OPTIMIZING DENSE WIRELESS NETWORKS OF MIMO LINKS

A Thesis
Presented to
The Academic Faculty

by

Luis Miguel Cortés-Peña

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2014

OPTIMIZING DENSE WIRELESS NETWORKS OF MIMO LINKS

Approved by:

Professor Douglas M. Blough, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor John R. Barry, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Xiaoli Ma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Carlos Santamarina
School of Civil and Environmental Engineering
Georgia Institute of Technology

Professor Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mary Ann Weitnauer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 12th of May 2014

To my loving Family,

always united

ACKNOWLEDGMENTS

I want to thank my advisors John Barry and Douglas Blough. It has been both a privilege and honor working with you guys. I have learned and matured a great deal thanks to you two. Your valuable inputs, even when conflicting, taught me how to deal with difficult problems and how to communicate my ideas more effectively. I am forever grateful for all your support.

I also want to thank my family. My father Miguel Cortés. My mother, alias *The Big Mama*, Yolanda Peña. My sisters Yomi Patricia and Aida Yoguely. My brother Yoel René. My wife Eliana Cristina. Thank you for all your support, for all your love, for all your patience, for all those special moments with me, for feeding me so that I could work late, for entertaining me and bringing me joy, for listening to me, and for helping when I encountered difficult times.

I want to thank all my friends and colleagues: Anjuli Patel, Vahid Mustafic, Jorge Quevedo, Gedeon Kamga (Dr. K), Amin Rida, Chika Umolu, Urenna Onyewuchi, Aaron Halbert, Selcuk Uluagac, Daniel Lertpratchya, Jordan Lee, Brian Hayes, Brian Swenson, Sarwat Nasir, Elnaz Banan, Ubaid Ullah, Kulin Shah, Sedrick Dongmo Nguela, Cyrus Harvesf, Ramya Srinivasan, Apurva Mohan and his wife Himali, Mike Sun, Raj Bhattacharjea, David Bauer, Elizabeth Lynch, Sung Park, Faisal Khan, and Haejoon Jung. I appreciate all the conversations we have had and will always remember the fun times we have enjoyed.

I would also like to thank several Georgia Tech professors and staff: Beverly Scheerer, Irene Rouse, Jorge Breton, Carlos Santamarina, Mary Ann Weitnauer (formerly Mary Ann Ingram), Raghupathy Sivakumar, Gregory Durgin, Giorgio Casinovi, and Scott Wills. Thank you for all our discussions (both technical and non-technical),

for always being so nice to me, and for making me feel part of the Georgia Tech community.

Finally, I want to thank the Goizuetta Foundation and the National Science Foundation for providing me with funding throughout all these years.

It is paradoxical, yet true, to say, that the more we know, the more ignorant we become in the absolute sense, for it is only through enlightenment that we become conscious of our limitations. Precisely one of the most gratifying results of intellectual evolution is the continuous opening up of new and greater prospects.

– Nikola Tesla

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Wireless communication systems have exploded in popularity over the past few decades. Due to their popularity, the demand for higher data rates by the users, and the high cost of wireless spectrum, wireless providers are actively seeking ways to improve the spectral efficiency of their networks. One promising technique to improve spectral efficiency is to equip the wireless devices with multiple antennas. If both the transmitter and receiver of a link are equipped with multiple antennas, they form a multiple-input multiple-output (MIMO) link.

The multiple antennas at the nodes provide degrees-of-freedom that can be used for either sending multiple streams of data simultaneously (a technique known as spatial multiplexing), or for suppressing interference through linear combining, but not both. Due to this trade-off, careful allocation of how many streams each link should carry is important to ensure that each node has enough degrees-of-freedom available to suppress the interference and support its desired streams. How the streams are sent and received and how interference is suppressed is ultimately determined by the beamforming weights at the transmitters and the combining weights at the receivers. Determining these weights is, however, made difficult by their inherent interdependency.

Our focus is on unplanned and/or dense single-hop networks, such as WLANs and femtocells, where each single-hop network is composed of an access point serving several associated clients. The objective of this research is to design algorithms for maximizing the performance of dense single-hop wireless networks of MIMO links. We address the problems of determining which links to schedule together at each time slot, how many streams to allocate to each link (if any), and the beamforming

and combining weights that support those streams.

This dissertation describes four key contributions as follows:

- We classify any interference suppression technique as either unilateral interference suppression or bilateral interference suppression. We show that a simple bilateral interference suppression approach outperforms all known unilateral interference suppression approaches, even after searching for the best unilateral solution.

