXOP will be picked. Each RH, selected in a round robin manner, gets to choose a user and add it to the list of users to be served in that TXOP (let this list be called $\mathtt{list}_{\mathtt{selected}}$). For instance, assume RH_4 gets chosen first; it might pick user 20



(a) Before the initialization phase starts. Notice the global unassigned list is full with all the users associated with the group and the assigned list at each RH is empty.



(b) After the initialization phase has been completed. Notice that all users have been moved from the unassigned list to the assigned lists oateach RH. Also observe that the assigned list of RHs do not contain any common users.

Figure 2.6: A D-MIMO group with four RHs (denoted by triangles) and twenty users. This arrangement is used to illustrate the working of the proposed light weight user selection algorithm.

randomly from its assigned list and add it to $\texttt{list}_{\texttt{selected}}$. Let the channel gain between RH_4 and user 20 be x. Next, RH_3 (selected in a round robin manner) chooses a user from its assigned list that has the channel gain closest to x and adds it to $\texttt{list}_{\texttt{selected}}$ (in this scenario, the user selected by RH_3 will be user 11). The same procedure is carried out by the other two RHs as well. Once $\texttt{list}_{\texttt{selected}}$ consists of four users, the selection process terminates for this TXOP. In the subsequent TXOPs, RHs choose users that were not served in previous TXOPs (so that all users are served fairly).

There are two major advantages to this approach of user selection:

- Since each RH chooses users from their respective assigned lists to be added to $\texttt{list}_{\texttt{selected}}$ and since the algorithm loops through all the RHs, the selected users will **not** cluster around any RH.
- Since the selected users in $\texttt{list}_{\texttt{selected}}$ have similar channel gains, (i) the resultant channel matrix will be well-conditioned, and (ii) downlink rates at the users will be similar.

Recall that a D-MIMO group can support up to eight simultaneous streams in the downlink. The discussion till now considered choosing four users per TXOP and serving each user with two streams. However, the D-MIMO group can potentially choose eight distinct users and transmit a single stream to each user. This is especially the case when users do not have high data demands and it suffices to serve a user with one stream. Additionally, each user may employ receiver diversity combining techniques to improve downlink SNR and hence the MCS index and data rates. Choosing eight users to serve, however, will incur a higher channel sounding overhead compared to choosing four users, which will in turn affect the group throughput (2.1). We study this behavior in Section 2.6.1. Furthermore, a D-MIMO group may choose to serve lower than the maximum number of streams it can support owing to other reasons (discussed in Section 2.6.1).

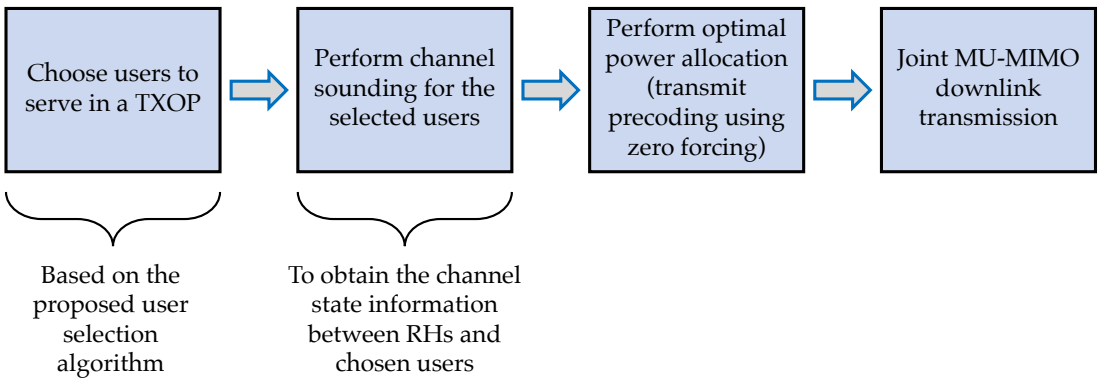


Figure 2.7: Timeline of a MU-MIMO transmission in a TXOP

2.5 Optimal Power Allocation

The timeline of a typical D-MIMO MU-MIMO transmission is shown in Figure 2.7. After determining the RHs that have won channel contention (as described in Section 2.3) and choosing users to serve with MU-MIMO transmission in a TXOP (as described in Section 2.4), the next task in the pipeline is to perform optimal power allocation among concurrent streams to (i) cancel interference between the streams, and (ii) to maximize sum group throughput, while adhering to the maximum power constraint per D-MIMO group, P_{group}. In order to nullify inter-stream interference, the PU will have to precode the transmissions from the RHs. In order to perform such transmit precoding, the PU needs to obtain the channel estimates between RHs and the users selected to be served in that TXOP, which are obtained through the procedure of channel sounding. The attractive feature of our user selection approach is that *we do not have to perform channel sounding for all the users associated with the D-MIMO group during the selection phase*—that is, channel sounding in a TXOP needs to be performed only for the users selected to be served in that TXOP.

Since a D-MIMO group can support multiple simultaneous streams in the downlink, the data symbols must be precoded to minimize interference between concurrent streams. Additionally, IEEE 802.11ac standards mandate that all streams received at a user must be with the same MCS index [36]. Since we work with a noise-normalized channel matrix, equal MCS index implies equal received power across all streams received at a user. Hence, a power optimization problem can be formulated based on

the above discussion and the formulation is described below.

Note that the following formulation is performed, for notational convenience, under the assumption that the number of streams at a user is equal to the number of receive antennas at the user. A similar optimization problem can be formulated if a user receives fewer streams than what it is capable of.

The parameters involved in the formulation of the optimization problem are defined below.

- Number of transmit antennas per D-MIMO group $= N_T$
- Number of chosen users $= C$
- Number of streams for k-th user $= N_u^{(k)}$
- Total number of streams: $N_S = \sum_{k=1}^{C} N_u^{(k)}$

The received signal at the user side can be written as

$$\mathbf{y} = \mathbf{HWs} + \mathbf{n}, \tag{2.2}$$

where $\mathbf{H} \in \mathbb{C}^{N_S \times N_T}$ is the noise-normalized channel matrix, $\mathbf{W} \in \mathbb{C}^{N_T \times N_S}$ is the precoding matrix, $\mathbf{s} \in \mathbb{C}^{N_S \times 1}$ is the transmitted symbol vector, $\mathbf{n} \in \mathbb{C}^{N_S \times 1}$ is additive noise with covariance matrix $\mathbf{I}_{N_S}$ ($\mathbf{I}_{N_S}$ is the identity matrix of dimension $N_S \times N_S$). Transmitted symbol vector $\mathbf{s}$ has covariance $\mathbf{D} \in \mathbb{R}^{N_S \times N_S}$. To ensure that all streams to a user have the same SNR (to support the same MCS index), $\mathbf{D}$ is chosen to be a block diagonal matrix:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & & & \\ & \mathbf{D}_2 & & \\ & & \ddots & \\ & & & \mathbf{D}_C \end{bmatrix}$$

where $\mathbf{D}_k = \rho_k \mathbf{I}_{N_u^{(k)}}$ and ρ_k is the per-stream SNR at user k.

We use zero-forcing (ZF) precoding to eliminate inter-stream interference by selecting $\mathbf{W}$ to be the pseudo-inverse of the channel matrix $\mathbf{H}$. Each column of $\mathbf{W}$ corresponds to the precoding vector for a stream j.

The objective of the optimization formulation is to find the optimal power allocation of streams to maximize the D-MIMO group throughput. That is, determine the SNR ρ_k for k = 1, 2, ... , C users such that

$$\arg\max_{\rho_k} \sum_{k=1}^{C} N_U^{(k)} \log_2(1+\rho_k), \tag{2.3}$$

subject to the following constraints.

1. PER-GROUP POWER CONSTRAINT: The power allocation should be performed such that the resultant sum power in all streams does not exceed the prescribed power limit for a D-MIMO group. That is,

$$\mathrm{tr}\left(\mathbf{WDW}^H\right) \leq P_{group}, \tag{2.4}$$

   where tr(.) denotes the trace operator of a matrix.

2. NON-NEGATIVITY CONSTRAINT: The resultant SNR in the streams should be non-negative. That is,

$$\rho_k \geq 0 \quad \forall\, k \in [1, C]. \tag{2.5}$$

This optimization problem can be solved efficiently using the water-filling algorithm. We emphasize here that power allocation for streams is performed after the users to be served in a TXOP have been selected and is not part of the user selection process.

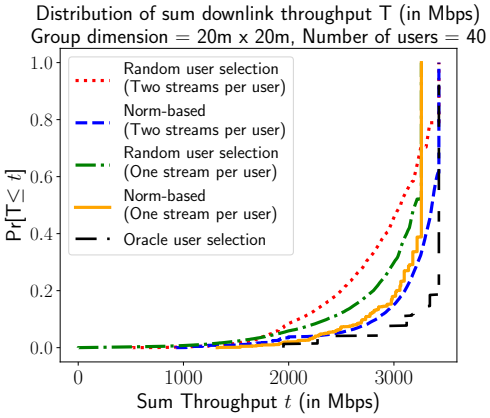
2.6 Network Simulation Results and Discussion

To study and analyze network performance quantities in case of realistic Wi-Fi networks with dense deployment of APs/RHs, we built a comprehensive *system-level event-based simulator* that integrated the framework of lightweight user selection and per-stream power allocation along with the updated channel access procedure for D-MIMO Wi-Fi. This simulator is capable of studying Wi-Fi networks in baseline as well as D-MIMO configurations. We modeled signal propagation according to the indoor path loss model in [37] with LOS and NLOS fading conditions. In this section, we first focus on the performance of one D-MIMO Wi-Fi group and quantify the improvements that the proposed user selection algorithm brings, followed by an overall network performance evaluation and analysis.

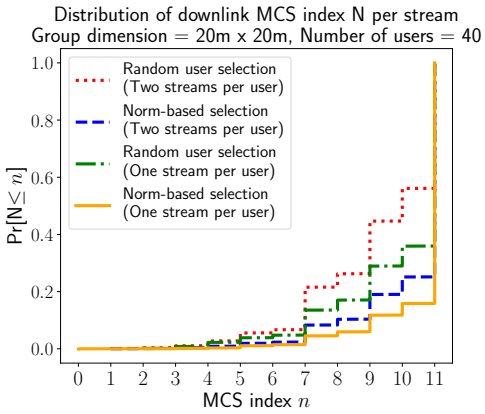
2.6.1 Results for One D-MIMO Group: Focus on User Selection

We considered a D-MIMO group of dimension 20 m $\times$ 20 m served by four RHs with two antennas each, as shown in Figure 2.5a. The RHs operated in a channel of bandwidth 80 MHz in the 5 GHz band. Let the maximum power constraint for the D-MIMO group be $P_{group} = 10$ dBm. We considered 40 users uniformly distributed in the group, that is, the density of user distribution was 40/400 i.e, 1 user per 10 m^2. Each user was equipped with two antennas. We performed selection of users in every TXOP using strategies explained in Section 2.4—(a) random selection, and (b) norm-based selection in which users were selected after computing the norms of the 2×2 estimated uplink channel. In the interest of completeness, two other strategies were also considered—(c) randomly select eight users and serve each user with one stream, and (d) select eight users from the norm-based list maintained in (b) with single stream service to each user. For strategies (c) and (d), users employed selection combining of the signals received by the two receive antennas. Additionally, we assumed the presence of an all-knowing oracle that had perfect knowledge of the channels between RHs and all the users prior to user selection and hence could *optimally* choose users to serve in every TXOP. The results from oracle user selection are used to gauge the efficacy of the proposed user selection algorithm in effectively choosing users.

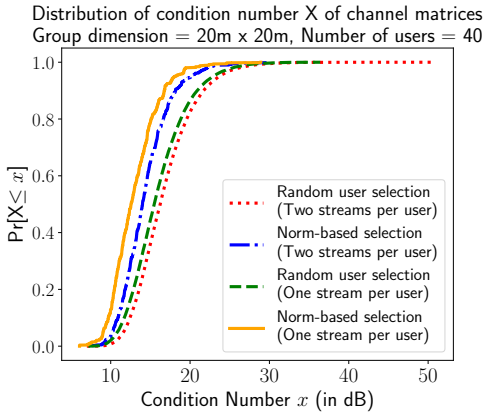
The following results were obtained by averaging over 100 random drops of users with each drop consisting of 500 TXOPs. Signal-to-noise ratio (SNR) of each stream was mapped to the corresponding modulation-and-coding scheme (MCS) index, and in turn the data rate, as per the tables in [38] prescribed for IEEE 802.11ax. The throughput plotted in Figure 2.8a accounted for the overhead of channel sounding and was computed as described in (2.1). *Note that if any streams, after optimal power allocation, could not support the minimum required SNR for MCS 0 (= 3.9 dB from tables in [38]), then such streams were not served in that TXOP and the power allocation algorithm was run again with the updated set of streams.* Rerunning the power allocation algorithm was not computationally prohibitive owing to the fact that we used water-filling to solve the optimization problem in Section 2.5 efficiently.



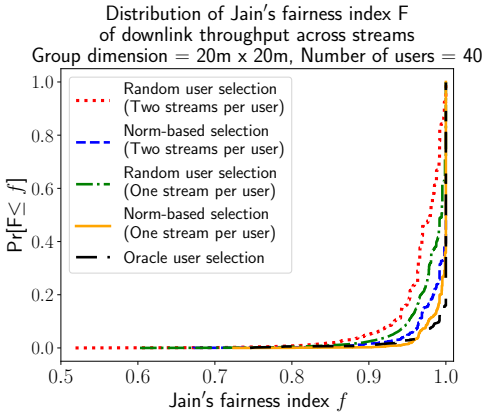
(a) Distribution of sum throughput of the group



(b) Distribution of MCS indices chosen by streams



(c) Distribution of condition number of channel vectors



(d) Distribution of Jain's fairness index of throughput across streams chosen in a TXOP

Figure 2.8: Characteristics for one D-MIMO group, described in Figure 2.5a, with 4 RHs and 40 users. The group can support 8 simultaneous downlink streams. Each plot is a cumulative distribution function (CDF).

From Figure 2.8a, it can be seen that random user selection (with each user receiving two streams) yielded the poorest throughput performance compared to the other strategies. This could be ascribed to the fact that random selection might result in user groupings in which users clustered around a RH. This would result in conservative power allocation to null out interference between streams to such users. This, in turn, led to poor downlink SNR in these streams and hence poor rates achieved by these streams. Choosing users according to the norm-based strategy (b) worked well; it resulted in user groups achieving a gain of more than 15% in median throughput compared to random selection. Note that strategy (b) would be preferred if users

have more data to receive, otherwise strategy (d) may be employed. Also, observe the difference in peak sum throughput numbers obtained by strategies (b) and (d). This can be explained by the fact that channel sounding for higher number of users would have to be performed for strategy (d), which lowered the group throughput (2.1). It can also be observed that strategy (b) achieved throughput characteristics close to oracle user selection (median throughput difference of only 1.58% with oracle selection), that is, choosing streams using the proposed algorithm was close to optimality. Of course, it may be argued that the algorithm is not fully optimal. This is the trade-off between optimal user selection and the entailed channel estimation overhead—collecting CSI information from all users associated with the D-MIMO group and then choosing users optimally versus a close to optimal user selection without collecting CSI feedback from all the users associated with the D-MIMO group.

Figure 2.8b plots the distribution of MCS indices chosen by the streams based on the signal-to-noise ratio (SNR) in the downlink. It is interesting to note that strategy (d) chose the maximum MCS index (= 11) more than 85% of the time, which indicates that selecting users with single stream service led to high downlink SNR after optimal power allocation, thanks to receiver combining gains. This is, in fact, better than the MCS indices chosen when strategy (b) was used. The receiver diversity technique that each user implemented was equal gain combining. However, this result should be treated with caution as this did not correspond to a similar trend in group throughput (as explained previously and seen in Figure 2.8a). Strategy (b) resulted in the choice of the highest MCS index more than 75% of the time. The takeaway from this discussion is that streams could support excellent MCS indices even when the choice of users to serve in a TXOP was performed without requesting CSI feedback from all the users associated with the D-MIMO group.

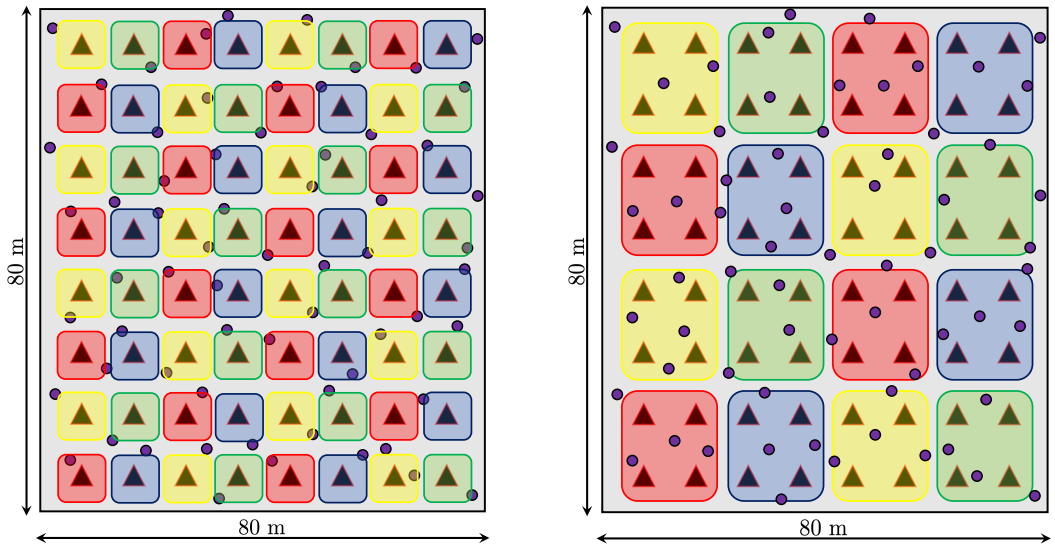
Figure 2.8c and Figure 2.8d describe a set of plots to further demonstrate the effectiveness of the proposed user selection algorithm. In case of a MIMO system, independent streams between a transmitter and receiver are realized when the channel between them is 'good', that is, if the channel matrix can be decomposed to independent eigenchannels. This is quantified by a parameter called the condition number, which is

defined as the ratio of the maximum and minimum eigenvalue of the channel matrix; smaller this number, more well-conditioned the matrix will be. Figure 2.8c describes the distribution of condition number of channel matrices chosen by different strategies through the course of the simulations. It is seen that norm-based methods produced a good distribution of condition numbers compared to random ones. Figure 2.8a, which describes the distribution of sum throughput of a D-MIMO group, does not convey the full story; the sum throughput could be high because a few of the chosen streams achieved high throughput while others achieved poor values. This is undesirable. To quantify such a bias, Figure 2.8d plots the distribution of Jain's fairness index [39] among throughput achieved by streams simultaneously served in a TXOP. It is seen that choosing users based on strategy (d) resulted in a fairness index of 1 among the streams more than 90% of the time. It is also evident that random selection achieved the least fairness of throughput among simultaneous downlink streams.

Furthermore, recall that although a D-MIMO group can support up to eight simultaneous downlink streams, it may choose not to do so if streams achieved an SNR lower than what is required for MCS0 after optimal power allocation. We noticed, from our simulation runs, that norm-based strategies chose eight streams to serve simultaneously almost always ($\sim$ 97% of the time) whereas random strategies served less than eight around 11% of the time.

2.6.2 Network Simulation Results

This section describes the results obtained from the network simulations of both baseline and D-MIMO arrangements. We considered an office space of dimensions 80 m $\times$ 80 m in which several two-antenna users were uniformly distributed. In case of the baseline deployment, 64 two-antenna APs were deployed at a distance of 10 m from each other (see Figure 2.9a), while in the D-MIMO configuration (Figure 2.9b), 16 groups were formed with four two-antenna RHs each. In each D-MIMO group, selection of users to serve in a TXOP was performed according to the norm-based strategy described in Section 2.4 with each user getting two streams and optimal power allocation of RHs was performed as described in Section 2.5. The following results



(a) Baseline scenario (b) D-MIMO scenario

Figure 2.9: Network simulation scenarios for baseline and D-MIMO arrangements. Triangles represent APs/RHs and circles represent users. The channel assigned to a AP or a D-MIMO group is identified by color.

were obtained from 3000 simulations with a different random drop of users in each simulation. Each simulation was run for a network time of 100 ms. The settings of different parameters involved in the simulations are listed in Table. 2.1.

Channel Access Characteristics

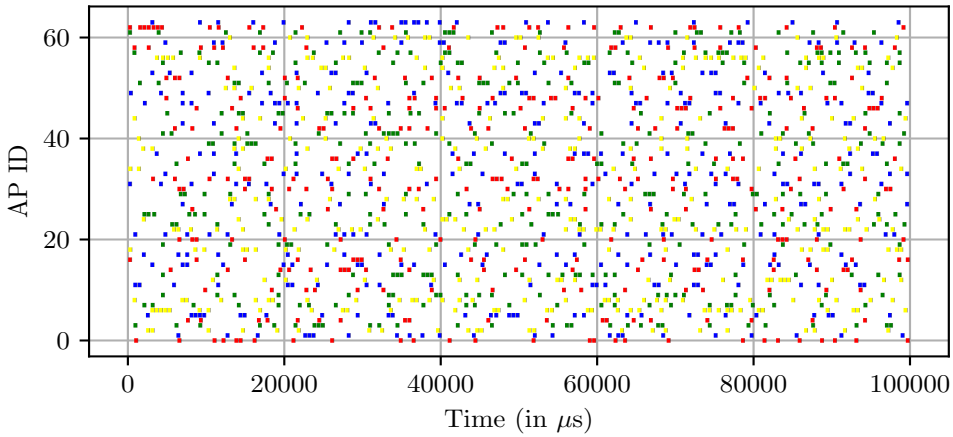
The traffic model at the AP/RH side was assumed to be full buffer with a contention window size of 15. There were four 80 MHz channels available for contention and access, each represented by a unique color (channel assignments of APs and D-MIMO groups are described in Figure 2.9a and Figure 2.9b respectively). Figure 2.10a and Figure 2.10b describe how much channel access (in time) each AP/D-MIMO group obtained in a network simulation run of time 100 ms. The x-axis in these plots bear the simulation time instants and the y-axis plots the index of AP/D-MIMO group. The length of a bar in these figures is indicative of the duration of channel access attained by an AP/RH. For the current scenario of interest, our simulations showed similar performance for strategies 1 and 2 described in Section 2.3.2, and so strategy 1 was preferred because it has the simplest implementation (a comparison between the

Table 2.1: Details of the network simulation setup

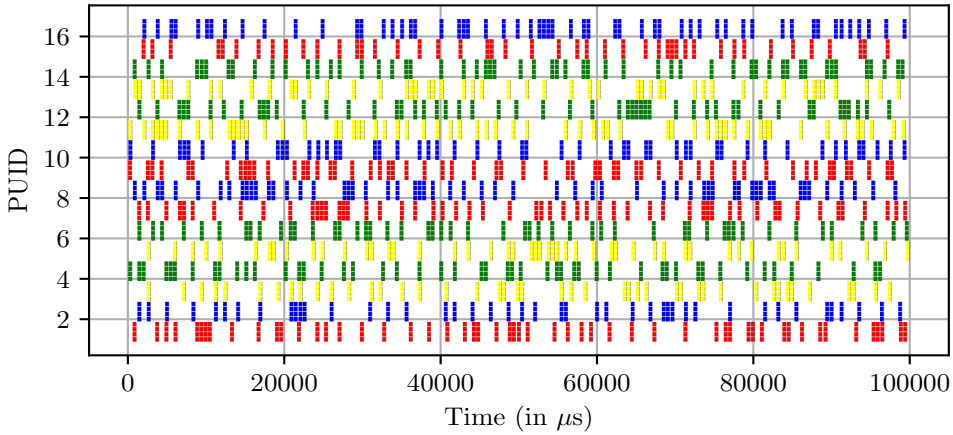
Parameter	**Value**
Channel center frequency	5 GHz
Channel bandwidth	80 MHz
Number of available channels	4
Path loss model	802.11 TGn model [37]
Number of APs/RHs	64
AP/RH inter-site distance	10 m
Number of RHs per D-MIMO group	4
Number of D-MIMO groups	16
Number of antennas (per device)	2
Directionality of antennas	Isotropic
AP/RH height	3 m
User height	1 m
Group power constraint	20 dBm
Traffic model	Full buffer
SINR to MCS mapping	IEEE 802.11ax [38]
Clear channel assessment (CCA) parameters	
CCA threshold	−82 dBm
DIFS duration	34 µs
SIFS duration	16 µs
Slot duration	9 µs

different strategies is provided in Section 2.6.2).