- We propose an algorithm based on bilateral interference suppression whose goal is to maximize the sum rate of a set of interfering MIMO links by jointly optimizing which subset of transmitters should transmit, the number of streams for each transmitter (if any), and the beamforming and combining weights that support those streams.

- We propose a framework for optimizing dense single-hop wireless networks. The framework implements techniques to address several practical issues that arise when implementing interference suppression, such as the overhead of performing channel measurements and communicating channel state information, the overhead of computing the beamforming and combining weights, and the overhead of cooperation between the access points.

- We derive the optimal scheduler that maximizes the sum rate subject to proportional fairness.

Simulations in ns-3 show that the framework, using the optimal scheduler, increases the proportionally fair aggregate goodput by up to 165% as compared to the aggregate goodput of 802.11n for the case of four interfering single-hop wireless networks with two clients each.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Research Objective

Wireless communication has come a long way since Nikola Tesla, using James Maxwell's and Heinrich Hertz's works on transmission of electromagnetic waves, demonstrated the transmission of information through a wireless medium using such waves [70]. The Second World War led to much interest in wireless communications, giving way to many of the theoretical foundations of communications. Claude Shannon's work in 1948, which provided an upper bond to the error-free data rate under a signal-to-noise ratio (SNR) constraint [86], appeared during that time. Today, wireless communications have become an integral part of our everyday lives. The high degree of mobility and freedom that wireless communication technologies provide has led to a wide adoption of devices that incorporate these technologies. Wireless local area networks (WLANs), wireless mesh networks, cell phone networks, femtocell networks, and personal area networks are just a few examples of widely used networks that incorporate wireless communication technologies. With the ever-increasing number of wireless users, the demand for higher throughput by these users, and the expensive cost of spectrum, service providers are actively seeking ways to increase the spectral efficiency by packing more bits/second/Hertz.

One promising technique for improving the spectral efficiency of a wireless system is to equip the wireless nodes with multiple antennas. A link in which both the transmitter and the receiver nodes have multiple antennas is called a multiple-input multiple-output (MIMO) link. The use of multiple antennas is attractive because, without the need of additional spectrum or power, the degrees-of-freedom provided

by the multiple antennas can be used for carrying more data within a given link (using a technique called spatial multiplexing) and/or for supporting multiple interfering links at a given time (using a technique called interference suppression). As a consequence of these capabilities, MIMO wireless systems have been integrated into cellular systems [41, 70] and wireless local area networks (WLANs) [51].

With *spatial multiplexing* only, a transmitter node with n_t antenna elements can send n_t data streams simultaneously, and, similarly, a receiver node with n_r antennas can receive up to n_r streams simultaneously. Therefore, in the absence of interference, a MIMO link can increase its capacity by a factor of $n = \min(n_r, n_t)$ when compared to the capacity of a single-input single-output (SISO) link, in which both the transmitter and receiver have one antenna element [70].

On the other hand, with *interference suppression* only, a transmitter node with n_t antenna elements can send a single data stream and suppress its interference from no more than $n_t - 1$ interfering receivers supporting a single stream each. Similarly, a receiving node with n_r antenna elements can receive a single data stream and suppress interference from no more than $n_r - 1$ interfering transmitters supporting a single stream each. By suppressing interference, participating nodes improve the overall performance since multiple interfering links can simultaneously communicate their streams reliably and independently.

When the wireless nodes collaborate to perform both spatial multiplexing and interference suppression, however, the overall performance improvement can be much larger than when only one of these techniques is used exclusively [94]. Yet, there exists a trade-off between these capabilities in that the number of spatially multiplexed streams plus the number of interfering streams suppressed cannot exceed the number of antenna elements on a node. That is, a node (transmitter or receiver) with n antennas can support d streams and suppress interference for at most i interfering streams as long as $n \geq d + i$ [7, 46, 65, 92, 94]. Therefore, each node can use the

degrees of freedom provided by its multiple antennas for spatial multiplexing or for interference suppression, but not both [46, 65, 92].

To highlight the benefit of performing a combination of spatial multiplexing and interference suppression in a network, consider the network in Figure 1.1, which consists of three interfering links where every node has four antenna elements. With spatial multiplexing only, each link can take turns transmitting four streams each, for an average of four streams at any given time. With interference suppression only, each link can transmit a single stream simultaneously, for an average of three streams. With spatial multiplexing and interference suppression, however, each link can support up to two streams simultaneously, for an average of six streams [25, 94].

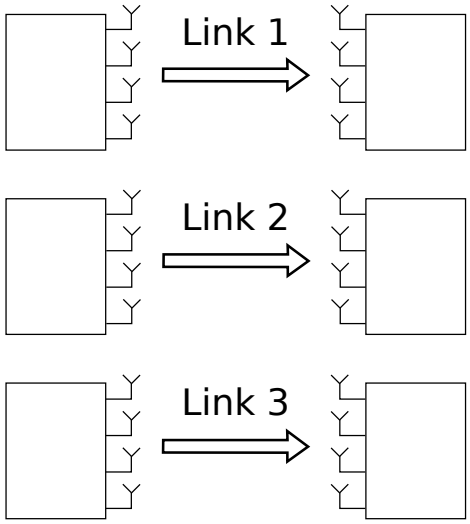

Figure 1.1: A three-link network example where each link can support two streams simultaneously when nodes perform a combination of spatial multiplexing and interference suppression.

For the network of Figure 1.1, it is known that six is the maximum number of interference free streams [14, 94]. In general networks, however, determining the number of interference-free streams that can be allocated in a network is not only an NP-complete problem [10, 83], but it does not necessarily maximize the performance of the network.

One key concept is that the energy (signal or interference) between a transmitter-receiver pair is a function of the beamforming weights at the transmitter, the corresponding channel between the transmitter and the receiver, and the combining weights at the receiver. The *beamforming weights* are used by the transmitter of a link to send a linear combination of its desired signals through each of its antenna elements. Similarly, the *combining weights* are used by the receiver to create a linear combinations of the signals received at each of its antennas so as to form an estimate of the desired signals. By adjusting how the signals are beamformed and combined using these weights, nodes can suppress interference and support multiple streams on their links. To compute these weights so that nodes can suppress interference, however, requires that the channel state information (CSI) between participating nodes be measured and collected —an overhead that must be reduced to achieve the promised gains of performing both spatial multiplexing and interference suppression.

Since interference between a pair of nodes is a function of the beamforming and combining weights of those nodes, the beamforming and combining weights between every pair of nodes that interferes are interdependent. This not only not only complicates their computation, but also complicates tasks performed at the upper layers since the performance of a set of links is unknown until the beamforming and combining weights are computed. One such task that is of interest in this dissertation is that of scheduling to maximize performance and achieve fairness.

Our focus is on unplanned and/or dense single-hop networks, such as WLANs and femtocells, where each single-hop network is composed of an access point serving several associated clients. The objective of this research is to design algorithms for maximizing the performance of dense single-hop wireless networks of MIMO links. We address the problems of determining which links to schedule together at each time slot, how many streams to allocate to each link (if any), and the beamforming and combining weights that support those streams.

1.2 Dissertation Contributions and Outline

We begin by reviewing wireless communication and MIMO concepts in Chapter 2. Our contributions are then presented in chapters 3 through 6.

- In Chapter 3, we classify any interference suppression technique as either unilateral interference suppression or bilateral interference suppression. We show that a simple bilateral interference suppression approach outperforms all known unilateral interference suppression approaches even after searching for the best unilateral solution.

- In Chapter 4, we design an algorithm based on bilateral interference suppression whose goal is to maximize the sum rate of a set of interfering MIMO links by jointly optimizing which subset of transmitters should transmit, the number of streams for each transmitter (if any), and the beamforming and combining weights that support those streams.

- In Chapter 5, we propose a framework for optimizing dense single-hop wireless networks. The framework implements techniques to address several practical issues that arise when implementing interference suppression in dense wireless networks, such as the overhead of performing channel measurements and communicating channel state information, the overhead of computing the beamforming and combining weights, and the overhead of cooperation between the access points.

- In Chapter 6, we derive the optimal scheduler that maximizes the sum rate subject to proportional fairness.

Finally, in Chapter 7, we provide our conclusions, suggestions for future work, and list of publications resulting from this dissertation.