It is seen that with the baseline configuration, APs did not gain access to channels frequently (bars are short and spaced far apart) as seen in Figure 2.10a. Observe from Figure 1.2a that *hearing range* of baseline AP included at least six other APs in the same channel (in our current simulation setting). This led to a higher competition for channel access and lower channel access times. In contrast, the channel access times were notably better for D-MIMO configuration. Observe that the bars are consistently longer and are spaced at small regular intervals in Figure 2.10b. This is due to the fact that the composite hearing range of a D-MIMO group (bold dotted circle in Figure 1.2b) did not include any other group operating in the same channel. Hence, the proposed channel access procedure for D-MIMO helped RHs achieved better access to the channels compared to baseline APs. The distribution of channel access times, which is plotted in Figure 2.13a, indicate that mean channel access time for a RH in a D-MIMO group was 61% lower than the corresponding number for a baseline AP.



(a) Baseline scenario



(b) D-MIMO scenario

Figure 2.10: Channel occupancy of APs/RHs plotted as line diagrams. The length of a line corresponds to the duration for which the corresponding AP/RH was able to access a channel. Each channel is color-coded uniquely.

Comparison of Strategies to Form Sensing Groups

The following discussion compares the three strategies to form sensing groups (SGs), described in Section 2.3.3 in terms of percentage of channel occupancy, average channel access delay, obtained user throughput. We also demonstrate the benefits of combining the winning SGs and extending the transmission group (as described at the end of Section 2.3.3).

Table 2.2 and Table 2.3 describe the channel access results compiled when inter-RH distances were 10 m and 25 m respectively. In the tables, SG strategy indicates the

Table 2.2: Channel access characteristics when inter-RH distance was 10 m

SG strategy	Operation	Combine winners	Extension of transmission group	Average channel occupancy	Average user throughput (in Mbps)	Average channel access delay (in ms)
Strategy 1	OR/AND	Not applicable	False/True	26.5%	191.306	1.361
Strategy 2	Not applicable	False	False	18.704%	69.028	2.091
Strategy 2	Not applicable	True	True	26.5%	191.306	1.361
Strategy 3	OR/AND	False/True	False/True	26.5%	191.306	1.361

Table 2.3: Channel access characteristics when inter-RH distance was 25 m

SG strategy	Operation	Combine winners	Extension of transmission group	Average channel occupancy	Average user throughput (in Mbps)	Average channel access delay (in ms)
Strategy 1	OR	Not applicable	True	27.72%	160.021	1.278
Strategy 1	AND	Not applicable	True	30.49%	177.729	1.103
Strategy 2	Not applicable	False	False	16.59%	107.28	2.068
Strategy 2	Not applicable	True	True	30.54%	176.22	1.109
Strategy 3	OR/AND	False/True	False/True	30.64%	179.29	1.101

strategy used to form SGs, and operation denotes how the channel sensing observations of different RHs belonging to a SG were combined (the rest of the fields are self-explanatory). For strategy 1, the 'combining winner' field is not applicable since there exists only a single SG. For strategy 2, since there exists only one RH per SG, the operation field is not applicable.

For the network scenario described in Table 2.1 (distance between RHs = 10 m in Figure 2.9b), the three strategies performed identically (results compiled in Table 2.2). However, combining the winning SGs and extending the transmission group led to a discernible increase in performance in case of strategy 2—an increase of 42% in channel occupancy, and a reduction of 35% in channel access delay. Recall that strategy 2 maintained as many SGs as the number of RHs with one RH per SG. Hence, if the transmission group was not extended, then only one RH would have obtained the chance to transmit in a TXOP. Table 2.3 presents the results when inter-RH distance was set to 25 m. The results for strategy 1 indicate that combining channel states of RHs using OR operator led to a conservative performance, since the group channel state would have been declared as busy even if just one of the RHs sensed the channel to be busy. Notice the improvement in performance that combining winning SGs as well as extension of transmission group brought to strategy 2—an improvement of 84% in channel occupancy, 64% in user throughput, and a reduction of 46% in channel access delay. Also, note that the performance of the three strategies were similar when both

Table 2.4: Summary of results obtained from all network simulations

Network performance quantity	Baseline	D-MIMO	Change
Mean user throughput	63.893 Mbps	185.922 Mbps	190.99% improvement
Ten percentile user throughput	26.47 Mbps	117.827 Mbps	345.134% improvement
Mean MCS index (rounded)	5	7	40% improvement
Mean channel access delay	3.481 ms	1.344 ms	61.39% reduction

the 'combine winners' and 'extension of transmission group' fields were enabled. If this was not the case, strategy 3 achieved a better performance compared to strategy 2, which in turn achieved a better performance than strategy 1. This observation can be ascribed to the better location awareness achieved by strategy 3 compared to the other two methods. However, note that strategy 3 requires maintaining a large number of SGs that scales exponentially with the number of RHs in a D-MIMO group.

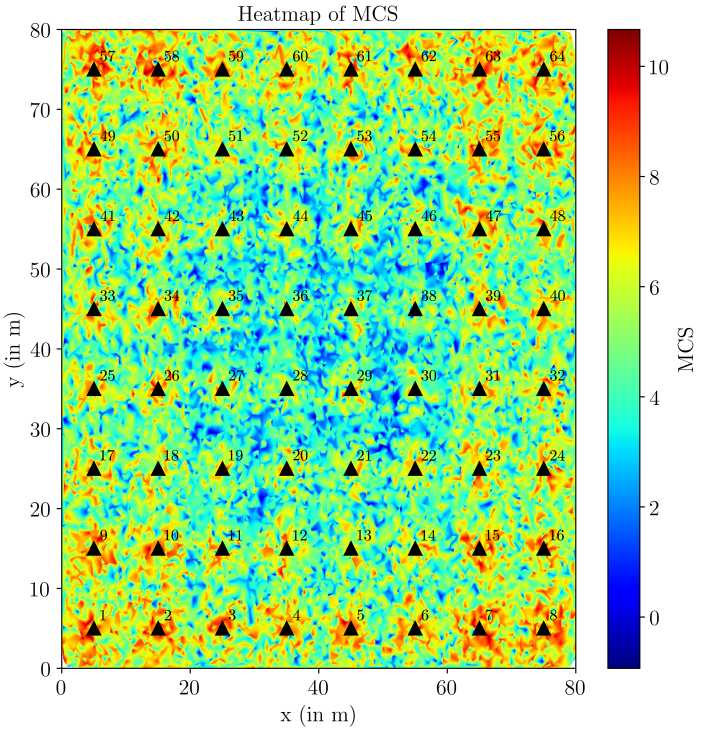
User Throughput Performance

Figure 2.11a and Figure 2.11b describe the spatial distribution of the mean MCS indices achieved by the users across all 3000 simulation runs with baseline and D-MIMO setups respectively. Since the network simulation time was longer than the duration of one TXOP, the mean MCS index for a user 's' was computed as the average of MCS indices observed across all TXOPs in which the user 's' was active in that simulation run. Likewise, Figure 2.12a and Figure 2.12b depict the spatial distribution of throughput achieved by users in case of baseline and D-MIMO configurations respectively. These figures are color-coded such that blue regions obtained lower values compared to orange/red regions. It can be seen that D-MIMO achieved a discernibly better performance in both MCS and throughput trends in the whole office space.

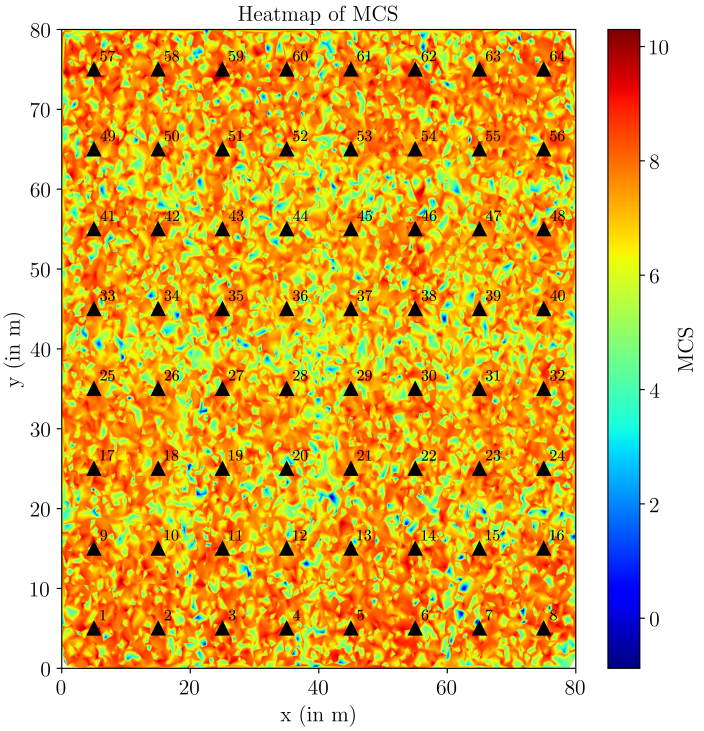
The cumulative distribution of throughput and MCS index achieved by the users in case of baseline and D-MIMO configurations are plotted in Figure 2.13c and Figure 2.13b respectively. It can be seen that D-MIMO achieved a gain of 3.5$\times$ gain in median and an increase of 191% in mean per-user throughput compared to baseline. From Figure 2.13b, it can be seen that in D-MIMO configuration, users picked an MCS index of more than or equal to 7 about 50% of the time as compared to baseline deployment in which

users picked an MCS index of less than or equal to 5 about 57% of the time. These results along with the heatmaps (in Figure 2.11 and Figure 2.12) advocate strongly for the implementation of D-MIMO as a technology to significantly improve throughput performance of users in dense Wi-Fi networks.

Table. 2.4 summarizes the results obtained from all the network simulations. Across all network performance quantities, D-MIMO achieved better numbers compared to baseline.

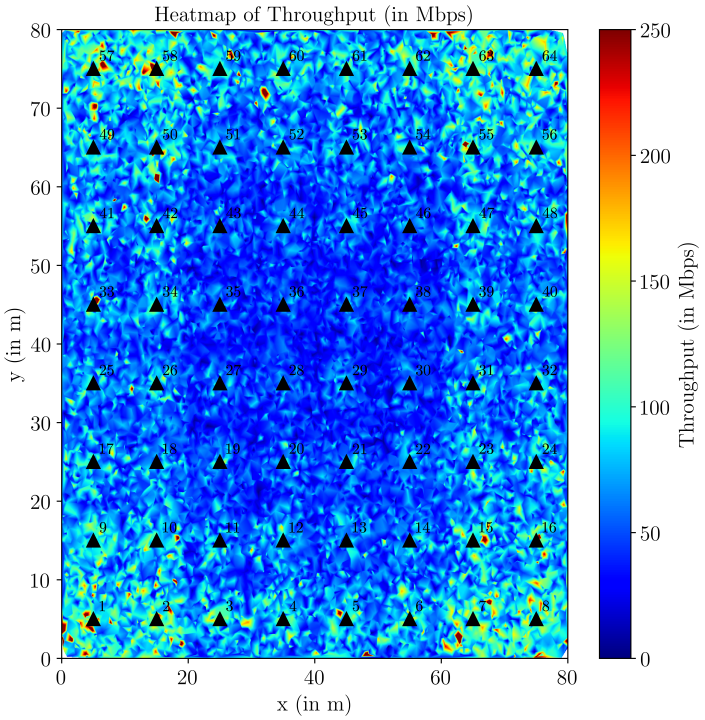


(a) Baseline scenario

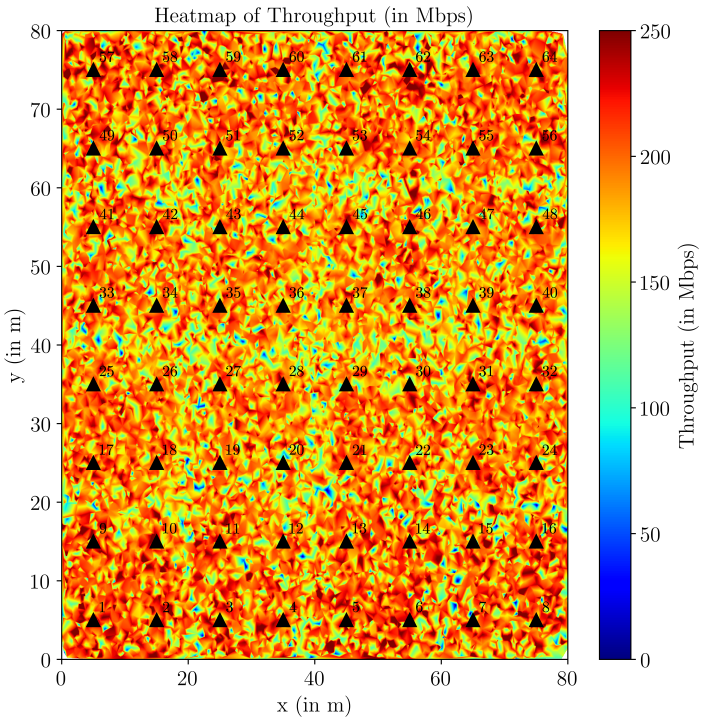


(b) D-MIMO scenario

Figure 2.11: Heatmap of mean MCS index achieved by users in the network from all simulation runs.

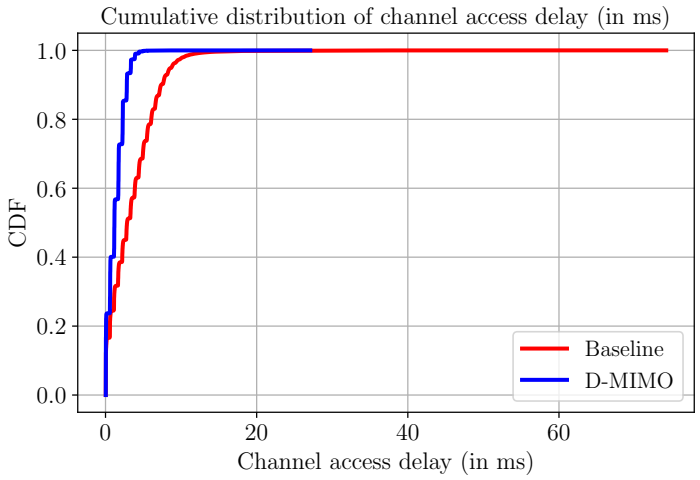


(a) Baseline scenario

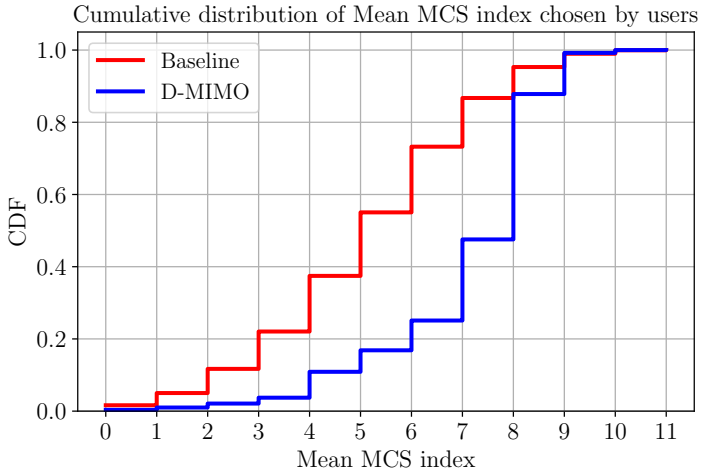


(b) D-MIMO scenario

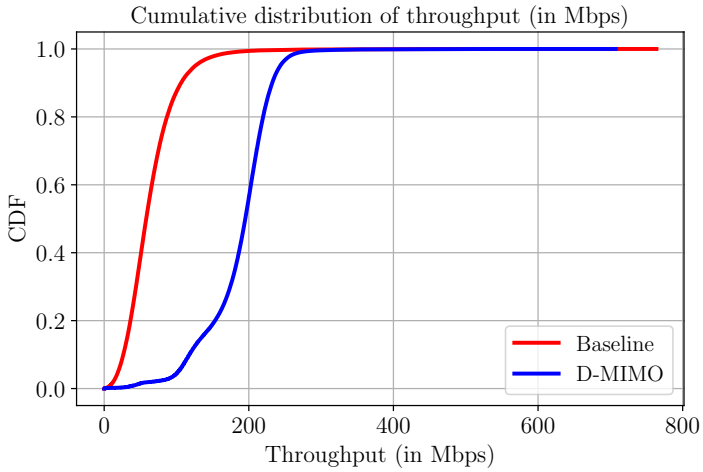
Figure 2.12: Heatmap of throughput achieved by users in the network from all simulation runs.



(a) Distribution of channel access delays



(b) Distribution of mean MCS index achieved by users



(c) Distribution of per-user throughput

Figure 2.13: Comparison of network performance quantities of baseline and D-MIMO configurations obtained from network simulations. Each plot is a cumulative distributive function (CDF).

Chapter 3

Implementation of D-MIMO Wi-Fi in an Indoor Testbed

3.1 Summary and Organization

This chapter describes the implementation of a D-MIMO group in an indoor testbed using software defined radio platforms. This setup was used as a proof-of-concept to demonstrate the merits of the lightweight user selection algorithm proposed in Chapter 2, Section 2.4. This chapter first reviews existing literature in the realm of D-MIMO system implementations using experimental platforms in Section 3.2. This is followed by a detailed description of our implementation of a D-MIMO Wi-Fi group using software defined radios in Section 3.3, including details of the hardware used, and the timeline of an experimental run. The results obtained from extensive experimental evaluations on the deployed setup are discussed in Section 3.4.

3.2 Literature Review

Practical system designs and implementations of D-MIMO are detailed in [9–12, 14], which describe the merits of using distributed MU-MIMO over deploying independent APs with co-located antennas. However, these works did not propose algorithms to choose users to maximize group throughput. Furthermore, they discussed the implementation in case of a single D-MIMO group and did not consider inter-group interference or how channel access would be mediated when multiple co-channel D-MIMO groups were present. The primary focus of these works was to achieve tight synchronization among APs/RHs either (i) over-the-air by employing a master-slave network architecture, that is, the slave APs/RHs synchronize to the reference/pilot signals transmitted by a master AP/RH [9–12, 14], or (ii) by connecting the APs/RHs to

Table 3.1: Details of the experimental setup

Parameter	Value
Number of RHs	4
Number of users	20
Number of antennas	2 per device
Max number of downlink streams	8
PHY layer parameters	
Wi-Fi standard	IEEE 802.11ac (VHT mode)
Guard interval	400 ns
Center frequency	5.4 GHz
Channel bandwidth	20 MHz

a common external clock [15], or (iii) by creating a tracking scheme to account for signal phase drifts between the APs [13]. Authors in [13, 15] proposed algorithms for user selection in D-MIMO but these algorithms were either computationally demanding (and needing channel state feedback from the users) or less efficient in terms of serving as many users in a transmission opportunity as possible. This chapter does not propose novel methods to facilitate synchronization among APs/RHs. The focus instead is to evaluate the effectiveness of the lightweight user selection algorithm proposed in Chapter 2, Section 2.4. To serve this purpose, synchronization among RHs may be established by any mechanism—either over-the-air or by connecting them to the same external clock. We adopt the latter method in our work because we believe that connecting RHs of a D-MIMO group to a common external clock, particularly in an enterprise network setting, is a plausible assumption.

3.3 Implementation of D-MIMO Wi-Fi Using Software Defined Radios

We implemented a D-MIMO Wi-Fi group in the main grid of the ORBIT indoor testbed [40] (see Figure 3.2). The topology of the experimental setup was as shown in Figure 3.1. Each node used in the implementation consisted of a computing unit, a software defined radio (SDR), and antennas connected to the SDR. The SDRs used for the current implementation were universal software radio peripheral (USRP) B210s and X310s [41]. Various specifics about the experiment and the associated hardware are tabulated in Table 3.1 and Table 3.2.

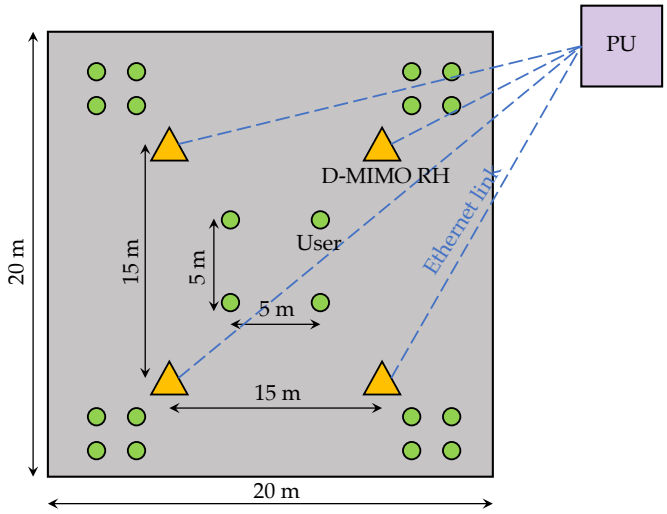


Figure 3.1: Implementation of a D-MIMO group in the ORBIT testbed. RHs are denoted by triangles and deployed using USRP B210s. Users are represented by circles and are deployed using USRP B210s (center nodes) and USRP X310s (corner nodes).