1.3 Notation and Nomenclature

$\boldsymbol{A}$	Matrix
$(\boldsymbol{A})_{*k}$	The k^{th} column of matrix $\boldsymbol{A}$
$(\boldsymbol{A})_{kl}$, a_{kl}	Element at the k^{th} row and l^{th} column of matrix $\boldsymbol{A}$
$\boldsymbol{a}$	Column vector
a_k	The k^{th} element of vector $\boldsymbol{a}$
a, A	Scalar
$(\cdot)^T$	Transpose of $(\cdot)$
$(\cdot)^\dagger$	Conjugate transpose of $(\cdot)$
$\boldsymbol{A}^{-1}$	Inverse of matrix $\boldsymbol{A}$
$\mathrm{tr}\,(\boldsymbol{A})$	Trace of matrix $\boldsymbol{A}$
$\mathrm{rank}\,(\boldsymbol{A})$	Rank of matrix $\boldsymbol{A}$
$\boldsymbol{A} \succeq 0$	Matrix $\boldsymbol{A}$ is positive semidefinite.
$\mathrm{diag}\,(\boldsymbol{A})$	Column vector of the main diagonal element of matrix $\boldsymbol{A}$
$\mathrm{diag}\,(\boldsymbol{a})$	Square diagonal matrix with the elements of vector $\boldsymbol{a}$ in its main diagonal
$\|\boldsymbol{A}\|$	Frobenius norm of $\boldsymbol{A}$
$\|\boldsymbol{A}\|$	Determinant of matrix $\boldsymbol{A}$
$\|a\|$	Absolute value of scalar a
$\mathbb{E}[(\cdot)]$	Expected value of $(\cdot)$
$\Re\,(a)$	Real part of scalar a
$\Im\,(a)$	Imaginary part of scalar a
$\boldsymbol{I}$	Identity matrix
$\mathbf{0}$	Zero matrix or vector
$\mathfrak{i}$	$\sqrt{-1}$

CHAPTER 2

BACKGROUND

The use of multiple antennas for increasing capacity dates back to Winters in 1987 [106]. With multiple antennas, a node can take advantage of multipath propagation, which was traditionally considered an undesired channel phenomena, to improve its performance. In this chapter, we will review basic wireless communication and MIMO concepts, including multipath propagation and how MIMO systems benefit from this property. We will also provide background information on the problem of optimizing single-hop wireless networks and discuss related works.

This chapter is organized as follows. In Section 2.1, we discuss several wireless channel impairments. In Section 2.2, we consider a single MIMO link and describe how we model this system and several techniques for optimizing its performance. In Section 2.3, we focus on a set of interfering MIMO links and review difficulties and techniques associated with optimizing these systems. In Section 2.4, we discuss the problem of network-wide optimization, we state our assumptions, and we discuss related works pertaining to this problem. Finally, in Section 2.5, we provide a summary of this chapter.

2.1 Wireless Channel Impairments

The performance of any communication system is governed by the communication channel. In wired communication systems, the channel is static and predictable. In wireless communication systems, the channel is dependent on the environment, which can be dynamic and unpredictable, especially if movements are involved. In general, the wireless channel impairments that cause this dynamic and unpredictable behavior can be broadly classified into either large-scale fading or small-scale fading. In the

following sections, we will explore these to classifications in greater detail.

2.1.1 Large-Scale Fading

Whenever a radio wave is transmitted, its power density decreases as it propagates through the medium. In free-space satellite communications, the transmitted and received power are related through the well-known Friis path loss model [38]

$$P_r(d) = P_t G_t G_r \left(\frac{\lambda_c}{4\pi d}\right)^2, \tag{2.1}$$

where $P_r(d)$ is the received power at distance d, P_t is the transmitted power, G_t is the gain of the transmit antenna, G_r is the gain of the receive antenna, and λ_c is the wavelength of the radio wave. In (2.1), it is clear that the received power decays with d^2.

In terrestrial scenarios, the received power decays with d^α, where $\alpha \geq 2$ is the path loss exponent. A more generalized path loss model is given by the log-distance path loss model

$$P_{r\ (\mathrm{dB})}(d) = \mu_{\ (\mathrm{dB})}(d_0) + 10\alpha \log_{10}\left(\frac{d_0}{d}\right) \tag{2.2}$$

where $P_{r\ (\mathrm{dB})}(d)$ is the received power in dB at a distance d, and $\mu_{(\mathrm{dB})}(d_0)$ is the average received power in dB at a known reference distance d_0. Typically, d_0 is 1 km for outdoor macrocells, 100 m for microcells, and 1 m for indoor picocells [96]. The path loss exponent can vary from 2 to 6 depending on the propagation environment [18]. Table 2.1 lists empirical values for the path-loss exponent for different terrains as reported by [63].