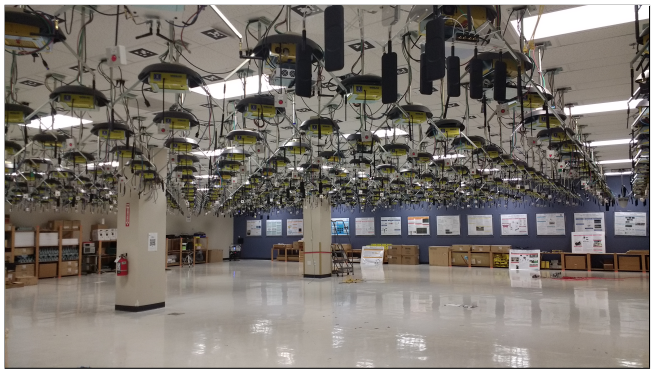
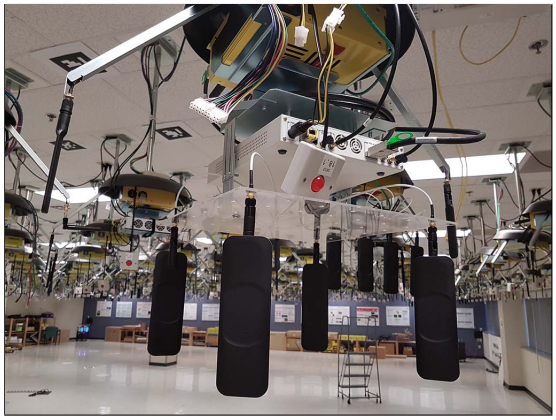


Figure 3.2: Picture of the indoor ORBIT testbed

The RHs were implemented using USRP B210s. The nodes used for the RHs were synchronized in frequency and phase using external OctoClock-G [42], which is a GPS-disciplined clock reference system. To tightly synchronize the daughterboards across the four RHs such that transmissions from these RHs occur at exactly the same time and clock tick, an external PPS input of the OctoClock-G was fed to all the RHs. The OctoClock-G used an internal clock source to consistently generate a 10 MHz reference signal as well as 1 PPS signal. The OctoClock-G also behaved as a distributor of the 10 MHz clock signal and the 1 PPS signal, when the PPS input switch in the USRP was switched to external. More details of the clock reference distribution system implemented in ORBIT can be found in [43]. The central PU required for co-ordinating

(a) USRP B210



(b) USRP X310

Figure 3.3: Pictures of the USRPs used in the implementation along with the attached antennas

Table 3.2: Details of hardware used in the experiments

	RH	**User**
USRP	B210	B210s/X310s
Antenna	Apex II TG.35 4G Wideband	Apex II TG.35 4G Wideband (*for B210s*) Abracon RHAMSJ-137 (*for X310s*)

the operation and transmissions of the RHs was implemented using a server node in the testbed. The server consisted of an Intel Xeon CPU E5-2630 v3 with a clock speed of 2.4 GHz and 120 GB hard drive capacity. The RHs were connected to the PU with 1G ethernet links.

The users in the D-MIMO group were implemented using USRP B210s and X310s. The user nodes were not synchronized with each other nor with the RHs (as is in the real world case) in frequency or time. The two antennas, however, of each user node were synchronized by an internal clock.

The implementation, being compliant to IEEE 802.11ac standards, used OFDM for channel sounding and data transmission and used a packet/frame structure in adherence with the standards. Preamble based packet detection was used in which short and long preamble were concatenated to the start of each OFDM packet. Short preamble was used for packet detection and long preamble was used for determination and suppression of *carrier frequency offset* (CFO) and *sampling frequency offset* (SFO)

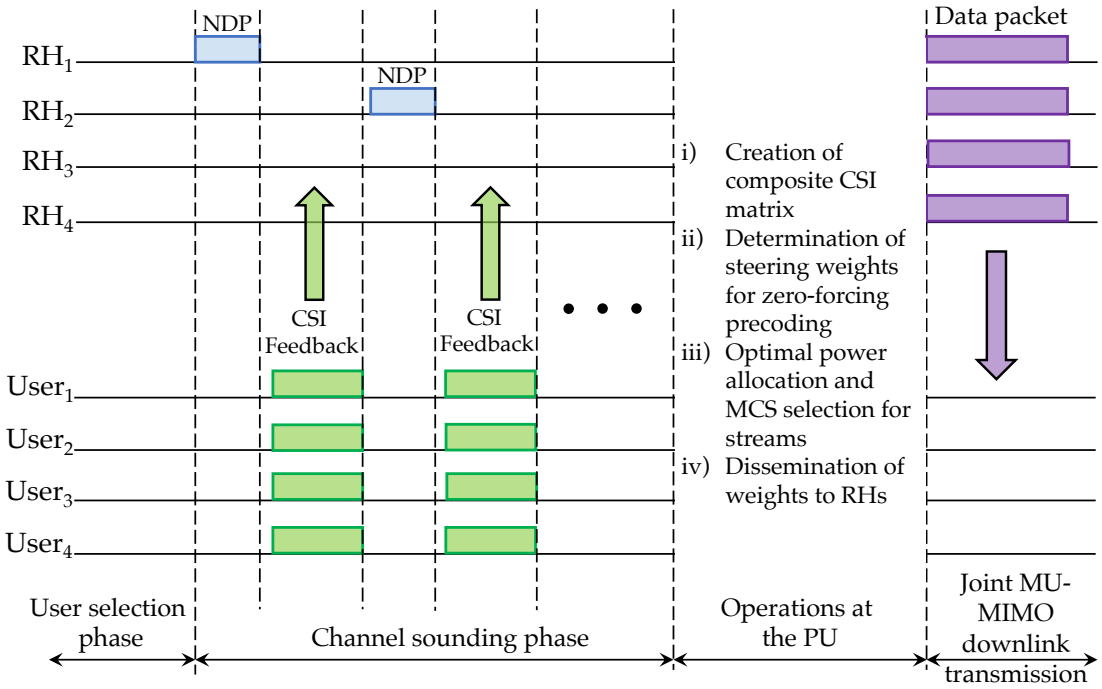


Figure 3.4: Timeline of an experimental run; each run lasts for the duration of one TXOP

between RHs and users.

3.3.1 Timeline of One Experimental Evaluation

Figure 3.4 describes the timeline of an experimental run. The duration of an experimental evaluation (i.e, the duration of a transmission opportunity, TXOP) was divided into three phases. In the *user selection* phase, the PU selected the users to serve with MU-MIMO transmission in that TXOP as described in Section 2.4. To study the benefits of the proposed user selection algorithm, the PU also performed user selection randomly, i.e, it chose four random users to serve in every TXOP. Note that, in the experimental runs, each user was served with two streams always and hence the PU chose four users to serve in each TXOP. However, the implementation can easily be extended to serve users with variable number of streams as well. Once the users to be served in a TXOP were selected, actual data transmission to the users occured in two phases: (i) *channel sounding phase* to obtain channel state information (CSI) between each RH-user pair, and (ii) *joint MU-MIMO downlink transmission phase*.

We note here that the user locations in the testbed were fixed and the users were

stationary. The experiments were carried out in the testbed at night when there was minimal dynamicity in the environment and hence the coherence time of the wireless channel was large.

Channel Sounding Phase

As described in Section 2.5, since a D-MIMO supported multiple users simultaneously, the RHs would have to precode their transmissions in order to suppress inter-user interference. To perform precoding, the RHs required accurate estimates of channel state information (CSI) between themselves and the users selected to be served in a TXOP. In 802.11 ac and ax, such channel estimates are collected through the procedure of channel sounding. Each RH broadcast a null data packet (NDP) to request CSI information from the users chosen to be served in the TXOP. Note that, according to the IEEE 802.11ac standards [36], the addresses of the users to be sounded are present in NDP announcement (NDPA) packet and not the NDP, but in our work, we updated the NDP packet structure to contain the user addresses. The presence of an Oracle was assumed, an all-knowing system, that could carry CSI from the users to the RHs (and, in turn, the PU) instantaneously without loss or degrading of information, and without the use of the conventional wireless channel. The PU concatenated the CSI matrices between each RH-user pair into a composite CSI matrix of dimension = number of receive antennas $\times$ number of transmit antennas $\times$ number of OFDM subcarriers.

MU-MIMO Downlink Transmission Phase

The MU-MIMO downlink transmission to users involved three steps:

1. DETERMINATION OF STEERING MATRIX/WEIGHTS: Since the D-MIMO group served multiple streams concurrently, the transmissions from the RHs had to be precoded in order to nullify inter-stream interference. The PU determined the precoding/steering matrix by computing the pseudo-inverse of the composite CSI feedback matrix (determined as described previously), assuming zero-forcing (ZF) as the transmit precoding technique.

2. OPTIMAL POWER ALLOCATION: The PU performed power allocation (as described in Section 2.5) to optimally allocate power to the concurrent streams in order to maximize group throughput performance subject to the maximum per-group power constraint. The results of the power allocation algorithm were the interference-free SNRs (per sub-carrier of the channel) of the spatial streams.

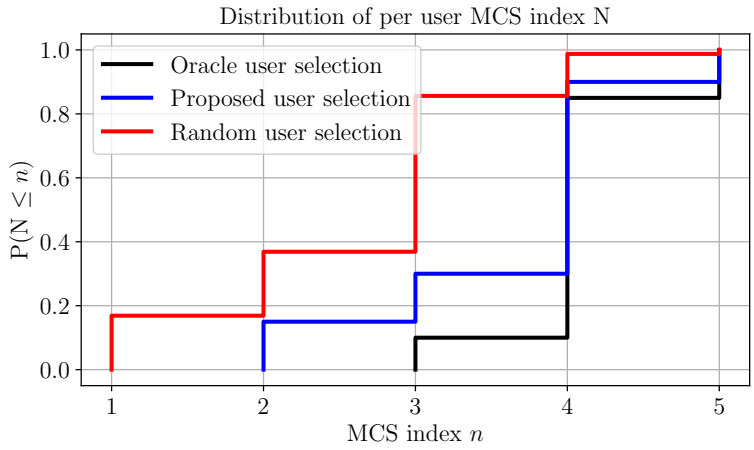
3. RATE SELECTION FOR USERS: The per sub-carrier SNRs in a stream were then combined to obtain an ***effective SNR*** [44] for the stream that was then used to choose the modulation and coding scheme (MCS) index for transmissions in the stream (according to the tables in [36]). The MCS index decides the rate and type of modulation to be used for transmission. IEEE 802.11ac standards mandate that all streams received at a user must be with the same MCS index [36]. Note that this requirement was included as a constraint in the optimal power allocation optimization described in Section 2.5.

After the aforementioned steps were completed, the PU disseminated the steering weights to the corresponding RHs and the RHs performed joint multi-user downlink transmission to the users selected in that TXOP. We note here that all operations at the PU (computation of steering weights, optimal power allocation, and MCS selection for streams) were performed off-line. The stationary nature of the users and the static nature of the testbed environment justify off-line processing at the PU.

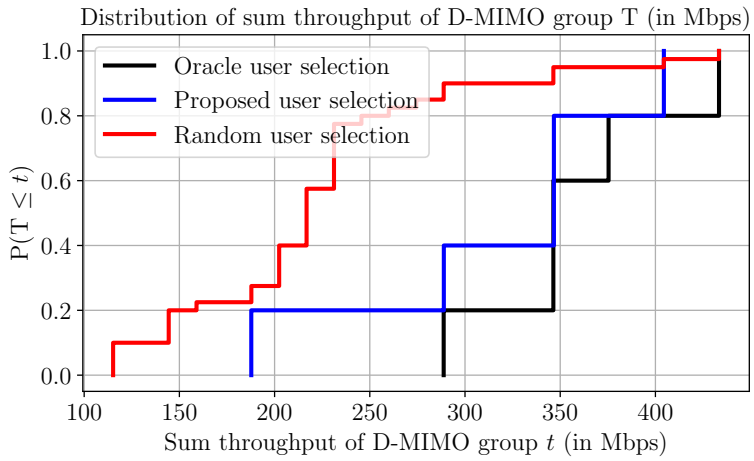
Once a user received the joint transmissions from the RHs, it first corrected for the CFO and SFO between itself and the RHs using the short and long preambles of the received packet. Recall that all the RHs were connected to the external clock and hence no frequency or sampling offset existed between the RHs. After the offset correction, the users demodulated the received packets, computed the corresponding bit error rates, and eventually the throughput. The users performed packet demodulation off-line.

3.4 Results from Experimental Evaluations

This section discusses the results obtained from extensive experimental evaluations on the implemented D-MIMO group. Experiments followed the timeline as shown in



(a) Distribution of MCS index chosen by users



(b) Distribution of sum throughput of the D-MIMO group

Figure 3.5: Results obtained from experimental evaluations performed on the D-MIMO Wi-Fi group. Each plot is a cumulative distribution function (CDF).

Figure 3.4. The users to be served in each TXOP were selected using two strategies: (i) random user selection, and (ii) the user selection algorithm proposed in Section 2.4. The experiments were performed at night in order to minimize dynamicity of the indoor environment and in the channels between RHs and users. Note that since the indoor environment was static and the users were stationary, the D-MIMO user selection algorithm always picked the same five groups consisting of four users per group. We considered forty random groupings of four users for comparison.

Figure 3.5 plots the distribution of per-user MCS index and group throughput

obtained from the experiments. In the interest of completeness, we also plot the distribution of results obtained from an oracle-based selection of users (similar to the results in Figure 2.8). That is, we assumed the presence of an all-knowing oracle that had perfect knowledge of the channels between RHs and all the users prior to user selection and hence could optimally choose users to serve in every TXOP. It can be seen from Figure 3.5a that the D-MIMO user selection algorithm chose an MCS index of 4 for 60% of the users whereas random selection chose the same MCS index for only around 13% of the users. Figure 3.5b describes similar trends in D-MIMO group throughput as well. The proposed user selection algorithm achieved an improvement of 60% in median and 43% in average group throughput compared to random user selection. It is also encouraging to observe that the proposed user selection algorithm attains a performance close to optimal selection—we observed a difference of only 13% in average group throughput between the optimal and proposed user selection methods. The results from experiments hence corroborate our simulation results compiled in Section 2.6.1 and demonstrate the effectiveness of the proposed user selection algorithm in selecting users to serve with MU-MIMO transmission that maximize D-MIMO group throughput performance without requesting CSI feedback from all the users associated with the D-MIMO group.

Chapter 4

Dynamic Resource Management in D-MIMO Wi-Fi Networks Using Deep Reinforcement Learning[§]

4.1 Summary and Organization

This chapter explores the potential of harnessing techniques and algorithms from deep reinforcement learning (DRL) to address two important dynamic resource management problems in D-MIMO Wi-Fi networks, which are described below, that are known to be NP-Hard:

1. **Channel assignment problem**: If there are N D-MIMO groups and K available channels ($K < N$), what is the best channel assignment policy to maximize user throughput performance?

2. **RH clustering problem**: How should RHs be clustered together to form D-MIMO groups in order to best serve users in the network? How should this grouping be updated in response to changing user distributions?

Since both the aforementioned problems are known to be NP-Hard and they get exacerbated when network conditions are dynamic, it is worthwhile to investigate if learning-based methods can address these problems wherein an agent learns about the dynamics of the network environment and acts accordingly. In other words, we propose to add intelligence to D-MIMO Wi-Fi networks to empower them with an autonomous adaptation to dynamic network scenarios. To the best of our knowledge, this is the first attempt at using DRL agents in the context of D-MIMO Wi-Fi networks. The main distinguishing feature of our work is that we consider training DRL agents in

[§]Parts of this chapter have been published as [17]

wireless network environments (i) that have large state and action spaces, and (ii) that are episodic in nature. We also consider practical network scenarios like non-uniform spatial distribution of users and user mobility. This work demonstrates that DRL agents with fairly simple implementations—in terms of number of hidden layers and the implemented learning algorithm—attain an improvement of up to 20% in user throughput performance in D-MIMO Wi-Fi networks compared to existing heuristic solutions.

This chapter is organized as follows. Section 4.2 reviews existing literature in the area of dynamic resource management problems in D-MIMO as well as applications of DRL in the context of wireless communications and networking. Section 4.3 provides some example scenarios to motivate the use of DRL in D-MIMO Wi-Fi networks. Section 4.4 gives an introduction to reinforcement learning and its associated terminology, a walk-through of an example learning episode, motivation for the use of DRL in our work, and the reasoning behind the choice of learning agents to address the problems of interest in this work. Section 4.5 describes the simulation and learning setup, and provides a detailed account of the results obtained from on-line training based on simulations of several network scenarios. Section 4.6 provides an account of the key takeaways from the results, and the lessons learned from implementing the DRL framework. Section 4.7 discusses the time spent on on-line training.

4.2 Literature Review

The problem of channel assignment of access points in Wi-Fi networks is known to be ***NP-Hard*** for which many heuristic-based solutions exist in the literature [45]. On the other hand, there exists limited literature on clustering of APs for distributed MIMO. Authors of [15, 46] grouped nearby antennas/APs into one cluster but the grouping was static. Another strategy to group antennas was on a per-packet basis [47–49] which needed full channel state information (CSI) from the users. Although the clustering was dynamic, the proposed solutions were difficult to implement in practice because of the time varying nature of CSI. References [50, 51] showed that the problem of determining the best AP clustering policy was ***NP-Complete*** (that is, both NP and NP-Hard) and the

authors in [51] performed clustering of APs based on balanced co-clustering of bipartite graphs and uplink traffic distribution of users. The focus of our work, however, is downlink service to users. Since both the aforementioned problems are known to be NP-Hard and they get exacerbated especially when the network environment is dynamic, it is worthwhile to investigate if learning-based methods can be used to address these problems wherein an agent learns about the dynamics of the network environment and acts accordingly.

Deep learning and reinforcement learning have emerged in recent years as technologies of valuable importance in a gamut of fields, be it image or pattern recognition, robotics, or even DNA synthesis. Deep reinforcement learning (DRL) is a synergistic combination of these two techniques in which deep neural networks are used in the training process to improve the learning speed and performance of reinforcement learning algorithms, especially in high-dimensional state and action spaces. DRL has been used to play a variety of games and the agents have been successful in outperforming human players and achieving superhuman scores [52]. They have proved their effectiveness in the area of wireless communications as well. Some examples of using DRL in communications include proactive data caching and computation overloading in mobile edge networks [53, 54], network security and preservation of wireless connectivity [55], traffic engineering and resource scheduling [56, 57], and enabling multiple access in wireless networks [58]. A comprehensive literature review of applications of DRL in the context of communications and networking is provided in [59].

4.3 Use of Reinforcement Learning in D-MIMO Wi-Fi Networks

The scenario described in Chapter 2 was a fairly simple deployment of a D-MIMO Wi-Fi network in which the users were static and distributed uniformly, and the network was not subject to any external Wi-Fi interference. However, practical wireless networks rarely exhibit such favorable characteristics. It is, hence, desirable to design D-MIMO Wi-Fi networks which address the dynamic resource management challenges previously

mentioned (known to be NP-Hard), particularly when network conditions are dynamic. With this high-level goal in mind, the following discussion describes a few example scenarios to motivate the use of DRL to improve the performance of D-MIMO Wi-Fi networks. The following discussion assumes an enterprise D-MIMO Wi-Fi network, managed by an administrator, as the network of interest.

Before studying these scenarios in detail, it is important to understand what is meant by the 'performance' of a D-MIMO Wi-Fi network. There are many different quantities of interest in a wireless network, the suitability of which depend greatly on the target applications. This chapter considers the throughput performance of users as the metric of interest; it could be average throughput of users or the throughput obtained by some percentile of users in the network. The specifics of the DRL framework as well as the different agents used will be explained in detail in Section 4.4 and Section 4.5 respectively; this section will keep the discussion to a generic DRL agent.

4.3.1 Vanilla Channel Assignment

First, consider a problem not specific just to D-MIMO networks but is common to most wireless networks: the problem of channel assignment. Considering Wi-Fi networks in general, the goal is to determine which Wi-Fi AP gets assigned which of the available channels in order to maximize user throughput performance. In the particular case of D-MIMO, the goal will be slightly modified as determining the best group-channel assignment policy. Consider a D-MIMO Wi-Fi network with sixteen groups and four non-overlapping channels available. If the network starts with the worst channel assignment strategy (see Figure 4.1 in which all groups are assigned the same red channel), a DRL agent should determine how to assign the available channels to the D-MIMO groups in order to attain the best user throughput performance.

The channel assignment problem is chosen as a starting point because it is not restricted only to D-MIMO networks. It is easy to see that the D-MIMO network in Figure 4.1 is, in fact, just a typical Wi-Fi network deployment consisting of 16 Wi-Fi APs, with each AP having spatially separated antennas (at the locations of the RHs). Hence, the discussion is applicable to simple Wi-Fi networks not implementing D-MIMO as

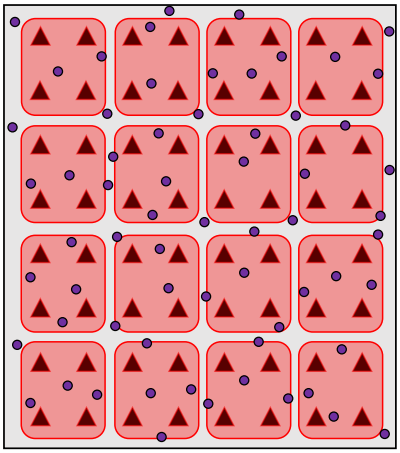


Figure 4.1: A D-MIMO Wi-Fi network with 16 groups (with four RHs each), all assigned to the same channel. Triangles represent RHs and circles represent users.

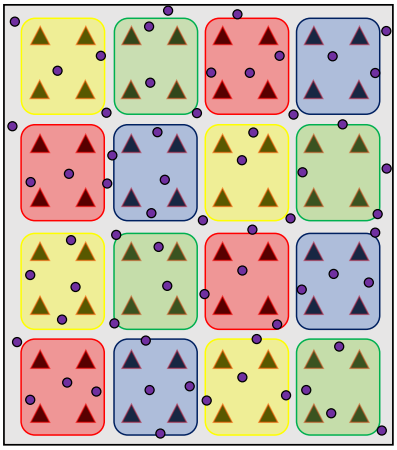


Figure 4.2: Channel assignment based on a simple heuristic. Each color represents a unique non-overlapping channel.

well.

Determination of the optimal channel assignment policy is known to be NP-Hard [60]. For the simple network described in Figure 4.1, channel assignment may be performed by simple heuristics like spacing out D-MIMO groups on the same channel far from each other in order to lower the channel contention among these groups as well as the interference from co-channel groups. The resulting channel assignment may look as shown in Figure 4.2. If the performance of the DRL agent converges to what the heuristic achieves, it suffices to say that the agent is effective in determining the best group-channel assignment policy. *It is important to study the effectiveness of DRL agents in basic problems like the vanilla channel assignment before using them to approach more complex network scenarios.*

4.3.2 Channel Assignment with External Wi-Fi Interference

The next scenario of interest is to use a DRL agent to make a D-MIMO Wi-Fi network resilient to random external Wi-Fi interference (see Figure 4.3). External Wi-Fi interferers may randomly appear in the orange zone in the vicinity of the considered D-MIMO network and each interferer may be assigned one of the four channels used by the D-MIMO groups. The goal of the DRL agent now is to update the channel

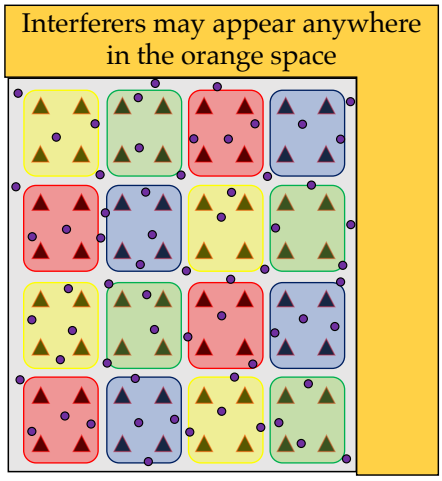


Figure 4.3: A D-MIMO Wi-Fi network with random external Wi-Fi interference in its vicinity. The interferers may operate in channels red, blue, yellow or green.

assignment policy such that groups close to external interferers do not use the same channel as the latter. This will trigger changes to channel assignments of other groups which need not be neighbors of the external interferers.

There exists extensive literature on channel assignment for APs in Wi-Fi networks based on heuristics, as described in [45]. Of these, HSUM is a popular heuristic algorithm based on weighted graph coloring, as described in [60]. Authors of the HSUM algorithm model channel assignment as a minimum-sum weighted vertex coloring problem in which different weights are put on interference edges. Looking at interference from the perspective of the users, this approach attempts to minimize the maximum interference as seen by clients in all common interfering regions. Specifically, in HSUM, Wi-Fi APs are required to transmit their interference metrics to their AP peers in order to facilitate a global view of the network topology at each AP. The maximum weighted interference is then minimized by HSUM in a global sense. Since the HSUM algorithm focuses mainly on Wi-Fi networks operating under the same administrative domain, we use HSUM to benchmark the performance of the DRL agent.

4.3.3 Meeting Multiple Objectives

The scenarios considered so far consisted of a single objective to be met, which was to maximize the throughput performance of users. This section, however, explores

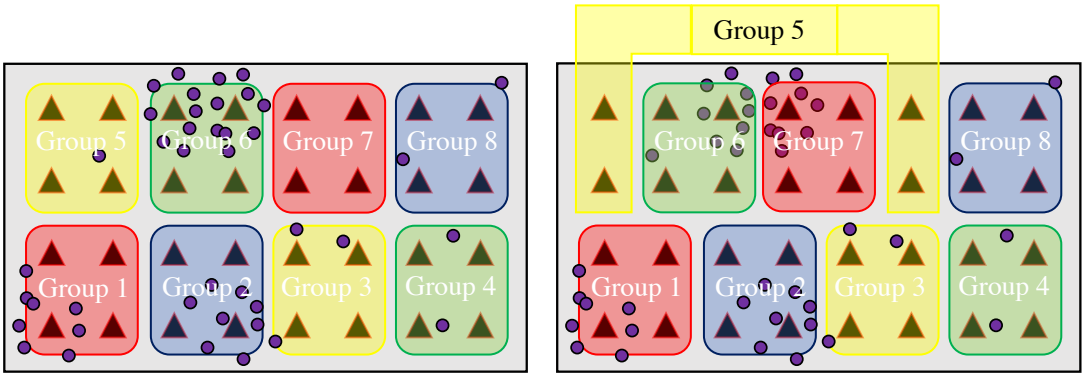
the use of a DRL agent to help a D-MIMO network meet multiple network objectives. Specifically, consider the case of channel assignment in the presence of random external Wi-Fi interference (see Figure 4.3) but now the agent has two goals to meet: maximize the average throughput of users ***and*** the fairness of throughput distribution among users. There is a separate branch in reinforcement learning literature studying this class of problems called multiple objective reinforcement learning (MORL). At a high level, MORL algorithms can be classified into different categories based on the following criteria:

- linear vs. non-linear scalarization of rewards from different objectives,
- meeting different objectives with a single policy vs. multiple policies, and
- training of the agent based on value iteration vs. policy iteration

Section 4.5.3 and the latter part of Section 4.4 discuss the ideas of scalarization and policy vs. value iteration based training respectively in more detail. This is so because these notions will be more accessible after familiarization with the basic framework of DRL, which the initial part of Section 4.4 provides. The objective of the DRL agent in this case is to update the channel assignments of D-MIMO groups in response to the external Wi-Fi interference such that both the average throughput of users as well as the fairness of throughput among the users is maximized. The yardstick for comparison of the performance of the DRL agent is again the HSUM-based channel assignment.

4.3.4 D-MIMO RH grouping

The following discussion examines a problem specific to D-MIMO networks. The D-MIMO Wi-Fi network considered in Figure 4.2 is such that four neighboring RHs form one group; this arrangement is referred to as *adjacent grouping*. It is not clear if this arrangement is indeed the best RH clustering policy, particularly when users are non-uniformly distributed in the network space. Consider the scenario described in Figure 4.4 in which a D-MIMO network consists of 32 RHs. The colors in Figure 4.4 identify the channel assigned to the groups. Several users have been distributed non-uniformly in the networks space such that there is a dense concentration of users



(a) Grouping of neighboring RHs; this arrangement is called adjacent grouping

(b) Updated grouping of RHs in response to the user distribution

Figure 4.4: A D-MIMO Wi-Fi network with 32 RHs (represented by triangles) and users (represented by circles) non-uniformly distributed in space. The arrangement in Figure 4.4b achieves an improvement of 20% in average user throughput compared to Figure 4.4a.

around a few RHs. A practical example of such a user distribution is the presence of a conference or a meeting room in an office space where users typically congregate. The D-MIMO network should cater to higher data demands in this room compared to the rest of the office space. With adjacent grouping of RHs (as shown in Figure 4.4a), the dense concentration of users will have to be served by RHs belonging to the same group (group 6 in Figure 4.4a). However, if the RH clustering is modified to be the arrangement as shown in Figure 4.4b, then the dense concentration of users gets split to be served by two groups (groups 6 and 7 in Figure 4.4b) which, in fact, improves the average user throughput performance by up to 20% (this result was obtained from simulations). It also helps that the rest of the office space is sparsely populated with users. Notice that RHs belonging to group 5 and group 3 are assigned channel yellow and they are close to each other. However, this does not become a complication since RHs in group 5 do not have any users to serve and hence do not contend with group 3 for access of channel yellow. Note that the arrangement in Figure 4.4b may not be the optimal grouping for that user distribution; this example is used to illustrate the importance of clustering of RHs in improving user throughput performance. The goal of the DRL agent, in this scenario, is to update the RH grouping arrangement in response to the distribution of users in the network space.

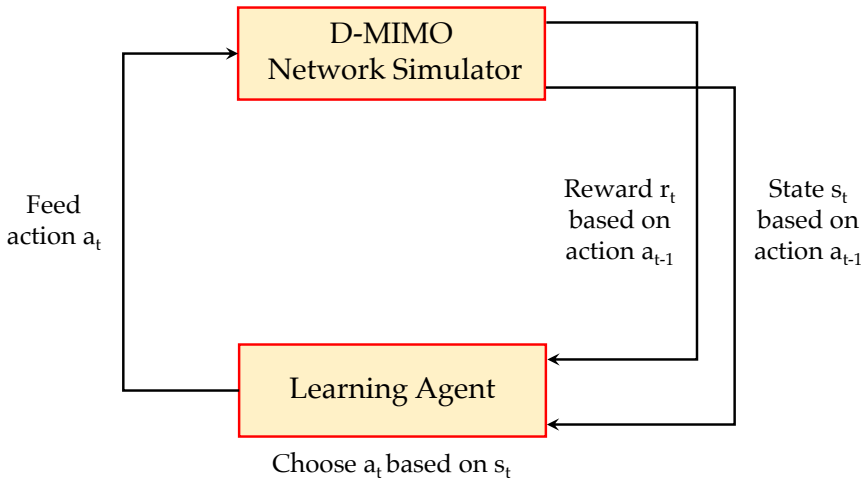


Figure 4.5: Reinforcement learning framework

4.4 Reinforcement Learning Framework

Figure 4.5 describes the high-level architecture of the reinforcement learning (RL) framework implemented in this chapter. The framework consists of the custom D-MIMO network simulator introduced in Chapter 2 at its core. It further consists of a learning agent that learns about the D-MIMO network environment by performing different actions (a) on it based on the current state (s) of the environment and the reward (r) that the agent receives for each action that it carries out; the agent receives a positive reward if the action it executed resulted in a better performance of the network compared to before, else it receives a negative reward.

This section first introduces some basic terminology associated with RL, provides a walk-through of an example learning episode, motivates the use of deep reinforcement learning (DRL) to address the problems described in Section 4.3, and ultimately provides a mathematical overview of the DRL algorithms implemented in this work. Let $\mathcal{A}$ denote the set of all possible actions that could be performed by the agent, and $\mathcal{S}$ denote the set of all states that the environment could be in. The subscript t in the following discussion denotes the time/step index at which the agent interacts with the environment.

- ENVIRONMENT: The D-MIMO Wi-Fi network of interest serves as the environment on which the learning agent acts. The environment can be modeled as

a markov decision process (MDP) with a state transition matrix, of dimension $|\mathcal{A}| \times |\mathcal{S}|$, which provides the probability $p(s_{t+1}|s_t, a_t)$ of moving from state s_t to s_{t+1} when action a_t is performed. An MDP is also characterized by the reward model that describes the real-valued reward value that the agent receives for choosing action a in state s. The environment feeds the state information and the obtained reward for an action to the agent.

- EPISODE: An episode is a sequence of actions carried out by the agent, and the corresponding states and rewards obtained from the environment. An episode terminates with either a terminal state or when a certain number of actions have been carried out. Typically, games are episodic wherein an episode completes when a player has won or lost the game. In the scenarios considered in this work, however, *one learning episode is terminated when the number of carried out actions exceeds a threshold* T, that is, an episode consists of a fixed number of actions. This is a deliberate choice because the objective of the agent is to maximize/improve the throughput performance of the D-MIMO network and hence there exists no clear winning or losing condition. One may advocate for a threshold based episode termination in which an episode is declared to be won if throughput (or any performance metric of choice) exceeds a predetermined threshold. That may work, but the DRL agents implemented in this work are free to train without any hard constraints imposed on them.

- POLICY: The actions of an agent are governed by a map called the policy, denoted by π. It decides the probability of the agent choosing an action $a_t = a$ when the environment is in state $s_t = s$. A policy is usually parameterized by parameters θ and is defined as

$$\begin{aligned} \pi &: \mathcal{S} \times \mathcal{A} \rightarrow [0, 1], \\ \pi_\theta(a|s) &= \mathbb{P}\left[a_t = a \;\middle|\; s_t = s \right]. \end{aligned} \tag{4.1}$$

  Note that policy π_θ is stochastic, that is, $\pi_\theta(a|s)$ is modeled as a probability distribution over the set of all actions $\mathcal{A}$ given the current state (s) of the environment.

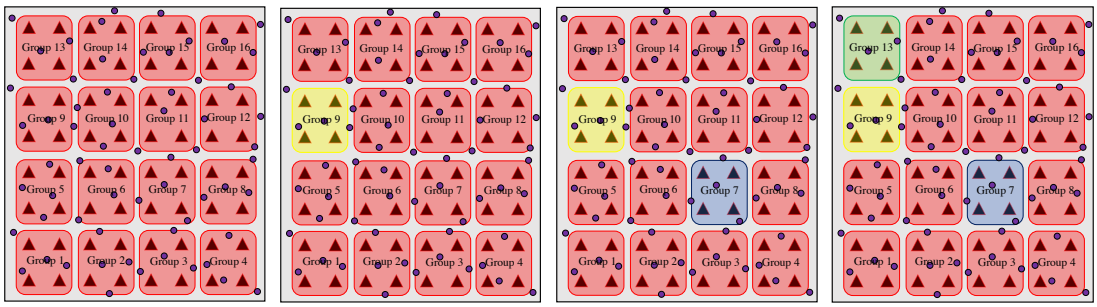
- STATE-VALUE FUNCTION: Value function quantifies how good it is to be in a given state 's'. It is defined as the expected return starting at state s_t following a policy π. The value function is formulated as

$$V^{\pi}(s) = \mathbb{E}[R] = \mathbb{E}\left[\sum_{k=0}^{T-t} \gamma^k r_{t+k+1} \;\middle|\; s_t = s\right], \tag{4.2}$$

  where R denotes the cumulative discounted return, T represents the number of steps/actions in a learning episode, and γ denotes the discount factor. Note that the reward obtained for choosing action $a_t = a$ when at state $s_t = s$ is represented by r_t.

- DISCOUNT FACTOR: Denoted by γ, discount factor was originally conceived as a mathematical trick, in case of non-episodic environments where the number of actions $|\mathcal{A}| \rightarrow \infty$, to make the infinite sum in (4.2) (when $T \rightarrow \infty$) finite. The value of the discount factor is bounded between [0,1] and it determines the importance of the future rewards. γ closer to 0 makes the agent opportunistic by considering rewards in the immediate future whereas γ close to 1 prioritizes rewards in the distant future.

It is worthwhile to digress a bit to clarify the idea of actions, states and rewards in the context of D-MIMO Wi-Fi networks. Consider the timeline of an example learning episode as described in Figure 4.6 in case of the vanilla channel assignment problem discussed in Section 4.3.1. The D-MIMO network of interest consists of sixteen groups with four RHs each. Assume the availability of four non-overlapping available channels. An episode begins with the D-MIMO network in its initial state s_1. **State**, at any step, is the current group-channel assignments. For the network shown in Figure 4.6, state at step t (denoted by s_t) is a 16×1 vector $[c_1, c_2, c_3, ..., c_{16}]^T$ where c_i represents the channel assigned to group i and $[.]^T$ denotes the transpose operator. Let the throughput metric of interest (which needs to be maximized) in the initial state be x_1. At step t_1, the agent chooses **action** a_1, based on the initial state s_1, which may be to assign channel yellow to group 9. The D-MIMO network simulator receives this action, performs the necessary channel assignment update (thereby generating the new network state s_2),



(a) Step t_1
Initial state: s_1
Choose action a_1 as assigning channel yellow to group 9 based on state s_1

(b) Step t_2
Obtained reward: r_2
Updated state (based on action a_1): s_2
Choose action a_2 as assigning channel blue to group 7 based on state s_2

(c) Step t_3
Obtained reward: r_3
Updated state (based on action a_2): s_3
Choose action a_3 as assigning channel green to group 13 based on state s_3

(d) Step t_4
Obtained reward: r_4
Updated state (based on action a_3): s_4
Choose action a_4

Figure 4.6: Timeline of a typical learning episode (vanilla channel assignment).

and simulates the network to obtain throughput metric x_2. The simulator computes the difference between the throughput metrics x_2 and x_1 as the **reward** r_2 for action a_1. The simulator feeds the updated network state s_2 and reward r_2 to the agent. Based on the new state s_2, the agent chooses action a_2 at step t_2, which may be assigning channel blue to group 7. The environment receives this action, performs the channel assignment update (thereby generating network state s_3), simulates the network to generate the throughput metric x_3, computes the reward r_3 as $x_3 - x_2$, and feeds both s_3 and r_3 to the learning agent. This cycle continues until the number of actions chosen exceeds the threshold T; this marks the end of one episode. Once an episode terminates, the environment resets back to its initial state s_1, that is, all groups are assigned channel red, and a new episode commences. After the completion of an episode, the agent computes the cumulative discounted returns (as shown in (4.2)) at each step t that are then used to update the weights of its neural network depending on the training algorithm of choice.

Note that state, action, and reward information change when considering problems other than the vanilla channel assignment problem. The aforementioned discussion is intended to be an example to better understand these concepts and see how an episode progresses with time. Specific details regarding the choice of state, action, and reward

will be discussed individually for each problem in Section 4.5.

4.4.1 Motivation for the use of Deep Reinforcement Learning

The D-MIMO Wi-Fi network environments considered in this work have large state ($\mathcal{S}$) and action ($\mathcal{A}$) spaces. For instance, in case of the aforementioned vanilla channel assignment problem (described in Section 4.3.1 as well), the number of possible actions that the agent can potentially perform in each step is 64, since there are 16 groups and 4 available channels, and each learning episode consists of T actions. Furthermore, since the state vector fed to the agent consists of the channel assignments of the 16 groups, the cardinality of the state space $|\mathcal{S}| = 4^{16}$. In such settings, typical reinforcement learning algorithms incur a high computational as well as storage overhead. For instance, Q-learning [61], a popular RL algorithm, entails maintaining a table of Q-values (a variant of the value function in (4.2)) of dimension $|\mathcal{S}| \times |\mathcal{A}|$. Therefore, the algorithm incurs a prohibitively high overhead in computing the Q-values over the entire action space as well as a high storage overhead in maintaining the Q-table. This feature is popularly known as the 'curse of dimensionality'. To circumvent this issue, we employ *deep reinforcement learning* (DRL) algorithms in our work. Such algorithms are symbiotic combinations of deep learning techniques and RL algorithms in which deep neural networks are used in the training process to improve the learning speed and performance of RL algorithms, especially in case of environments with large state and action spaces. Accordingly, the learning agent in Figure 4.5 is modeled as a deep neural network that approximates the non-linear relationship between its input 'state' and the output 'action' in each step of an episode. The configuration of the neural network as well as its working as a learning agent are specific to the problems of interest; both of these aspects are clarified in detail in Section 4.5.1 and Section 4.5.4 respectively.

Coming back to reinforcement learning terminology, the objective of an agent is to determine the best policy π which will maximize the value function given in (4.2). The optimal policy π^* can be formulated as follows:

$$\pi^* = \arg\max_{\pi} V^{\pi}(s), \forall s \in \mathcal{S} \tag{4.3}$$

The optimal policy π^* is optimal for all states s, that is, the same policy can be used regardless of what the initial state of the environment is. Broadly, reinforcement learning algorithms are classified into two categories based on how they arrive at the optimal policy π^*: (i) *policy-iteration based* in which policies are directly searched, evaluated and improved, and (ii) *value-iteration based* in which the optimal value function is first determined from which the optimal policy is extracted. Policy-iteration based methods converge quicker compared to value-iteration and have been more commonly used when the action space of the environment is large, for example, continuous action space in robotics. Policy-iteration based methods, specifically policy gradients, were used by DeepMind for playing the game AlphaGo which has a large discrete action space [62, 63]. This work will also use policy gradients to train the learning agents since the action space in case of D-MIMO Wi-Fi networks is large (as described previously).

In the following section, we first provide a brief mathematical overview of policy gradients and later focus on the two algorithms implemented in this work.

4.4.2 Policy Gradients

As discussed before, policy π is modeled as a parameterized function with respect to θ. Policy gradient methods aim at modeling and optimizing the policy directly. Let the reward function be redefined as follows:

$$J(\theta) = \sum_{s \in \mathcal{S}} d^{\pi}(s) V^{\pi}(s) = \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\theta}(a|s) Q^{\pi}(s, a), \tag{4.4}$$

where $d^{\pi}(s)$ represents the stationary distribution of Markov chain for π_{θ}, and $Q^{\pi}(s, a)$ denotes the state-action pair value function that is similar to the value function described in (4.2) and is defined as

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{T-t} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right], \tag{4.5}$$

i.e, it describes the value of a state-action pair when the agent follows the policy π. Since the objective of the DRL agent is to maximize its reward function, the parameters θ of the policy need to be moved in the direction of the gradient of the reward function

to find the best θ that produces the highest return, that is,

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{4.6}$$

where α denotes the learning rate of the agent.

At first glance, computing the gradient of the reward function in (4.4) might seem difficult because the reward function depends both the action selection as well as the stationary distribution of states, both of which depend on θ directly or indirectly. However, the policy gradient theorem (described in [64,65]) provides a formulation for the gradient which does not involve the derivative of the state distribution $d^\pi(s)$ and the gradient can therefore be computed as

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} Q^\pi(s,a) \pi_\theta(a|s) \\
&\propto \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} Q^\pi(s,a) \nabla_\theta \pi_\theta(a|s) \\
&= \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^\pi(s,a) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \\
&= \mathbb{E}_\pi[Q^\pi(s,a) \nabla_\theta \ln(\pi_\theta(a|s))],
\end{aligned} \tag{4.7}$$

where $\mathbb{E}_\pi$ is a simplified notation for $\mathbb{E}_{s \sim d^\pi, a \sim \pi_\theta}$, that is, both state and action distributions follow the policy π_θ. Such algorithms are called *on-policy* algorithms in which training samples are collected according to the policy that the agent is optimizing for. Algorithms which use a different policy to sample training data are called *off-policy* algorithms.

There exists extensive literature on different kinds of policy gradient algorithms. The following discussion will focus on two such algorithms used in this work: REINFORCE agent [66] and deep deterministic policy gradients [67].

REINFORCE Agent

REINFORCE or the Monte-Carlo policy gradient [66] relies on an estimated return by Monte-Carlo methods using episode samples to update the policy parameters θ. The REINFORCE agent makes use of the fact that the expectation of the sample gradient

is equal to the actual gradient. The parameter update is described in the following procedure:

1. Randomly initialize the parameters θ of the policy π_θ.

2. Obtain an episode of length T (i.e, T number of actions) that consists of a sequence of state-action-reward-state-action (SARSA). The full sequence of an episode is called a trajectory.
   Trajectory $\tau = \{s_1, a_1, r_2, s_2, a_2, r_3, ..., s_T, a_T, r_{T+1}\}$

3. For $t = 1, 2, ... , T$
   - Compute the cumulative discounted return $G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k+1}$
   - Update the parameters of the policy as $\theta \leftarrow \theta + \alpha\gamma^t G_t \nabla_\theta \ln \pi_\theta(a_t|s_t)$

This agent is *on-policy* as it computes the cumulative discounted returns from sample episodes collected according to the policy π_θ and use them to update parameters θ of the same policy. This algorithm requires the full trajectory of an episode and hence is called a Monte-Carlo method.

Deep Deterministic Policy Gradients (DDPG)

This is an *off-policy deterministic actor-critic* algorithm. The following discussion discusses each of these terms individually.

Off-policy algorithms were introduced in the introductory part of this section. There are several advantages to using *off-policy* algorithms compared to on-policy, some of which are listed below:

- Off-policy algorithms do not require full episodes and can reuse any past experiences for better sample efficiency. This is called 'experience replay' through which the agent randomly samples from past stored state-action-reward-next state experiences; these experiences need not be part of the same episode.

- The agent uses a different policy for sample collection than the target policy, which leads to better exploration.

Throughout the discussion in this section, the policy $\pi_\theta(a|s)$ had been modeled as a probability distribution over the set of all actions $\mathcal{A}$ given the current state of the environment and hence it is stochastic. In *deterministic policy gradients*, the policy is modeled as deterministic, that is, an action is a deterministic function of the current state ($a = \mu_\theta(s)$). The policy μ is again parameterized by parameters θ.