Using (2.2) we can compute the signal-to-noise ratio (SNR) as

$$\mathrm{SNR}_{(\mathrm{dB})}(d) = \mathrm{SNR}_{\mathrm{ref}\ (\mathrm{dB})}(d_0) + 10\alpha \log_{10}\left(\frac{d_0}{d}\right), \tag{2.3}$$

where $\mathrm{SNR}_{\mathrm{ref}\ (\mathrm{dB})}(d_0) = \mu_{\ (\mathrm{dB})}(d_0) - n_{(\mathrm{dB})}$ is the average reference signal-to-noise ratio (SNR) in dB at a known distance d_0, $n_{(\mathrm{dB})}$ is the noise power in dB, and $\mathrm{SNR}_{(\mathrm{dB})}(d)$ is the SNR in dB at distance d.

Table 2.1: Empirical path loss exponents for several terrains [63].

Terrains	Path Loss Exponent (α)
Free Space	2
Open Area	4.35
North American Suburban	3.84
North American Urban (Philadelphia)	3.68
North American Urban (Newark)	4.31
Japanese Urban (Tokyo)	3.05

The model on (2.2) does not capture the fact that the received signal power of two equidistant paths may be different due to shadowing caused by obstacles on the path. *Shadowing* occurs when large obstacles such as a hill or building obscures the main signal path. To account for shadowing, an error ϵ_{dB} is introduced so that (2.2) becomes

$$P_{r\ (\text{dB})}(d) = \mu_{\ (\text{dB})}(d_0) + 10\alpha \log_{10}\left(\frac{d_0}{d}\right) + \epsilon_{\text{dB}}. \tag{2.4}$$

In (2.4), ϵ_{dB} is modelled as a zero-mean Gaussian random variable with standard deviation $\sigma_{\epsilon\ (\text{dB})}$. Values of α and $\sigma_{\epsilon\ (\text{dB})}$ for various indoor scenarios can be found in [2]. The received signal power in the presence of shadowing as defined by (2.4) is called the local mean, and the received signal power in the absence of shadowing as defined by (2.2) is called the area mean.

The log-distance path loss model presented in this section is one of the simplest and most general models for the received signal power. Other specialised models include Okumura-Hata's [74] and Lee's [63] models for macrocells , and COST231-Hata's model for microcells [29].

2.1.2 Small-Scale Fading

In wireless systems, radio waves sent by a transmitter can encounter objects along its path, causing the signal to reflect, diffract, and/or scatter before arriving at the receiver. Definitions of these physical phenomena are as follows:

- Reflection – when the radio wave changes direction due to an impact against an object that has much larger dimensions than its wavelength.
- Diffraction – when the radio wave bends or spreads out due to an impact against an object with sharp irregularities or small openings that are comparable to its wavelength.
- Scattering – when the radio wave changes direction due to impacts against one or more objects with small dimensions as compared to its wavelength.

Figure 2.1 depicts a representation of these physical phenomena.

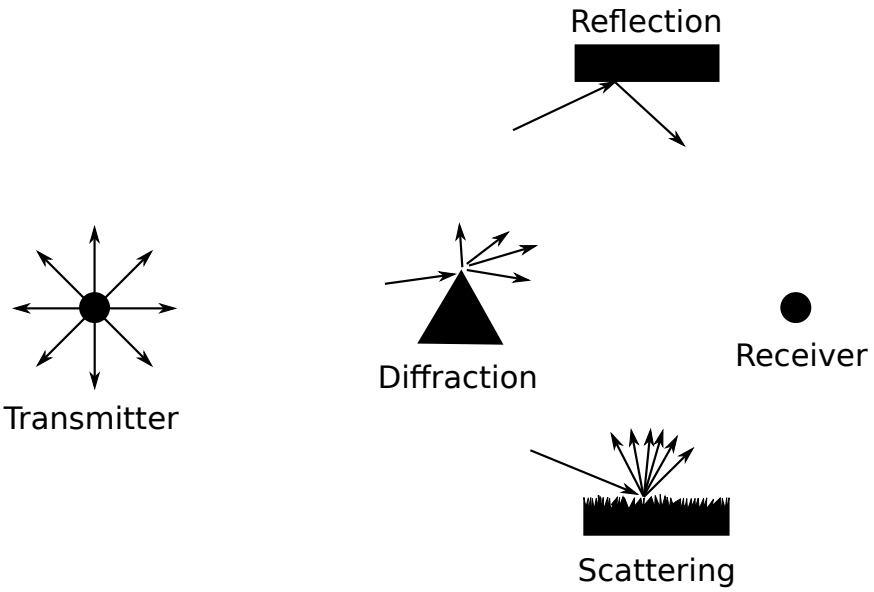

Figure 2.1: A visual representation of reflection, diffraction, and scattering. The receiver sees a composite of the multiple copies of the transmitted signal.