Use of deterministic policies necessitates a reformulation of the reward function. Since this algorithm is off-policy, let the training trajectories (sequence of state-action-reward-next state) be collected according to a stochastic policy $\beta(a|s)$. Let $\rho_0(s)$ denote the initial state distribution, $\rho^\beta(s \rightarrow s', k)$ denote the visitation probability density at state s' after moving k steps from state s following the policy β, and $\rho^\beta(s') = \int_{\mathcal{S}} \sum_{k=1}^{\infty} \gamma^{k-1} \rho_0(s) \rho^\beta(s \rightarrow s', k) ds$ represent the discounted state distribution. Then, the reward function (similar to the reward function in (4.4)) is redefined as

$$J(\theta) = \int_{\mathcal{S}} \rho^\beta(s) Q^\mu(s, \mu_\theta(s)) ds. \tag{4.8}$$

Computing the gradient of the reward function in (4.8) with respect to parameters θ using chain rule yields

$$\begin{aligned} \nabla_\theta J(\theta) &= \int_{\mathcal{S}} \rho^\beta(s) Q^\mu(s, a) \nabla_\theta \mu_\theta(s)|_{a=\mu_\theta(s)} ds \\ &= \mathbb{E}_{s\sim\rho^\beta} \left[\nabla_a Q^\mu(s, a) \nabla_\theta \mu_\theta(s)|_{a=\mu_\theta(s)} \right]. \end{aligned} \tag{4.9}$$

The computation of the gradient in (4.9) requires taking an expectation over the state space only. Comparing this with the gradient in (4.7), it is evident that the computation of the same gradient requires performing an expectation over both state and action spaces for the stochastic policy case, thus necessitating collecting more samples. Deterministic policy gradients are hence helpful when action spaces are vast.

DDPG algorithm belongs to the class of *Actor-Critic methods* that attempt to learn the value function assisting the policy update in addition to the policy itself. The *critic* updates the parameters of the value function (it could be either the state value function $V^\pi(s)$ or the state-action value function $Q^\pi(s, a)$) and the *actor* updates the parameters θ of the policy π_θ in the direction suggested by the critic.

This chapter uses an extension of the DDPG algorithm to be applied in large discrete action spaces. Specifically, this work uses the Wolpertinger agent [68] to reduce the

size of the action space, particularly for the problem of RH grouping (described in Section 4.3.4). This agent avoids the heavy cost of evaluating all actions while retaining generalization over actions. The Wolpertinger architecture consists of three main parts: *an actor network, K-Nearest Neighbors (K-NN), and a critic network*. The actor network reasons over actions within a continuous space and maps this output to a discrete action. The critic network is used to correct the decision made by the actor network. The DDPG algorithm is applied to update both the critic and actor networks. K-NN helps to explore a set of actions to avoid poor decisions. The reader is referred to [68] for details regarding the implementation of the Wolpertinger agent.

4.5 Results from On-line Training

This section revisits the problems described in Section 4.3, discusses the specifics of the DRL agents used to address these problems, and studies the results obtained from extensive training of the DRL agents. The DRL framework, shown in Figure 4.5, was implemented using a combination of **OpenAI Gym** [69] for the environment and **TensorFlow** [70] for the deep learning agent. OpenAI Gym was used as a wrapper outside the custom D-MIMO Wi-Fi simulator introduced in Chapter 2. For the problems described in Sections 4.3.1, 4.3.2, and 4.3.3, the D-MIMO network of interest was an office floor of dimension 80 m $\times$ 80 m with 16 D-MIMO groups (with four RHs per group) and 64 users uniformly distributed throughout the office space (see Figure 4.1). The RHs were separated (in x and y directions) by 10 m. There were four non-overlapping channels, each of bandwidth 80 MHz, assumed to be available in the 5 GHz band. Other simulation parameters and the channel model used in the simulations can be found in Chapter 2. The scenarios considered in this work were exclusively downlink with full buffer traffic to all users. Note that learning was episodic and the number of actions per episode was arbitrarily chosen to be 50. Each step/action in a DRL episode involved running a simulation of the D-MIMO network for a network time of 100 ms. In all figures in the following discussion, light lines plot the per-episode numbers whereas bold lines plot the moving averages over windows of 50 episodes.

Table 4.1: Specifics of the learning agent used in Sections 4.5.1, 4.5.2, and 4.5.3

Number of hidden layers	1
Number of input nodes	16
Number of hidden nodes	48
Number of output nodes	64
Configuration of layers	Densely connected
Hidden node activation	Rectified linear unit (ReLU)
Output node activation	Softmax
Learning rate of the agent	3×10^{-3}
Optimizer	Adam [71]
Loss function	Softmax cross-entropy

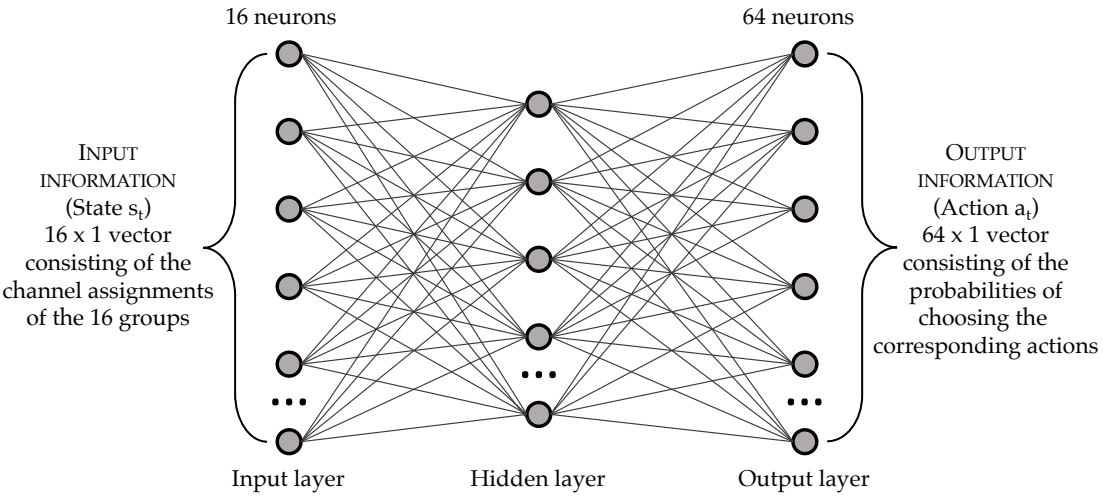


Figure 4.7: Description of the inputs to and outputs from the learning agent—modeled as a neural network—for the problems described in Sections 4.5.1, 4.5.2, and 4.5.3.

4.5.1 Vanilla Channel Assignment

State s_t: Group-channel assignments at step t; a 16×1 vector with element at index i indicating the channel assigned to group i

Initial state of each episode s_1: Channel red assigned to all 16 groups

Action a_t: Change the channel assignment of one group

Performance metric x_t: Throughput of the thirtieth percentile of all users

Reward $r_t = x_t - x_{t-1}$

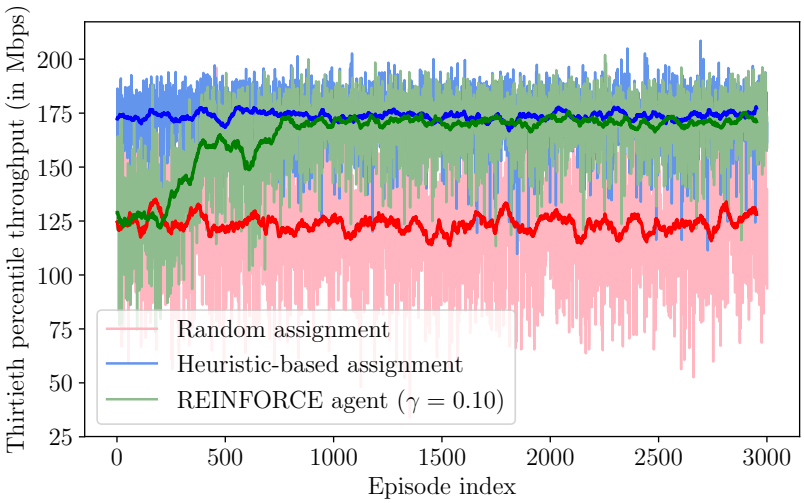
Agent: On-policy REINFORCE (details of the implementation provided in Table 4.1)

First, consider the case of vanilla channel assignment in D-MIMO Wi-Fi (described in Section 4.3.1). The learning agent used policy gradients REINFORCE (Section 4.4.2)

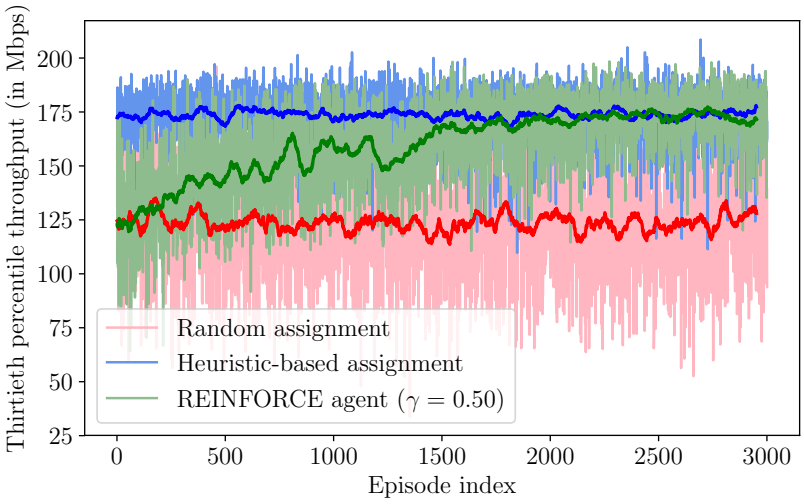
for training.

OPERATION OF THE LEARNING AGENT: Figure 4.7 describes the learning agent used in vanilla channel assignment (as well as in the problems considered in Sections 4.3.2 and 4.3.3). The learning agent was modeled as a three-layer neural network. As can be seen from Figure 4.5, the objective of the agent was to choose action a_t based on the network state s_t at step t. The input layer of the neural network consisted of 16 neurons that received the channel assignments of the 16 groups in step t as their input. The activation of the neurons in the hidden layer was chosen to be *rectified linear units (ReLU)* [72] since they have better training characteristics compared to sigmoid or hyperbolic tangent (tanh) activations; the latter two suffer from the problems of vanishing gradients and are hence slower to train. Furthermore, ReLU activation incurs lower computational complexity since it does not involve expensive exponential operations unlike sigmoid and tanh. The output layer consisted of neurons that corresponded to the different actions that might be performed in each step. In case of the channel assignment problems, the output layer consisted of 64 neurons corresponding to the 64 potential actions, and the output of each neuron represented the probability of choosing the corresponding action. To enable this, the activation of the output layer of the neural network was chosen to be *softmax*. The learning agent then chose the action with the highest probability to be performed in step t. In effect, the training of the agent could be perceived as a *multi-class classification problem* wherein the agent classified the action to be performed in the next step into the 64 possible actions. The state, action, and reward obtained in each step of an episode were stored and once the episode was completed, the parameters of the neural network were updated as described by the algorithm provided in Section 4.4.2.

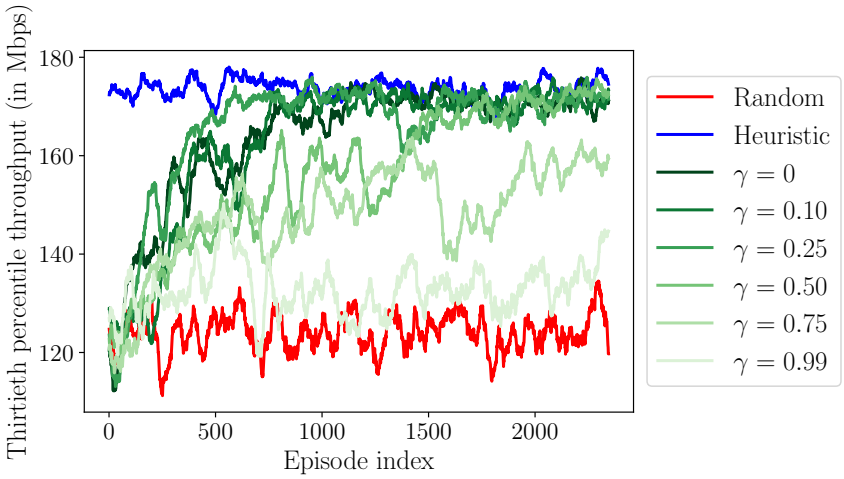
To compare the results obtained from using the REINFORCE agent, the following channel assignment strategies were also considered : (i) *random assignment* in which D-MIMO groups were assigned channels randomly in every episode, and (ii) *heuristic assignment* in which groups were assigned channels as shown in Figure 4.2. Each episode began with the initial state s_1 (the worst-case channel assignment), and an independent uniform distribution of users in the network space. Figure 4.8a shows



(a) DRL Agent with $\gamma = 0.10$



(b) DRL Agent with $\gamma = 0.50$



(c) Comparison of REINFORCE agents with different discount factors (moving averages of throughput over intervals of 50 episodes)

Figure 4.8: Throughput of the thirtieth percentile of users with different channel assignment schemes (results pertaining to Section 4.5.1)

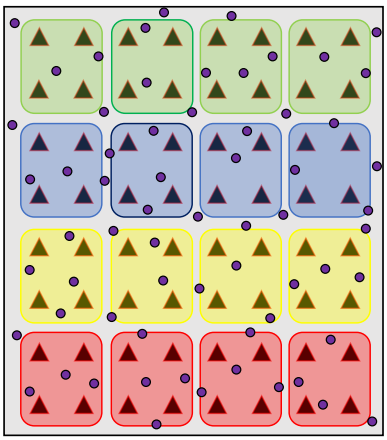


Figure 4.9: D-MIMO Wi-Fi network with a channel assignment that is different from the worst-case state

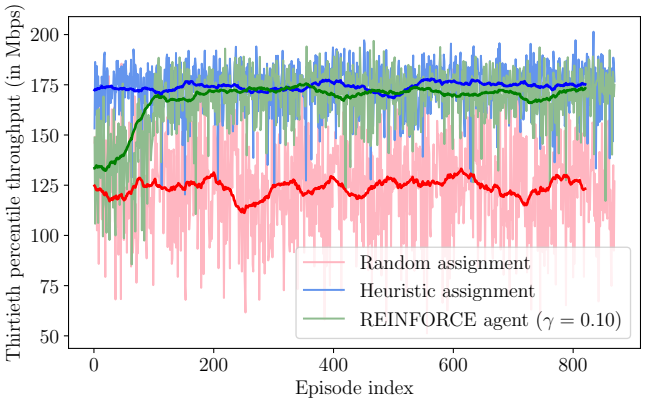


Figure 4.10: Throughput of thirtieth percentile of users when the network environment in each episode started from the state as shown in Figure 4.9

the throughput of the thirtieth percentile of all users observed over several episodes when using a DRL agent with discount factor $\gamma = 0.10$. Notice that the DRL agent was able to attain a throughput value similar to when using the heuristic-based channel assignment within a few hundred episodes, even though each episode began with the worst case channel assignment strategy.

We began each learning episode from the worst-case channel assignment *to demonstrate the efficacy of the learning agent in converging to the optimal channel assignment policy even when starting from an apparently undesirable state*. We also performed training such that the initial state of each episode was some other channel assignment policy (say, red channel assigned to groups 1–4, yellow channel assigned to groups 5–8, blue channel assigned to groups 9–12, and green channel assigned to groups 13–16 as shown in Figure 4.9). In that setting, the agent required around 200 episodes to converge to the performance of heuristic-based channel assignment (see Figure 4.10) compared to around 1000 episodes when each episode started from the worst-case state (see Figure 4.8a).

IMPACT OF DISCOUNT FACTOR: Discount factor (γ) of the learning agent determines how it values rewards; γ close to zero prioritizes immediate rewards while γ close to one values rewards in the future. The reinforcement learning framework built in

this work is such that the learning agent executed its training episodically and each episode consisted of a finite number of actions (T). Furthermore, *the learning agent received a reward for every action that it performed*. That is, in the environments of our interest, the agent received T rewards ($r_1, r_2, ..., r_T$) in each episode. This is different from typical DRL environments like games wherein the agent would receive a big reward at the end of a game depending on whether the game was won or lost. Therefore, the rewards that the agent received through the course of a learning episode are all equally important and not just the final reward. In this setting, immediate rewards or payoffs are important and hence the discount factor of the learning agent should be chosen closer to zero.

Figure 4.8b describes the learning performance of the agent when used with $\gamma = 0.50$ and it can be observed that the agent was yet again successful in determining the best channel assignment for D-MIMO groups. The difference in discount factors determined the aggressiveness with which the agent reached the best channel assignment; the agent with $\gamma = 0.50$ took longer to reach the best assignment compared to the agent with $\gamma = 0.10$. In fact, we compared the performance of the learning agent with different discount factors between zero and one. The results of this comparison are presented in Figure 4.8c, which shows that agents with lower discount factors $\gamma \leq 0.25$ achieved a better convergence behavior compared to agents with higher discount factors $\gamma \geq 0.50$. In fact, when $\gamma = 0.99$, the performance of the agent was similar that of random channel assignment, which indicates that the agent did not actually learn to optimally assign channels to the groups.

4.5.2 Channel Assignment with External Wi-Fi Interference

Next, we considered channel assignment in a D-MIMO Wi-Fi network but in the presence of external Wi-Fi interference in its vicinity (described in Section 4.3.2). The external interferers could be located within 15 m (in x and y directions) of the D-MIMO network (as shown in Figure 4.3). Note that ***the location of the interferers as well as the channels assigned to the interferers were different in different episodes*** (channels assigned to different interferers in the same episode could be different as well). This

may be interpreted as the network environment changing at the end of each learning episode.

The definitions of state and action for this scenario were the same as in problem 4.5.1 and the performance metric in this case was the throughput of the tenth percentile of all users. Each episode began with the channel assignment determined by the DRL agent in scenario 4.5.1 (that is, when there was no external interference present), denoted by s^*. To compare the performance obtained using the DRL agent, two different channel assignment strategies were implemented: (i) *sensing based assignment* in which D-MIMO groups assigned channels to themselves (in a distributed manner) based on the energy sensed in each channel (note that all groups assign channels to themselves synchronously), and (ii) *HSUM based assignment* [60] (after inducing necessary changes to the algorithm work in a distributed MIMO setting).

The DRL learning agent used the REINFORCE algorithm with a discount factor $\gamma = 0.25$. Figures 4.11a and 4.11b describe the results when one and three random external interferers were present respectively. It can be observed that the DRL agent was able to update the channel assignments of the D-MIMO groups in response to the presence of external interferers. The DRL agent was able to achieve similar throughput numbers (in fact, better for several episodes) compared to when HSUM was used. These results demonstrate the success of the DRL agent in identifying the fact that groups near interferers should be assigned channels different from the interferers. Observe that even though the locations and channels of the interferers were changed after each episode, the agent was still able to converge to the best policy within a few hundred episodes.

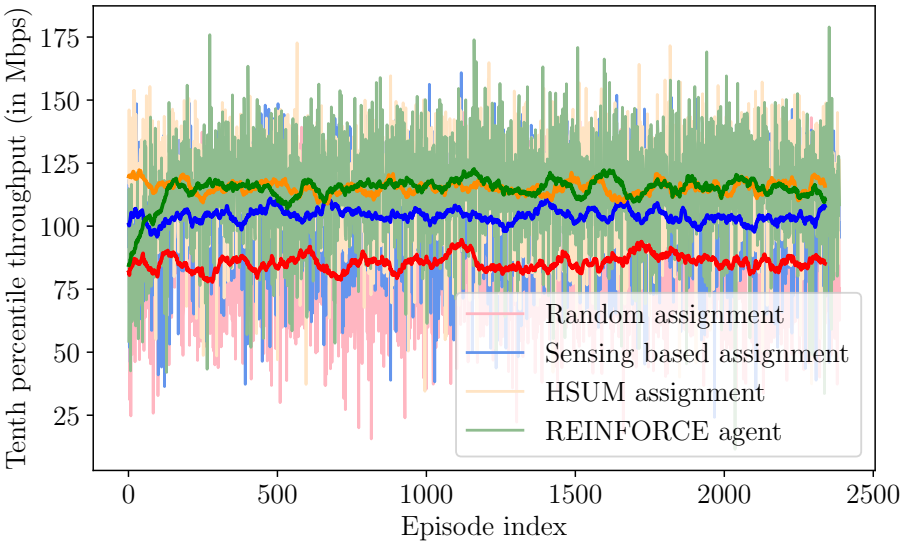
4.5.3 Meeting Multiple Objectives

State s_t and **Action** a_t: Same as problems in Sections 4.5.1 and 4.5.2

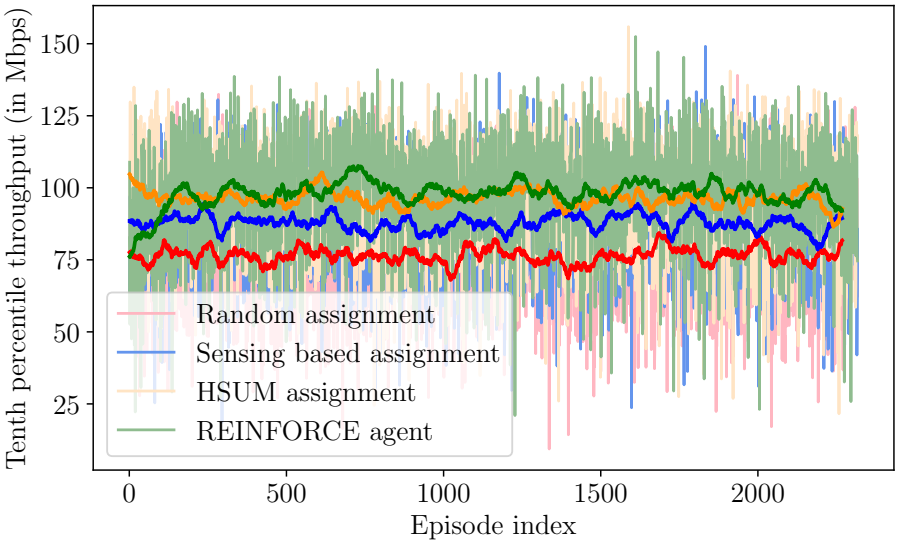
Initial state of each episode s_1: State s^* as described in Section 4.5.2

Performance metric x_t = Average throughput of users $\times$ Jain's fairness index of throughput

Reward $r_t = x_t - x_{t-1}$



(a) Results with one random external interferer



(b) Results with three random external interferers

Figure 4.11: Throughput of the tenth percentile of users obtained using different channel assignment schemes in the presence of random external Wi-Fi interference (results pertaining to Section 4.5.2)

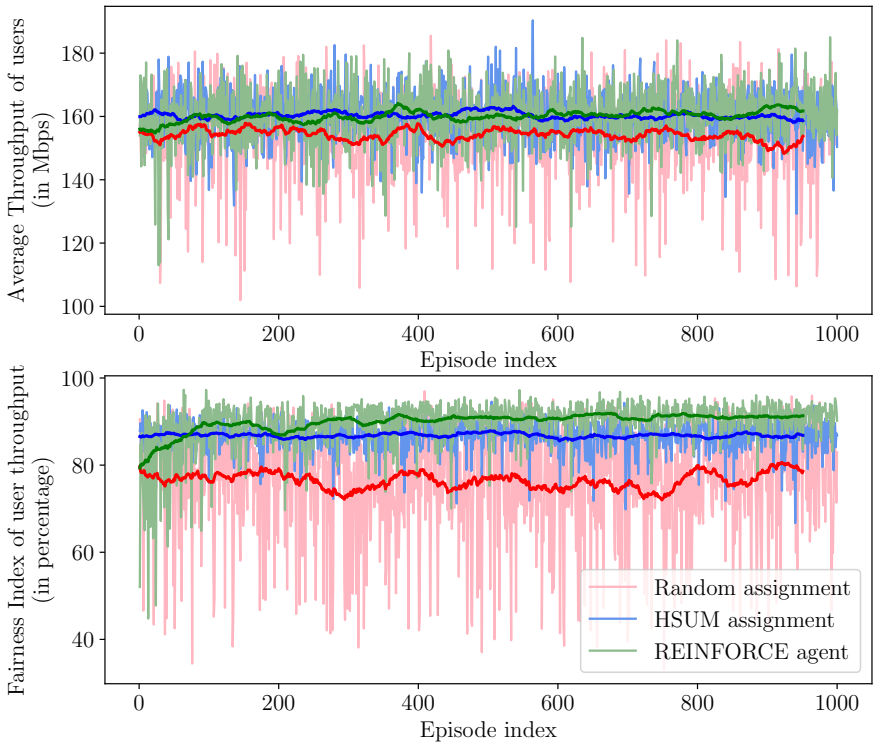
Agent: On-policy REINFORCE agent; $\gamma = 0.25$

Consider the scenario of D-MIMO with external Wi-Fi interference but with two objectives—maximize the average throughput of users ***and*** the fairness index of throughput among users. Fairness, in this context, is the Jain's fairness index [39], which is defined as:

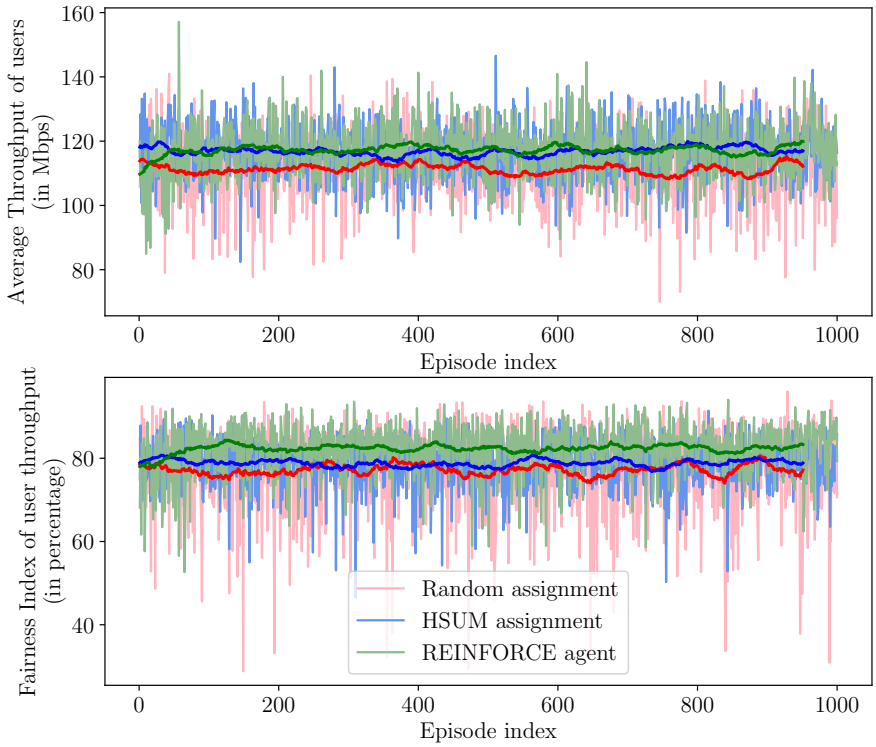
$$\text{Fairness} = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i^2}, \tag{4.10}$$

where $x_1, x_2, ..., x_n$ represent the throughput numbers of users $1, 2, \ldots, n$. The objective of the DRL agent was to find a single policy that would meet both these objectives. The performance metric, in this case, was defined as the product of the average throughput of users and the fairness index. The mapping of the two performance measures to a single quantity is called *scalarization*. If there exist k objectives $O^1, O^2, ..., O^k$ with corresponding rewards $r^1, r^2, ..., r^k$, then scalarization condenses these k rewards into a single metric $R = f(r^1, r^2, ..., r^k)$. There exist several methods for scalarization in literature: linear scalarization (where the function f is linear; usually a weighted sum of rewards), Chebyshev scalarization [73] to name a few. However, in the current scenario of maximizing average user throughput and fairness index, these methods did not yield a good policy. This behavior may be ascribed to the disparity in the scale of the rewards; fairness index was constrained to the range of [0,1] while the throughput performance metric was not.

Figure 4.12 compares the performance of the DRL agent with HSUM-based channel assignment for two cases: three external interferers (Figure 4.12a) and eleven external interferers (Figure 4.12b). Note that, similar to the scenario considered in Section 4.5.2, ***the location of interferers as well as the channels assigned to the interferers were changed after each episode.*** It can be observed that the DRL agent was able to achieve similar throughput performance as the HSUM-based assignment. However, the gains of using DRL were more evident in the trends of fairness index. The DRL agent was able to achieve higher throughput fairness among users while achieving a similar throughput performance as HSUM. Notice that the gains in fairness index are lower in Figure 4.12b compared to Figure 4.12a. This is understandable since there were more external Wi-Fi interferers in the vicinity of the D-MIMO network (in case of Figure 4.12b) and hence



(a) Results with three random external interferers



(b) Results with eleven random external interferers

Figure 4.12: Average throughput of users and Jain's fairness index of throughput among users in the presence of random external Wi-Fi interference (results pertaining to Section 4.5.3)

Table 4.2: Specifics of the learning agent used in the Wolpertinger architecture in Section 4.5.4

Number of hidden layers	2
Number of input nodes	82
Number of output nodes	1 (Continuous proto action)
Actor Network Configuration	
Number of hidden nodes	256 (first layer), 128 (second layer)
Learning rate (α_a)	3×10^{-3}
Target actor update parameter (τ_a)	1×10^{-3}
Critic Network Configuration	
Number of hidden nodes	64 (first layer), 32 (second layer)
Learning rate (α_c)	7×10^{-3}
Target critic update parameter (τ_c)	1×10^{-3}
Hidden node activation	Softplus
Output node activation	Hyperbolic tangent (tanh)
Optimizer	Adam [71]
Replay buffer size	10,000 samples
Mini-batch size	100

the DRL agent had limited scope to update the channel assignments to improve the performance of the network. Even then, it was able to perform better than HSUM in terms of the achieved fairness index. The takeaway message from this section is that the DRL agent was successful in (i) imbuing the D-MIMO network with resilience to dynamic external interference, and (ii) achieving a superior performance in simultaneously meeting both the objectives compared to HSUM.

4.5.4 D-MIMO RH Grouping

State s_t: RH-group assignments (a 32×1 vector with element at index i indicating the group to which RH i belongs) concatenated with the user-RH assignments (a 50×1 vector with element at index i representing the RH to which user i is associated) at step t

Initial clustering of RHs in each episode: Adjacent grouping strategy (see Figure 4.4a)

Action a_t: Swap/exchange RH a in group b with RH c in group d

Performance metric: x_t = Average throughput of users

Reward: $r_t = x_t - x_{t-1}$

Agent: Wolpertinger agent implementing DDPG for training (details in Table 4.2)

Consider the D-MIMO network shown in Figure 4.4 with 32 RHs and eight D-MIMO groups (with four RHs per group), with the RHs clustered according to the adjacent grouping policy. Consider 50 users non-uniformly distributed in the office space. The goal of the agent was to determine the best RH clustering policy in each episode based on the user distribution in that episode. ***Note that the non-uniform distribution of users in the network was changed after each episode.*** This may be interpreted as the users being mobile and changing their locations at the end of every episode.

At each step, the agent performed an action that was to choose one pair of RHs belonging to different groups and exchange those RHs between the groups. The reasoning behind defining action in such a manner was to maintain the number of RHs per group as four always. This, in turn, was a deliberate decision to control the size of the action space. With such a definition of action, the number of actions from which the agent chose one, at each step, was 448. Note that each episode began with RHs clustered according to the adjacent grouping policy (see Figure 4.4a).

The users in the network were distributed such that some users clustered around a few (one/two) RHs (as shown in Figure 4.4). This was to purposely emulate the scenario of congregation of users in a conference/meeting room in the office space while the rest of the space was sparsely distributed with users.

OPERATION OF THE LEARNING AGENT: For the D-MIMO RH clustering problem, we adopted the Wolpertinger architecture—specifically created for environments with large discrete action spaces—that is based on an actor-critic configuration as described in [68]. We adopted the neural network architecture, including the activation of the different layers, as described in [67] since it was tailor-made to support this framework. The motivation behind the construction of the neural network, the choice of activation functions, and further details regarding the working of the agent can be found in [67,68]. However, we had to tweak certain hyper-parameters, including the learning rates of the actor and critic networks, the number of hidden neurons, and the target update parameters, as listed in Table 4.2.

The agent stored the observed episodes in memory (in a *replay buffer*) and randomly generated samples (called a *mini-batch*) from this buffer that were used to perform

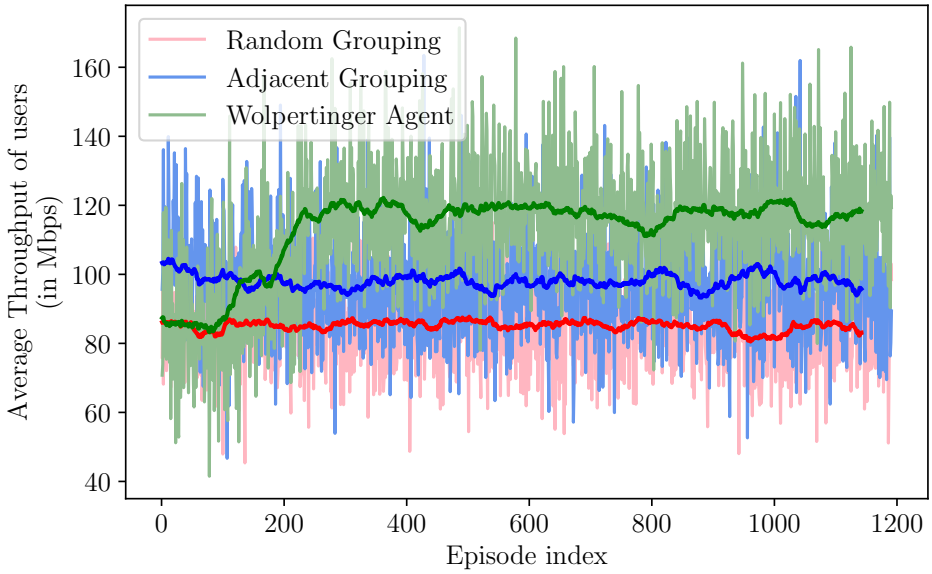


Figure 4.13: Average throughput of users when users were non-uniformly distributed in space (results pertaining to Section 4.5.4)

its training. This idea is called *experience replay* [62, 67, 74] and it helped the agent achieve better convergence behavior and a higher sample efficiency. This is so because the training data are randomly sampled from the replay buffer and hence they act as uncorrelated data that help particularly in on-line training and non-linear function approximation.

Figure 4.13 plots the average throughput performance of users when the Wolpertinger agent was used to update the RH grouping in response to the user distribution in each episode, along with results from two other clustering policies: (i) when RHs were randomly clustered in groups of four (referred to as *random grouping*), and (ii) when RHs were clustered according to the *adjacent grouping* strategy (see Figure 4.4a). Clearly, random clustering performed worse than the other two policies because it was blind to the distribution of users. It is also evident that the Wolpertinger agent was successful in reorganizing the clustering of RHs to achieve a higher user throughput performance compared to random clustering as well as the static adjacent grouping strategy; in fact, the Wolpertinger agent was able to attain an improvement of up to 20% in the performance compared to the latter. This is encouraging because recent implementations of D-MIMO [12, 14] have demonstrated the feasibility of achieving

synchronization among RHs belonging to a group over the air and hence there is no logistical concern regarding clustering RHs located far away from each other. Also, observe that the performance of the Wolpertinger agent was consistent across episodes even though the user distributions were changed in each episode.

4.6 Discussion of Results

Key takeaways from the results, and the lessons learned from implementing the DRL framework are summarized below:

1. DRL agents learned the optimal policies episodically. One learning episode consisted of a fixed number of actions. After the completion of one episode, the network environment was modified. That is, the locations of interferers, channels assigned to the interferers, and the distribution of users in the network space were different between different learning episodes.

2. Since learning agents performed training episodically and since they received a reward for every action that they executed, agents with lower discount factors showcased a better convergence performance compared to agents with higher values.

3. DRL agents employed in this work had fairly simple configurations in terms of the number of hidden layers, number of hidden nodes, and the implemented training algorithm.

4. Using a single hidden layer in the implementation of the learning agent resulted in the following benefits (compared to agents with more hidden layers):

   (a) lower computational complexity,

   (b) lower variance in performance between episodes,

   (c) lower storage requirements (to store the parameters of the neural network), and

   (d) faster training time.

All of these aspects are attractive in the context of mobile networks as it reduces computational, memory, and energy requirements. Although using an agent with more hidden layers may improve training accuracy, the aforementioned advantages make a stronger case for using agents with a single hidden layer.

5. DRL agents performed on-line training, that is, the agents learned with every observed episode (or with a random mini-batch of episodes in case of Section 4.5.4). Furthermore, the agents were fed *simple state information* that could be easily obtained by the network administrator. This is desirable since traditional deep learning methods in the context of wireless networking rely heavily on collecting large amounts of data prior to training, the collection of which may be prohibitively expensive. Also, in the traditional deep learning case, there is a risk of developing a model that may over-fit to the training data if the amount of data collected is not sufficiently large and hence the model may not generalize.

6. The DRL wolpertinger agent was able to respond effectively to non-uniform distribution of mobile users in the network space by updating the clustering of RHs to maximize the throughput performance of users. Building an agent with low complexity was pivotal in this case as it was desired that the agent generalized well among different episodes with vastly different distribution of users. As evidenced by Figure 4.13, the learning agent could successfully generalize between different episodes.

7. Maintaining a replay buffer with previously observed experiences and training the Wolpertinger agent based on a mini-batch randomly sampled from the buffer helped the agent achieve better convergence behavior and sample efficiency in Section 4.5.4. However, this comes at a price–memory/storage overhead to store the replay buffer. It is encouraging to see the success of DRL agents in the channel assignment problems (in Sections 4.5.1, 4.5.2, and 4.5.3) without needing a replay buffer.

4.7 Discussion on the Duration of On-line Learning

In our simulation-based DRL framework, each action involved simulating the D-MIMO Wi-Fi network for a network time of 100 ms. The custom D-MIMO network simulator required about 1.45 s, on average, to complete one simulation using one core of an Intel Xeon Processor E5 v4 family processor. Time required per action of the learning agent may be computed as the sum of (i) the time taken by the agent to choose an action, and (ii) the time required to perform one network simulation. The learning agent required around 175 µs to choose the action at step $t+1$ based on the state at step t; the agent performed its computations in one core of the aforementioned processor. Hence, the time per action ≈ 1.45 s. Since each episode consisted of 50 actions, the time per learning episode = time for 50 actions ($= 50 \times 1.45$ s) + time for training the learning agent at the end of the episode (≈ 500 µs) ≈ 72.5 s.

However, in a real network environment, *the learning agent will not have to simulate the performance of the network*; the agent will collect the information that is required for its training—that is, network state and reward—from the network itself. The time per action in a real system consists of (i) the time required to apply the action, (ii) the time spent collecting feedback (state and reward), and (iii) the time required to compute the next action. Computing the time required to apply the chosen action in the network is a bit involved. For instance, in case of the channel assignment problems, one action corresponds to moving a D-MIMO group—and hence the constituent RHs as well as the users associated with the group—to a different channel. As defined by the IEEE 802.11-2016 standards, a Wi-Fi AP (or a D-MIMO PU) sends a channel switch announcement (CSA), as part of the beacon, to its associated users informing them of the impending switch to a new channel. Based on the details furnished in Sections 9.4.2.19 and 11.9.8.2 in [75], the time required to perform a channel switch of the associated users (and hence the time to apply an action in the network) may be approximated as 400 ms. Note that we assume wired connectivity between the PU (where the learning agent resides) and the RHs and hence no delay in communicating the action from the agent to the RHs. We assume the time for data collection (for feedback) to be 100 ms. This choice worked reasonably well

in case of the simulation-based DRL framework and hence we believe that it will translate nicely into a real system as well. Although collecting data for 100 ms worked acceptably for us, determining the optimal duration of data collection needs further investigation. If it is too low, then the learning agent may not receive sufficient feedback from the network to reasonably estimate the impact of its actions. If it is too high, however, it will prolong the execution of the next action and hence the duration of the learning episodes. The learning agent in our framework took about 175 µs to choose the action for step $t+1$ based on the state at step t (as described in case of the simulation-based DRL framework). Hence, the time per action in a real system may be computed as 175 µs + 100 ms + 400 ms $\approx$ 500 ms. Assuming episodic learning and fifty actions in each episode, the time spent per episode in a real system may be estimated as (50$\times$time per action) + time required for training of the agent at the end of the episode ($\approx$ 500 µs) $\approx$ 25 s.

A strategy to decrease the duration of a learning episode may be to reduce the number of steps/actions in the episode. At the outset, this may seem promising but this may lead to the agent training for a higher number of episodes before its performance converges. For instance, we revisited the problem of channel assignment in D-MIMO Wi-Fi with three external interferers and re-ran the training with twenty steps per episode (instead of fifty). In such a setting, the learning agent trained for around 550 episodes before its performance converged compared to about 200 episodes when each episode consisted of fifty steps (see Figure 4.11b).

Chapter 5

D-MIMO Wi-Fi Networks in mmWave Bands

5.1 Summary and Organization

We extend the concept of D-MIMO and the solutions presented in Chapter 2 to Wi-Fi networks operating in 60 GHz bands (i.e, IEEE 802.11ad networks) with high channel bandwidths (2160 MHz). In particular, the objective of this work is to study the performance improvement attained by using D-MIMO, in terms of modulation-and-coding scheme (MCS) indices and throughput achieved by the users, over a baseline configuration in dense Wi-Fi networks operating in mmWave bands. Realistic network simulations are carried out, using recommended path loss models, and the results of the two arrangements (baseline and D-MIMO) are compared. The contributions of this work are listed below:

- To the best of our knowledge, this is the one of the first studies exploring the potential of D-MIMO in enhancing the performance of Wi-Fi networks in 60 GHz bands.
- This work calls attention to an important behavior wherein the baseline setup is able to achieve a better performance, in terms of supporting high MCS indices at a user, compared to the D-MIMO setup, especially when the user is located close to an AP. This observation provides valuable insights into designing future networks as a compound of both baseline and D-MIMO configurations.

This chapter is organized as follows. Section 5.2 includes a detailed description of the channel model and the different scenarios considered in network simulations. Section 5.3 provides an extensive discussion of the results obtained from rigorous network simulations.

Table 5.1: Details of the network simulation setup

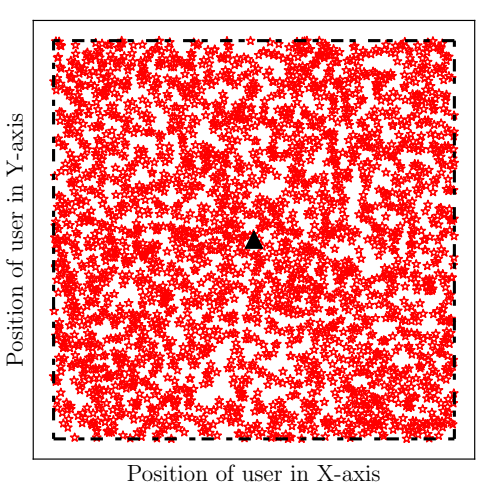
Parameter	Value
Channel center frequency	60 GHz
Channel bandwidth	2160 MHz
Number of available channels	4
Path loss Model	5GCM channel model [76]
Number of APs/RHs	64
AP/RH inter-site distance	10 m
Number of RHs per D-MIMO group	4
Number of D-MIMO groups	16
Number of users	64
Number of antennas (per device)	2
Directionality of antennas	Isotropic
AP/RH height	3 m
User height	1 m
Group power constraint	43 dBm EIRP [77]
Traffic model	Full buffer
SINR to MCS mapping	802.11ad single carrier mode [78]
CCA parameters	
CCA threshold	−67 dBm
DIFS duration	13 µs
SIFS duration	3 µs
Slot duration	5 µs

5.2 Simulation Setup and Channel Model

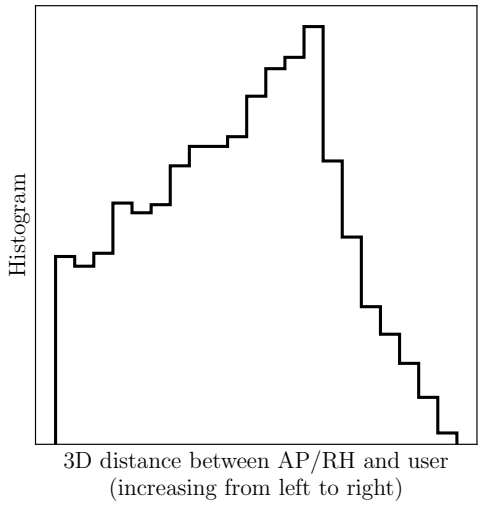
This section describes the setup used for performing network simulations, the different simulation scenarios considered, and the channel path loss model used in the simulations.

The various parameters involved in the simulations are mentioned in Table 5.1. The baseline and D-MIMO network scenarios studied in the simulations are shown in Figure 2.9a and Figure 2.9b respectively. The network consisted of 64 APs/RHs deployed with an inter-site distance of 10 m. The overall network dimensions were 80 m $\times$ 80 m. The RHs were divided in groups of four to form 16 groups. For illustrative convenience, the availability of four non-overlapping channels was assumed, which were assigned to the APs/groups in case of baseline/D-MIMO scenarios as identified by the colors in Figures 2.9a/2.9b.

A NOTE ON THE DISTRIBUTION OF USERS: The users were distributed in the network such that each AP had one associated user. That is, in case of Figures 2.9a and 2.9b,



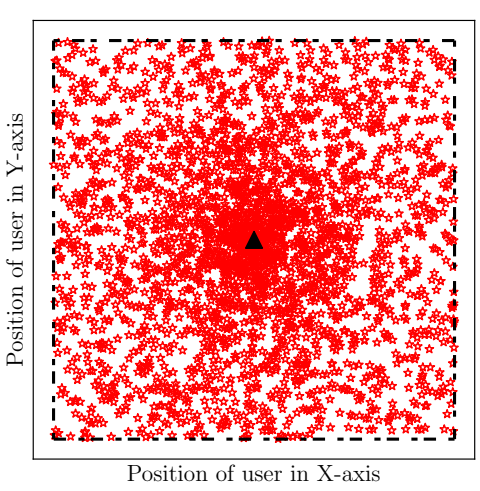
(a) Locations of a user around a AP/RH (represented as stars). The square boundary represents the cell of the AP/RH (represented by the triangle).