As a consequence of these physical phenomena, multiple copies of the transmitted radio wave can arrive at the receiver from different paths and with different delays, a property called *multipath propagation*. Here, the multipath radio waves combine vectorially at the receiver to produce a composite signal. Each multipath component, however, can arrive with a different phase because each multipath component can experience a different delay. Therefore, some multipath components will add constructively (if the phases are equal) and others will add destructively (if the phases

are different) —a property referred to as *multipath fading*.

When the received signal is composed of a large number of multipath components arriving a slightly different delays and with no direct line-of-sight component, the received signal can be represented as a complex Gaussian random variable, as given by $Z_1 + \mathfrak{i}Z_2$. Here, Z_1 and Z_2 are independent and identically distributed (i.i.d.) Gaussian random variables with a zero mean and variance σ^2 and $\mathfrak{i} = \sqrt{-1}$. This rich scattering scenario with no direct line-of-sight component is known as *Rayleigh fading* because the amplitude of the received signal $X = \sqrt{Z_1^2 + Z_2^2}$ is a Rayleigh random variable with parameter σ.

If the received signal contains a direct line-of-sight component in addition to the large number of multipath components, the received signal can be represented by $\nu + Z_1 + jZ_2$, where $\nu \in \mathbb{C}$ is the line-of-sight component, and Z_1 and Z_2 are Gaussian random variables with zero mean and variance σ^2 as in the Rayleigh fading scenario. This rich scattering scenario with a direct line-of-sight component is know as *Rician fading* because the amplitude of the received signal

$$X = \sqrt{\left(\mathfrak{R}(\nu) + Z_1\right)^2 + \left(\mathfrak{I}(\nu) + Z_2\right)^2} \tag{2.5}$$

is a Rician random variable with parameters ν and σ. In (2.5), $\mathfrak{R}(\cdot)$ and $\mathfrak{I}(\cdot)$ denote the real and imaginary components of $(\cdot)$, respectively,

Often, it is useful to characterize the multiple components using a *power delay profile* [18]. A power delay profile specifies the average power and relative delays of the multipath components. Figure 2.2 shows the power delay profile for a channel realisation representing a 2.45 GHz indoor channel as defined by the IEEE 802.11b Task Group [17]. Each stem element in Figure 2.2 is called a tap and represents the average power of a copy of the transmitted signal. For convenience, the corresponding values of each tap are listed in Table 2.2.

Figure 2.3 shows the corresponding frequency response of the channel realization of Table 2.2. Notice that the multipath components cause the gain of the channel

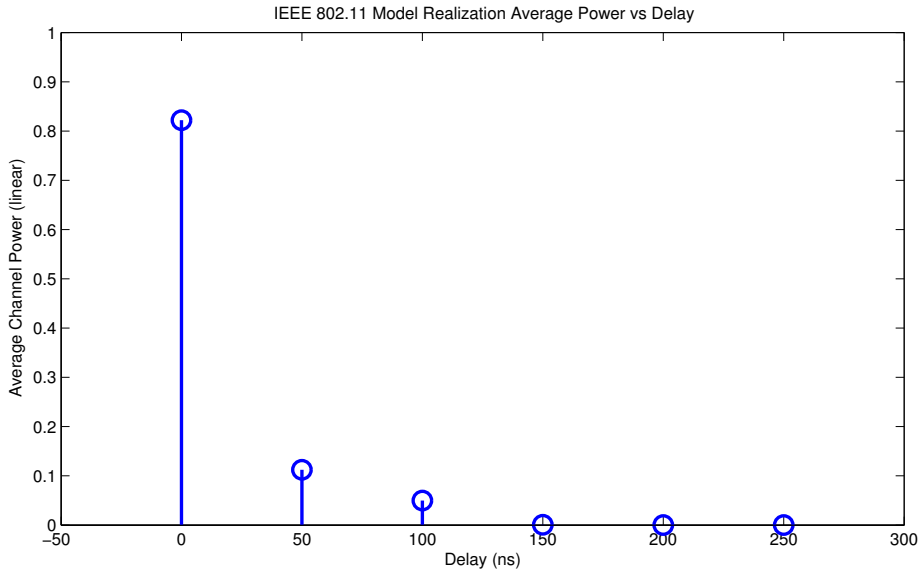

Figure 2.2: Power delay profile for a realization of an indoor channel at 2.45 GHz for 802.11b.