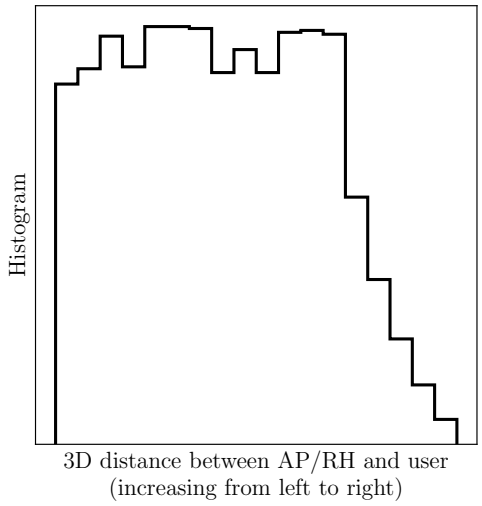
(b) Histogram of distance between user and AP/RH. Note that the distance is 3D because of the difference in height between AP/RH and user (see Table 5.1).

Figure 5.1: Distribution of distances between a user and a AP/RH, and the corresponding histogram when the Cartesian coordinates of the user location were uniformly chosen.

there were 64 users present in the network with exactly one user associated per AP (baseline) or four users associated per group (D-MIMO). We used a distance-based association of a user to an AP or D-MIMO group. That is, a user would associate with its nearest AP in case of baseline or the group containing the nearest RH in case of D-MIMO. We tessellated the network space in Figure 2.9 with square cells with each cell containing one AP/RH. The details of distribution of distances between a user and its associated AP/RH are plotted in Figure 5.1 (results from 3000 random drops a user in a cell). The locations of the user were distributed such that the Cartesian coordinates (x and y) were picked uniformly between the boundaries of the cell (see Figure 5.1a). This, however, led to a triangular distribution of distances between the AP and the user (see Figure 5.1b). This is because sum of squares of two uniformly distributed random variables is triangularly distributed. The implication of this distribution is that there were fewer distances very close to and very far away from the AP/RH compared to medium-range distances. This observation becomes important later when studying the results from network simulations.



(a) Locations of a user around a AP/RH (represented as stars). The square boundary represents the cell of the AP/RH (represented by the triangle).



(b) Histogram of distance between user and AP/RH. Note that the distance is 3D because of the difference in height between AP/RH and user (see Table 5.1).

Figure 5.2: Distribution of distances between a user and a AP/RH, and the corresponding histogram when the AP-user distance was uniformly chosen.

A valid alternative choice to consider to model the distribution of distances between a user and its AP/RH would be uniform. That is, uniformly randomly generate the polar co-ordinates (r, θ) of the location of the user and map it into Cartesian coordinates. The results of such mapping are presented in Figure 5.2. The histogram of distances between the user and the AP/RH follows a uniform distribution. The reason why it declines toward the right is because the generation of distances is constrained by the boundaries of the square cell. Note the dense concentration of user locations close to the AP/RH in Figure 5.2a. We felt that Figure 5.1a is a more realistic modeling of user locations compared to Figure 5.2a and hence generated the user locations, in our simulation studies, by uniformly randomly choosing the Cartesian coordinates of their locations.

The channel model used in the simulations was the 5GCM channel model [76] in the indoor hotspot open office scenario (InH-Open office) which probabilistically models whether a particular link is line-of-sight (LOS) or not. The path loss model used is described below:

- Probability of a link being line-of-sight: P_{LOS}

$$P_{LOS} = \begin{cases} 1, & d_{2D} \leq 1.2\,m \\ \exp(-(d_{2D} - 1.2)/4.7), & 1.2\,m < d_{2D} < 6.5\,m \\ \exp(-(d_{2D} - 6.5)/32.6) \cdot 0.32, & 6.5\,m \leq d_{2D} \end{cases} \tag{5.1}$$

- Path loss in case of a line-of-sight link: PL_{LOS}

$$PL_{LOS} = 32.4 + 17.3 \log_{10}(d_{3D}) + 20 \log_{10}(f_c)$$

  with standard deviation of shadow fading (σ_{SF}) = 3.02 dB

- Path loss in case of a non line-of-sight link: PL_{NLOS}

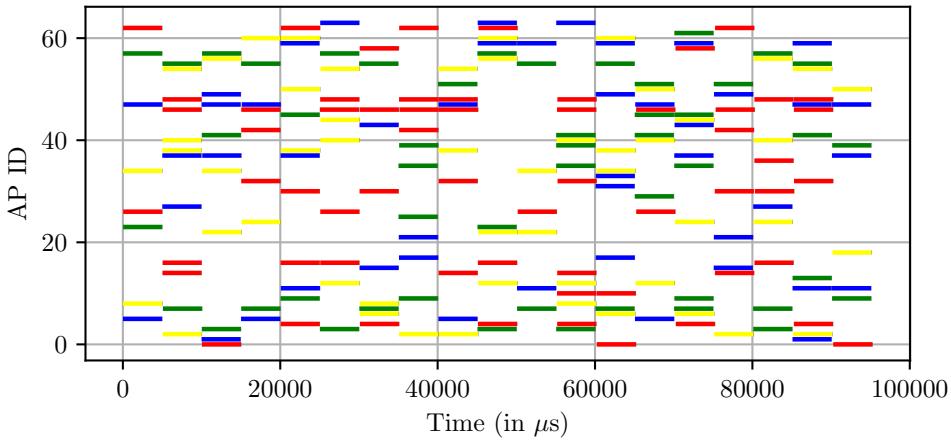
$$PL_{NLOS} = 38.3 \log_{10}(d_{3D}) + 17.30 + 24.9 \log_{10}(f_c)$$

  with standard deviation of shadow fading (σ_{SF}) = 8.03 dB

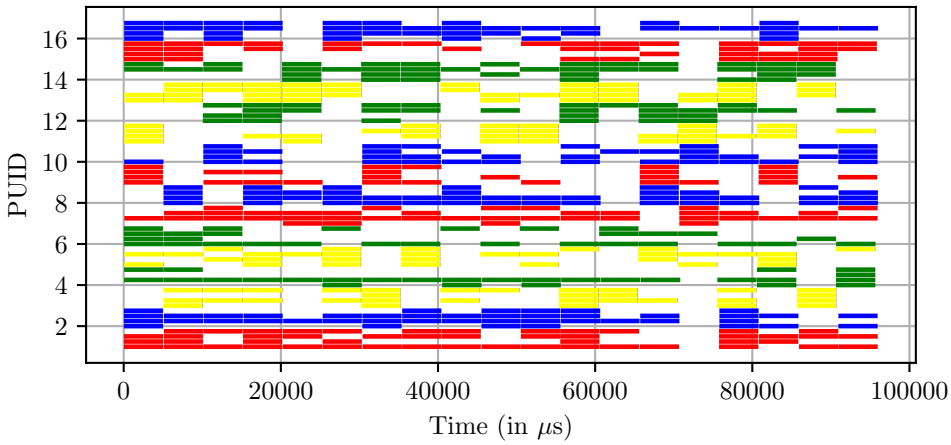
In the above model, d_{2D} and d_{3D} denote the 2D and 3D distance (in m) between the transmitter and receiver of a link respectively, and f_c represents the carrier frequency (in GHz). We deliberately did not use the IEEE 802.11 TGad channel model [79, 80] because although the documents prescribe path loss models separately for LOS and NLOS scenarios to be used for network simulations, they do not describe how to model whether a link is LOS or not. Power allocation among the RHs belonging to group was performed as described in Section 2.5, assuming zero-forcing precoding to cancel interference between streams. Two streams were formed to each user since a user was equipped with two antennas.

5.3 Network Simulation Results and Discussion

The following presented results were obtained from 3000 simulation runs with a random drop of users in each run. Each simulation was executed for a network time of 100 ms. The simulation seeds (corresponding to the simulation runs) were maintained to be the same between baseline and D-MIMO configurations in order to make the comparison fair.



(a) Baseline scenario



(b) D-MIMO scenario

Figure 5.3: Channel occupancy of APs/RHs plotted as line diagrams. The length of a line corresponds to the duration for which the corresponding AP/RH was able to access a channel. Each channel is color-coded uniquely.

5.3.1 Channel Access Characteristics

Figure 5.3 describes the channel access characteristics of both baseline and D-MIMO arrangements as line diagrams. The x-axis plots the duration of the simulation time (100 ms) and the y-axis plots the index of AP/group. The length of a line corresponds to the duration for which a AP/RH obtained access to a channel. The colors uniquely represent the four non-overlapping channels considered available in the simulations.

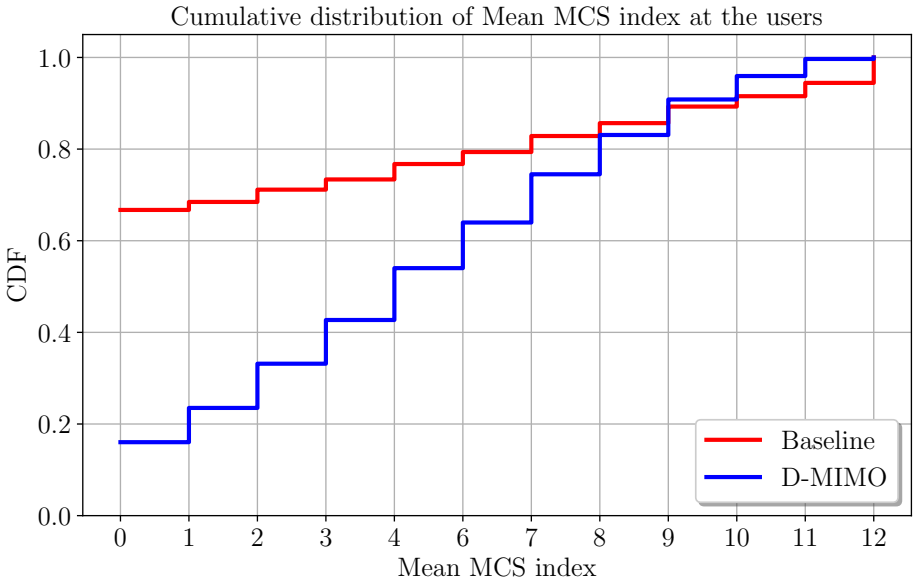
As observed in the case of simulations in 5 GHz bands (see Figure 2.10), the baseline APs did not gain access to a channel frequently (bars are short and spaced far apart

in Figure 5.3a). This behavior can be ascribed to the fact that multiple co-channel APs were in close proximity to each other which resulted in increased channel contention and lower duration of channel access. In contrast, the channel access times are notably better for the D-MIMO configuration; observe that the bars are consistently long and are spaced at small regular intervals in Figure 5.3b. Hence, the proposed medium access protocol for D-MIMO (in Section 2.3) helped RHs achieve better access to channels compared to the baseline arrangement. In fact, D-MIMO was able to attain a 75% reduction in average channel access delay of RHs compared to baseline APs.

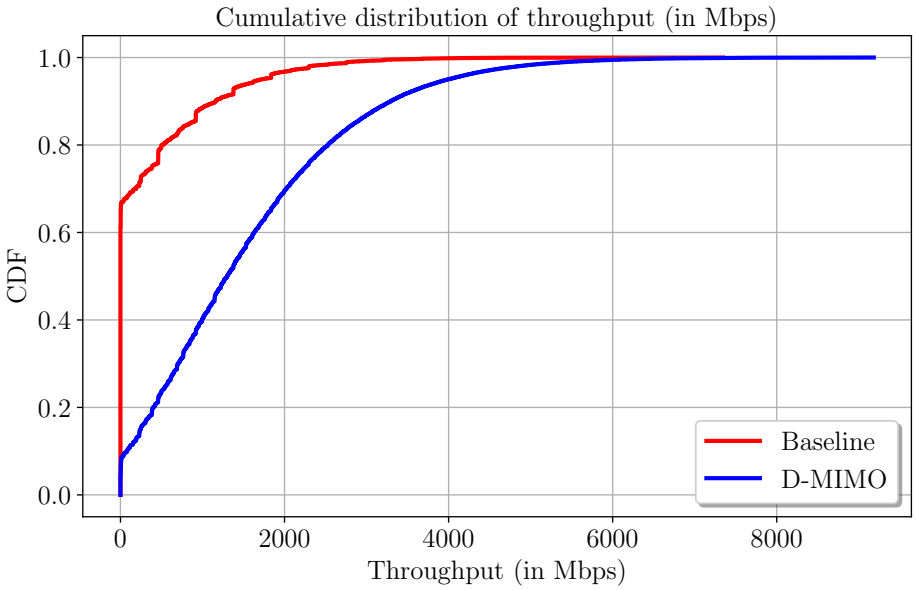
5.3.2 User Throughput Characteristics

Figure 5.4a plots the cumulative distribution of the mean modulation and coding scheme (MCS) index chosen by the users in all simulations. The MCS index for each transmission was obtained from the tables presented in [78] that map receiver sensitivity (and in turn the signal-to-noise ratio (SNR)) to the corresponding MCS index for single carrier (SC) mode of operation. Note that in SC mode, both MCS5 and MCS7 are supported when receiver sensitivity is -62 dBm but MCS7 is chosen as it provides a higher achievable rate [78] (notice the absence of MCS5 in Figure 5.4a). Since the network simulation time was longer than the duration of a transmission opportunity (TXOP), the mean MCS index for a user s was computed as the average of MCS indices observed over all TXOPs in which user s was active in that simulation run. For the most part, D-MIMO was able to achieve a better MCS index than the baseline arrangement, especially in the range of low/medium MCS indices. However, an interesting cross-over in the trends was observed in case of high MCS indices (particularly after MCS9). It can be deduced from Figure 5.4a that the probability of achieving high MCS indices (greater than MCS9) is higher with baseline compared to D-MIMO configuration. It is imperative to understand why this cross-over occurs as it might yield valuable insights into designing, potentially, a hybrid network involving both baseline and D-MIMO arrangements.

Figure 5.5 describes the histogram of distances from AP/RH at which different MCS indices were observed at a user in case of baseline and D-MIMO configurations. The



(a) CDF of mean MCS index chosen at the users. Notice the cross-over in trends at high MCS indices (greater than 9).



(b) CDF of throughput achieved by users. D-MIMO achieved a ~ 395% improvement in average user throughput.

Figure 5.4: Cumulative distribution of the mean MCS index and average throughput achieved by the users in the simulations.

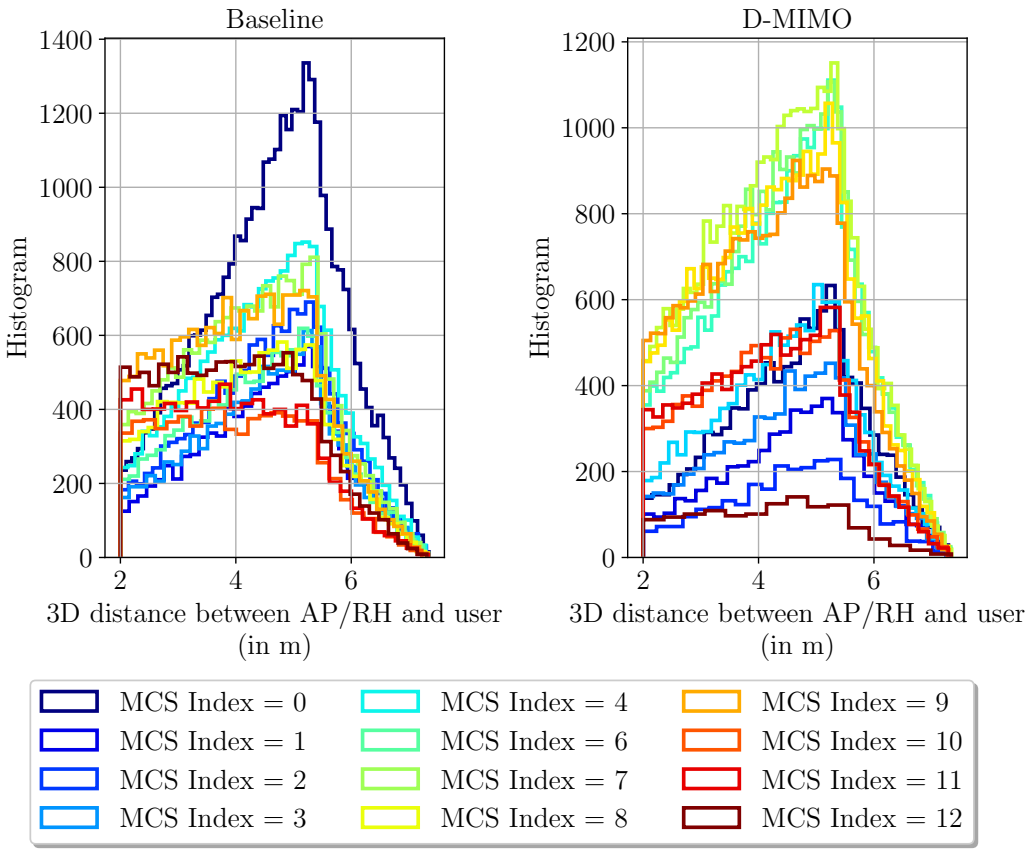


Figure 5.5: Histogram of distances from AP/RH at which different MCS indices were achieved by a user in case of baseline and D-MIMO scenarios. Note that the distance is 3D since APs/RHs are located higher than users.

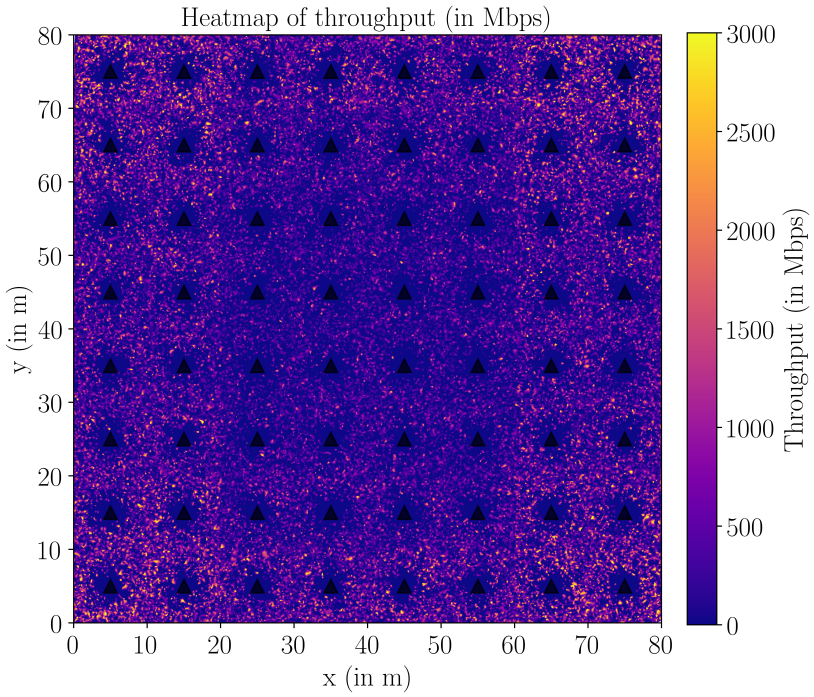
shape of the distribution of distances between a user and its associated AP/RH has been discussed in Figure 5.1b. In the following discussion, we use AP in the context of baseline and RH in the context of D-MIMO arrangements. First, consider the baseline scenario. It is evident that baseline was able to support very high MCS indices (greater than 10) at a user when it was located close to its associated AP (observe brown/dark lines achieve higher numbers than other curves). This was the case because when the user was located close to its associated AP, the probability of the link being line-of-sight (LOS) was high (equation 5.1) which corresponded to a higher signal power and in turn a high MCS index achieved at the user. However, as the user moved farther away from the AP, the baseline setup could support only low MCS indices. Notice how the blue/dark blue curves start low when the user was located close the AP (indicating that at close distances, baseline could support higher MCS indices) but they grow as

the distance between the AP and user increased. As the distance of the AP–user link increased, two factors were at play: (i) lower probability of LOS of the AP–user link, and (ii) increased interference from other co-channel APs. In fact, at distances greater than 4 m, MCS0 was the dominant index, indicating that the baseline network was highly interference limited.

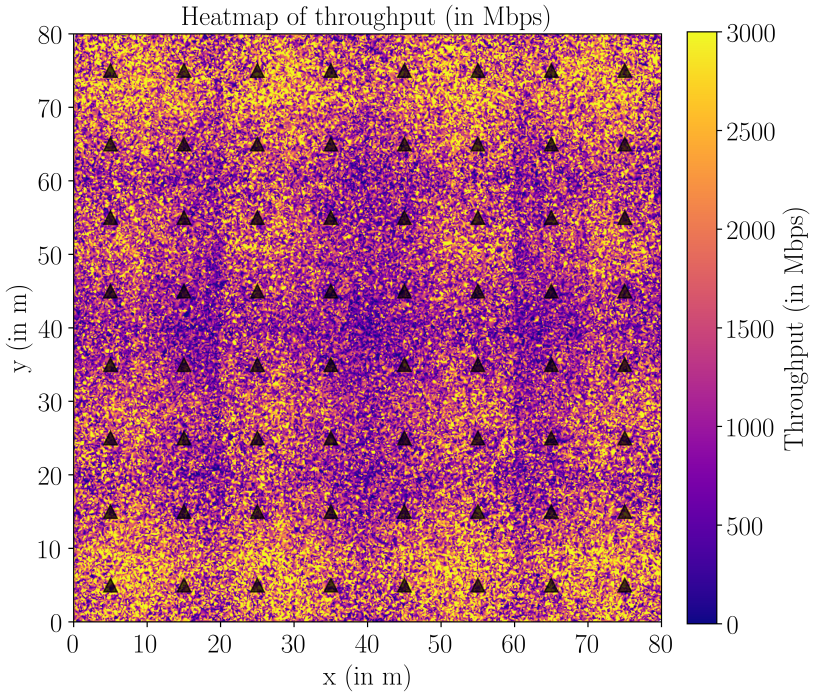
On the flip side, consider the D-MIMO case. It could not support very high MCS indices to the user when it was located close to one of the RHs of the D-MIMO group. Notice the brown line (corresponding to MCS12) is the lowest among all other curves at close distances to the RH. This behavior is a consequence of using zero-forcing precoding to facilitate MU-MIMO transmissions in D-MIMO (as described in Section 2.5). The precoding algorithm cancels inter-stream interference but at the expense of some signal power in the streams. When a user was located very close to one of the RHs of the D-MIMO group, the algorithm would have had to sacrifice more signal power in the streams for this user in order to minimize interference to other streams, which led to a lower MCS index being achieved at this user. However, at distances greater than 4 m, D-MIMO performed better than baseline. Observe that D-MIMO could support MCS between 7 and 9 in this distance range whereas baseline achieved MCS0 more frequently. The plot also shows that D-MIMO did not perform poorly when the user was located close to a RH. In fact, in this distance range, D-MIMO was able to achieve MCS between 7 and 9 compared to baseline supporting between 10 and 12.

The aforementioned discussion brings to light an important behavior that will prove valuable in the design of future networks, possibly as a hybrid of the two arrangements: *if a user is located close to a AP/RH, baseline should be used to achieve a very high MCS index at the STA; else, if the user is not located close to a AP/RH, D-MIMO should be preferred to achieve better MCS indices than baseline*.

Figure 5.4b describes the cumulative distribution of throughput achieved by users across all the simulation runs. It is apparent that the D-MIMO configuration was able to provide better throughput to users compared to the baseline setup. In fact, the D-MIMO scenario achieved a **395%** improvement in average user throughput compared to the baseline case.



(a) Baseline scenario



(b) D-MIMO scenario

Figure 5.6: Heatmap of mean throughput achieved achieved by users across all simulation runs. Triangles represent APs in baseline and RHs in D-MIMO arrangements respectively.

Figures 5.6a and 5.6b illustrate the spatial distribution of throughput attained by users from all the simulations with baseline and D-MIMO scenarios respectively. The maps are color coded in such a way that users in lighter regions attain higher throughput compared to users in darker regions. It can be seen that users achieve a higher throughput with D-MIMO compared to the baseline setup over the entire network space. Furthermore, the distribution of throughput across the spatial dimensions is equitable with D-MIMO.

Chapter 6

Conclusions

Emerging data-intensive applications such as augmented and virtual reality (AR/VR) and 8K video streaming will proliferate throughput requirements of next-generation Wi-Fi networks. To meet the rising throughput demands over time, Wi-Fi has steadily added support for wider channel bandwidths, including 40 MHz in 802.11n (in 2.4 GHz), 80/160 MHz in 802.11ac/ax (in 2.4/5 GHz), and 320 MHz now under consideration [1]. Another approach to achieving higher throughput is to 'densify' the network, i.e, reduce the distance between neighboring Wi-Fi access points (APs). This allows each AP to serve a smaller area and provide its users with a higher average signal to noise ratio (SNR) and hence better data rates. In practice, however, these two approaches of using wider channels and 'densifying' networks are incongruent. Interference between closely-spaced APs (due to their dense deployment) limits availability of the larger bandwidth channels, which are accessed on a best-effort basis, causing devices fall back to narrower channels.

The focus of this dissertation was to explore the use of distributed multi-user (MU) MIMO (hereon referred to as D-MIMO) architecture to boost network performance (be it user throughput or channel access) in dense Wi-Fi networks. A D-MIMO system consists of several time and phase-synchronized APs that jointly transmit and receive signals, thereby acting as a single spatially-distributed virtual antenna array to simultaneously serve multiple users. The cooperation between APs reduces intra-network interference and hence improves spatial reuse of channels.

The central idea of D-MIMO is to divide the functionality of a quintessential Wi-Fi access point (AP) into two entities (as shown in Figure 1.1): (i) a radiohead (RH), which is a simple remote radio front end that transmits and receives wireless waveforms,

and (ii) a processing unit (PU), to which all the other functionalities of a Wi-Fi AP are off-loaded. A D-MIMO group consists of multiple phase-synchronized RHs located at different spatial positions and they cooperatively serve several users simultaneously.

6.1 Takeaways from Each Chapter

The conclusions and key takeaways from each chapter of this dissertation are described below:

- **Chapter 2** considered dense Wi-Fi networks, operating in 5 GHz bands, and explored the potential of employing the D-MIMO architecture to improve network performance compared to state-of-the-art Wi-Fi access points with co-located antennas (baseline configuration). Realizing D-MIMO Wi-Fi networks invited us to rethink some fundamental concepts like Wi-Fi channel access and downlink MU-MIMO user selection. We prescribed novel lightweight solutions to effectively address these challenges. For the channel access problem, we assimilated channel sensing observations of radioheads (RHs) belonging to a D-MIMO group and proposed various strategies to resolve channel contention for the group. We harnessed the theory of weak channel reciprocity to effectively choose users to serve with MU-MIMO transmission—without requesting channel state information (CSI) feedback from them—in every transmission opportunity in order to maximize group throughput performance. These solutions along with the general architectural change enabled D-MIMO Wi-Fi networks to achieve an enhancement of up to 3.5$\times$ in median user throughput performance, an improvement of 191% in average user throughput, and a gain of 345% in the throughput of tenth percentile of users; these results were obtained from extensive network simulations performed using a custom simulator. The D-MIMO architecture facilitated better channel access as well; the channel access time of D-MIMO RHs was 62% lower than baseline APs.

- **Chapter 3** described the implementation of a D-MIMO Wi-Fi group using software-defined radio platforms in an indoor testbed as a proof-of-concept of the lightweight

user selection algorithm proposed in Chapter 2. Specifically, we implemented a D-MIMO Wi-Fi group—compliant with the IEEE 802.11ac standards—consisting of four RHs and twenty users using universal software radio peripherals (USRPs) in the indoor ORBIT testbed at WINLAB. The RHs were realized using USRP X310s and were synchronized by a GPS-disciplined clock reference system in order to establish tight synchronization among them in phase and time. The users were deployed using a combination of USRP B210s and X310s and were located in different parts of the testbed. Extensive experimental evaluations on this setup corroborated the results from Chapter 2, i.e, the proposed lightweight user selection algorithm was successful in choosing users to serve that could maximize group throughput performance in every TXOP without requesting CSI feedback from all users associated with the D-MIMO group. Results from experiments revealed that the proposed algorithm could achieve an improvement of up to 60% in median and 43% in average group throughput performance compared to a simple random user selection strategy. We also performed Oracle-based user selection wherein an all-knowing Oracle could optimally pick users to maximize the group throughput performance in every TXOP. The difference in performance between the proposed algorithm and the Oracle-based user selection was observed to be just 13%, further underscoring the effectiveness of the proposed algorithm.

- **Chapter 4** explored the potential of harnessing concepts from deep reinforcement learning (DRL) to address two major dynamic resource management problems pertaining to D-MIMO Wi-Fi networks: (i) *optimal channel assignment of groups*, and (ii) *optimal clustering of radio heads to form groups*, to maximize user throughput performance. These problems are known to be NP-Hard for which only heuristic solutions exist in literature. A DRL framework was constructed to effectively address the aforementioned problems. This chapter considered practical dynamic network scenarios in which users were mobile and could be distributed non-uniformly in space, and the network itself was subjected to random external Wi-Fi interference. The implemented DRL agents belonged to the policy iteration class, owing to the vastness of the state and action spaces of the considered

scenarios. Through extensive network simulations and on-line training of the learning agents, this work demonstrated that DRL agents could successfully address the aforesaid problems as well as achieve an improvement of up to 20% in user throughput performance compared to popular heuristic solutions. The DRL agents were also more effective, compared to heuristic solutions, in simultaneously meeting multiple network objectives, say maximize throughput of users as well as fairness of throughput distribution among them.

- **Chapter 5** focused on exploring the potential of using D-MIMO as a technique to enhance the throughput performance of Wi-Fi networks in mmWave bands. Particularly, it compared the performance of a D-MIMO Wi-Fi network operating in 60 GHz bands (with high channel bandwidths) with a baseline setup in terms of the modulation-and-coding scheme (MCS) indices and average throughput achieved at the users. Results from rigorous network simulations revealed that D-MIMO was able to attain an improvement of up to 395% in average user throughput compared to the baseline configuration. The simulation results also brought to light an interesting behavior wherein the MCS performance of users was better with baseline compared to D-MIMO, especially in achieving high MCS indices (greater than MCS9). This observation was substantiated based on the histogram of distances at which different MCS indices were attained and this behavior was ascribed to the use of zero-forcing as the transmit precoding algorithm of choice. Additionally, this chapter provided a guideline for the design of future networks—*if a user is located in close proximity to a AP/RH, the baseline setup will be a better choice to achieve a very high MCS index at the user; otherwise, D-MIMO achieves a higher MCS index at the user compared to baseline.* A corollary to this observation is that if a user needs to be served with very high MCS indices, it might be better off moving close to its associated AP and the latter using single-user MIMO service. This observation is attractive since recent works [14] have shown that over-the-air synchronization of RHs for D-MIMO is feasible and hence future networks may harness the benefits of both baseline and D-MIMO configurations by dynamically switching between the two depending on the

distribution of users.

6.2 Looking Ahead: Future Research Directions

The training results provided in Section 4.5 serve as an impetus to continue studying the use of DRL to solve more complex and non-trivial problems in D-MIMO Wi-Fi networks. Further research directions may be along the following lines:

1. D-MIMO RH GROUPING: While addressing the problem of RH grouping in D-MIMO networks in Section 4.5.4, a restriction on the number of RHs per group was imposed. It will be interesting to study the performance of the DRL agent if such restrictions are relaxed and the agent is free to cluster the RHs in an unconstrained manner. Additionally, channel assignment and RH grouping may be combined together, that is, a DRL agent may update the clustering of RHs as well as the group-channel assignment in response to the distribution of users in the network.

2. MEETING MULTIPLE CONFLICTING OBJECTIVES: The multiple objective problem in Section 4.3.3 considered two objectives that went hand-in-hand with each other. However, there may exist simultaneous network objectives that need not be symbiotic, that is, improvements in one objective may lead to deterioration of others. In such cases, the multi-objective reinforcement learning framework becomes more complex with a single policy solution becoming highly improbable and the agent develops a convex convergence set of policies to meet different objectives of differing priority levels [81]. The problem formulation usually requires a trade-off to be made between these objectives, especially when they are disjointed.

3. PARAMETER SETTINGS FOR DRL: The training of DRL agents involves extensive efforts of trial and error, especially in the setting of different hyper-parameters of the neural network, deciding the contents of the state information fed to the agent from the environment, and the formulation of rewards. Searching for the

optimal configuration of the aforementioned parameters is analogous to looking for a needle in a haystack. Furthermore, these parameters have a high impact on the performance of the learning model. There is some recent work addressing this challenge by employing a progressive neural architecture search [82] but this is a computationally expensive task.

4. INTERPRETABILITY OF LEARNING ALGORITHMS: Although it is evident from the training results that DRL agents were successful in performing better than heuristic solutions, we are still limited in our understanding of why the agents made certain decisions. This lack of interpretability has been widely regarded as a major reason impeding the pervasive application of DRL in a variety of domains, including the networking industry. Active research has been undertaken to address this limitation and facilitate a better interpretability of learning algorithms [83].

In the context of Wi-Fi networks operating in mmWave bands, the results in Section 5.3 show great promise in the ability of the D-MIMO architecture to enhance user throughput performance. Further research may involve implementing a D-MIMO Wi-Fi group using software defined radio platforms in an indoor testbed—as is the case in Chapter 3—but operating in mmWave bands. This will be a lucrative exercise since signal propagation in mmWave bands is susceptible to a variety of factors, including high path loss, high scattering and blockage losses, and so on, and existing path loss models in literature do not capture all of these effects comprehensively. Furthermore, since the channel bandwidths in mmWave bands are high (up to 2160 MHz), the amount of data transferred between the RHs and the PU will be high. Hence, it will be interesting to study how to connect the RHs to the PU (choice between wired or wireless); the links between these entities must be able to handle such large amounts of data as well as meet the strict timing requirements of Wi-Fi. Additionally, research efforts may be dedicated in the direction of designing future dense Wi-Fi networks that can harness the benefits of both baseline and D-MIMO configurations (as described in Chapter 5); that is, the networks should be able to dynamically switch between the two arrangements depending on the distribution of users.

References

[1] L. Cariou *et al.* (2018, May) EXtreme Throughput (XT) 802.11, Doc.: IEEE 802.11-18/0789r10. [Online]. Available: https://mentor.ieee.org/802.11/dcn/18/11-18-0789-10-0wng-extreme-throughput-802-11.pptx (Accessed 2019-10-10).

[2] X. Chen *et al.* (2018, September) Discussions on the PHY features for EHT, Doc.: IEEE 802.11-18/1461r0. [Online]. Available: https://mentor.ieee.org/802.11/dcn/18/11-18-1461-00-0eht-discussions-on-the-phy-features-for-eht.pptx (Accessed 2019-10-10).

[3] B. Yang *et al.* (2018, September) Considerations on AP Coordination, Doc.: IEEE 802.11-18-1576-01-0eht. [Online]. Available: https://mentor.ieee.org/802.11/dcn/18/11-18-1576-01-0eht-considerations-on-ap-coordination.pptx (Accessed 2019-10-10).

[4] D. López-Pérez *et al.* (2019, January) Distributed MU-MIMO architecture design considerations, Doc.: IEEE 802.11-18/1190r1. [Online]. Available: https://mentor.ieee.org/802.11/dcn/19/11-19-0089-01-0eht-distributed-mu-mimo-architecture-design-considerations.pptx (Accessed 2019-10-10).

[5] D. López-Pérez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11 be – Extremely High Throughput: The Next Generation of Wi-Fi Technology Beyond 802.11 ax," *arXiv:1902.04320*, 2019.

[6] D. Gesbert, M. Shafi, D.-s. Shiu, P. J. Smith, and A. Naguib, "From Theory to Practice: An Overview of MIMO Space–Time Coded Wireless Systems," *IEEE Journal on Selected Areas In Communications*, vol. 21, no. 3, p. 281, 2003.

[7] W. Roh and A. Paulraj, "Outage performance of the distributed antenna systems in a composite fading channel," in *Proceedings IEEE 56th Vehicular Technology Conference*, vol. 3. IEEE, 2002, pp. 1520–1524.

[8] H. Dai, H. Zhang, and Q. Zhou, "Some analysis in distributed mimo systems." *JCM*, vol. 2, no. 3, pp. 43–50, 2007.

[9] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: scaling wireless capacity with user demands," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 2012, pp. 235–246.

[10] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "AirSync: Enabling distributed multiuser MIMO with full spatial multiplexing," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 6, pp. 1681–1695, 2013.

[11] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving high data rates in a distributed MIMO system," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 41–52.

[12] E. Hamed, H. Rahul, M. A. Abdelghany, and D. Katabi, "Real-time distributed MIMO systems," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 2016, pp. 412–425.

[13] T. Wang, Q. Yang, K. Tan, J. Zhang, S. C. Liew, and S. Zhang, "DCAP: Improving the capacity of WiFi networks with distributed cooperative access points," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 320–333, 2018.

[14] E. Hamed, H. Rahul, and B. Partov, "Chorus: Truly Distributed Distributed-MIMO," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 2018, pp. 461–475.

[15] X. Zhang, K. Sundaresan, M. A. A. Khojastepour, S. Rangarajan, and K. G. Shin, "NEMOx: Scalable network MIMO for wireless networks," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 453–464.

[16] N. Nurani Krishnan, E. Torkildson, E. Rantala, I. Seskar, N. Mandayam, and K. Doppler, "D-MIMOO–Distributed MIMO for Office Wi-Fi Networks," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2018, pp. 1–10.

[17] N. Nurani Krishnan, E. Torkildson, N. Mandayam, D. Raychaudhuri, E. Rantala, and K. Doppler, "Optimizing Throughput Performance in Distributed MIMO Wi-Fi Networks Using Deep Reinforcement Learning," *IEEE Transactions on Cognitive Communications and Networking (TCCN)*, forthcoming. [Online]. Available: https://dx.doi.org/10.1109/TCCN.2019.2942917

[18] N. Nurani Krishnan, I. Seskar, and N. Mandayam, "Distributed Multi-User MIMO Wi-Fi Networks in 60 GHz Bands: Are They Better Than Baseline Always?" *currently under review for publication*.

[19] N. Nurani Krishnan, G. Sridharan, I. Seskar, and N. Mandayam, "Coverage and rate analysis of super Wi-Fi networks using stochastic geometry," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2017, pp. 1–10.

[20] N. Nurani Krishnan, R. Kumbhkar, N. Mandayam, I. Seskar, and S. Kompella, "Co-existence of radar and communication systems in CBRS bands through downlink power control," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 713–718.

[21] N. Nurani Krishnan, R. Kumbhkar, N. Mandayam, I. Seskar, and S. Kompella, "How Close Can I Be?-A Comprehensive Analysis of Cellular Interference on ATC Radar," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[22] N. Nurani Krishnan, N. Mandayam, I. Seskar, and S. Kompella, "Experiment: Investigating Feasibility of Coexistence of LTE-U with a Rotating Radar in CBRS Bands," in *2018 IEEE 5G World Forum (5GWF)*. IEEE, 2018, pp. 65–70.

[23] C.-X. Wang, X. Hong, X. Ge, X. Cheng, G. Zhang, and J. Thompson, "Cooperative MIMO channel models: A survey," *IEEE Communications Magazine*, vol. 48, no. 2, 2010.

[24] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 146–157.

[25] S. Venkatesan, A. Lozano, and R. Valenzuela, "Network MIMO: Overcoming Intercell Interference in Indoor Wireless Systems," in *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*. IEEE, 2007, pp. 83–87.

[26] H. Huang, M. Trivellato, A. Hottinen, M. Shafi, P. J. Smith, and R. Valenzuela, "Increasing downlink cellular throughput with limited network MIMO coordination," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, 2009.

[27] S. A. Ramprashad, H. C. Papadopoulos, A. Benjebbour, Y. Kishiyama, N. Jindal, and G. Caire, "Cooperative cellular networks using multi-user MIMO: trade-offs, overheads, and interference control across architectures," *IEEE Communications Magazine*, vol. 49, no. 5, 2011.

[28] H. Huh, A. M. Tulino, and G. Caire, "Network MIMO with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2911–2934, 2012.

[29] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: cooperative packet recovery using physical layer information," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 147–158.

[30] M. Gowda, S. Sen, R. R. Choudhury, and S.-J. Lee, "Cooperative packet recovery in enterprise WLANs," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1348–1356.

[31] T. Bansal, B. Chen, P. Sinha, and K. Srinivasan, "Symphony: Cooperative packet recovery over the wired backbone in enterprise WLANs," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 351–362.

[32] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 387–398.

[33] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Practical MU-MIMO user selection on 802.11 ac commodity networks," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 122–134.

[34] Y. Zeng, I. Pefkianakis, K.-H. Kim, and P. Mohapatra, "MU-MIMO-Aware AP Selection for 802.11Ac Networks," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. Mobihoc '17, 2017, pp. 19:1–19:10.

[35] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Proceedings of PIMRC'96-7th International Symposium on Personal, Indoor, and Mobile Communications*, vol. 2. IEEE, 1996, pp. 392–396.

[36] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, IEEE Std 802.11ac™, 2013.

[37] V. Erceg, "IEEE P802. 11 wireless LANs TGn channel models," *IEEE 802.11-03/940r4*, 2004.

[38] SNR-MCS mapping for 802.11ax. [Online]. Available: https://mentor.ieee.org/802.11/dcn/15/11-15-1070-03-00ax-1024-qam-proposal.ppt (Accessed 2019-10-10).

[39] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Tech. rep., Department of CIS, The Ohio State University, Tech. Rep., 1999.

[40] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3. IEEE, 2005, pp. 1664–1669.

[41] M. Ettus and M. Braun, "The Universal Software Radio Peripheral (USRP) family of low-cost SDRs," *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, pp. 3–23, 2015.

[42] Ettus Research. OctoClock CDA-2990 – Ettus Knowledge Base. [Online]. Available: https://kb.ettus.com/OctoClock_CDA-2990 (Accessed 2019-10-10).

[43] Clock reference distribution system in ORBIT testbed. [Online]. Available: https://www.orbit-lab.org/wiki/Hardware/dInfrastructure/tClockReference (Accessed 2019-10-10).

[44] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 159–170.

[45] S. Chieochan, E. Hossain, and J. Diamond, "Channel assignment schemes for infrastructure-based 802.11 wlans: A survey." *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 124–136, 2010.

[46] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath, "Networked MIMO with clustered linear precoding," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, 2009.

[47] A. Papadogiannis, D. Gesbert, and E. Hardouin, "A dynamic clustering approach in wireless networks with multi-cell cooperative processing," in *2008 IEEE International Conference on Communications*. IEEE, 2008, pp. 4033–4037.

[48] J. Liu and D. Wang, "An improved dynamic clustering algorithm for multi-user distributed antenna system," in *2009 International Conference on Wireless Communications & Signal Processing*. IEEE, 2009, pp. 1–5.

[49] R. Weber, A. Garavaglia, M. Schulist, S. Brueck, and A. Dekorsy, "Self-organizing adaptive clustering for cooperative multipoint transmission," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2011, pp. 1–5.

[50] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[51] K. C.-J. Lin, W.-L. Shen, M.-S. Chen, and K. Tan, "User-Centric Network MIMO With Dynamic Clustering," *IEEE/ACM Transactions on Networking*, 2017.

[52] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[53] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–6.

[54] Y. He, N. Zhao, and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, 2018.

[55] C. Wang, J. Wang, X. Zhang, and X. Zhang, "Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 858–862.

[56] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1871–1879.

[57] Z. Zhao, R. Li, Q. Sun, Y. Yang, X. Chen, M. Zhao, H. Zhang *et al.*, "Deep Reinforcement Learning for Network Slicing," *arXiv preprint arXiv:1805.06591*, 2018.

[58] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[59] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, 2019.

[60] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 19–31, 2005.

[61] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[62] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[63] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[64] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[65] L. Weng. Policy Gradient Algorithms. [Online]. Available: https://lilianweng.github.io/lil-log/2018/04/08/policy-gradient-algorithms.html#policy-gradient-theorem (Accessed 2019-10-10).

[66] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[67] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[68] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, and B. Coppin, "Deep reinforcement learning in large discrete action spaces," *arXiv preprint arXiv:1512.07679*, 2015.

[69] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[70] A. Martín *et al.* (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. [Online]. Available: http://tensorflow.org/ (Accessed 2019-10-10).

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[72] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[73] K. Van Moffaert, M. M. Drugan, and A. Nowe, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2013.

[74] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in Neural Information Processing Systems*, 2017, pp. 5048–5058.

[75] *IEEE 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*, IEEE Standard 802.11, Dec 2016.

[76] 5GCM. 5G Channel Model for bands up to 100 GHz. [Online]. Available: http://www.5gworkshops.com/5GCM.html (Accessed 2019-10-10).

[77] S.-K. Yong, P. Xia, and A. Valdes-Garcia, *60GHz Technology for Gbps WLAN and WPAN: from Theory to Practice*. John Wiley & Sons, 2011.

[78] J. Kim, L. Xian, and A. S. Sadri, "60 GHz Modular Antenna Array Link Budget Estimation with WiGig Baseband and Millimeter-Wave Specific Attenuation," *International Journal of Antennas and Propagation*, vol. 2017, 2017.

[79] A. Maltsev, R. Maslennikov, A. Sevastyanov, A. Lomayev, and A. Khoryaev, "Statistical channel model for 60 GHz WLAN systems in conference room environment," in *Proceedings of the Fourth European Conference on Antennas and Propagation*. IEEE, 2010, pp. 1–5.

[80] A. Maltsev *et al.*, "Channel Models for 60 GHz WLAN Systems," doc: IEEE 802.11-09/0334r8, Tech. Rep., 05 2010.

[81] K. Van Moffaert and A. Nowé, "Multi-objective reinforcement learning using sets of pareto dominating policies," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3512, 2014.

[82] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[83] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao *et al.*, "Interpretability of deep learning models: a survey of results," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2017, pp. 1–6.