Table 2.2: Power delay profile for a realization of an indoor channel at 2.45 GHz for 802.11b.

Tap	Relative Delay (ns)	Average Power (dB)
1	0	-0.8516
2	50	-9.4898
3	100	-13.0348
4	150	-32.1285
5	200	-42.0895
6	250	-44.2322

to vary over frequency. If the pulses of the transmitted signal have a very short duration, the transmitted signal will have a wide bandwidth, and the received signal will experience *frequency-selective fading* since different frequency components of the transmitted signal will experiencing different gains. If, on the other hand, the pulses of the transmitted signal have a very long duration, the transmitted signal will require a narrow bandwidth, and the received signal will experience *frequency-none-selective fading* or *flat fading* since the gain of the channel is relatively constant over small changes in frequency. To overcome the frequency-selective nature of wideband

signals, practical systems adopt orthogonal frequency-division multiplexing (OFDM) modulation [18, 96]. With OFDM, the bandwidth is divided into a large number of closely spaced orthogonal narrow sub-bands or sub-carriers. Since each sub-carrier is narrow, each will experience flat fading.

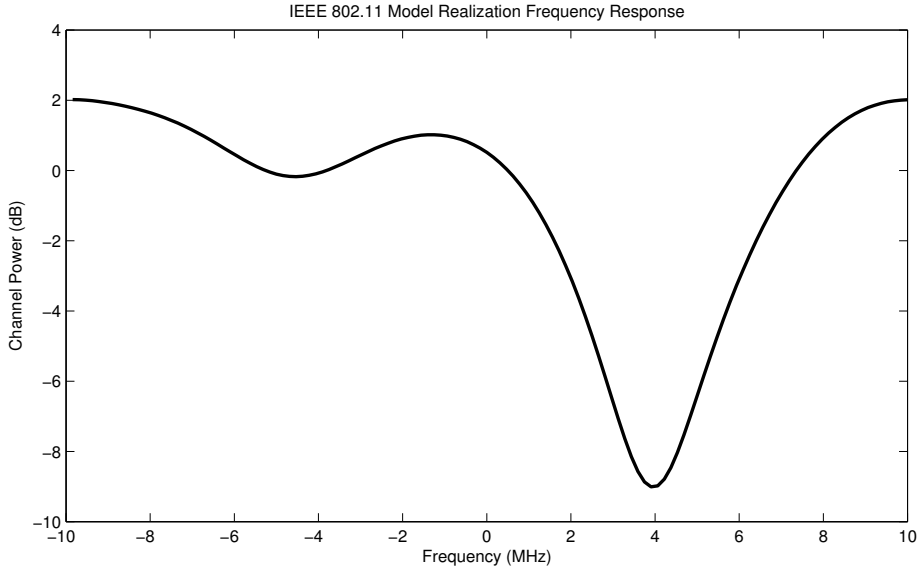

Figure 2.3: Frequency response for a realization of an indoor channel at 2.45 GHz for 802.11b.

If the transmitter, receiver, or surrounding objects are moving, the received signal will experience a frequency shift, or *Doppler shift*. The maximum Doppler shift experienced by the received signal determines the coherence time. The *coherence time* is a measure of the time it takes for the channel to become uncorrelated from its previous value, and is inversely proportional to the maximum Doppler shift. Slow movements by the transmitter, receiver or surrounding objects will cause the channel to have a high coherence time so that the channel changes slowly; this is called *slow fading*. If the channel change is slow enough, it can be considered static over the duration of one or more symbols. In contrast, fast movements by the transmitter, receiver or surrounding objects will cause the channel to have a low coherence time so that the channel changes rapidly; this is called *fast fading*. Figure 2.4 shows the magnitude of the envelope of a received signal as a function of time as modeled using

Jakes' fading simulator [53] with a maximum Doppler shift of 11.43 Hz (corresponding to a velocity of 1.4 meters/sec for a 2.45 GHz carrier frequency).

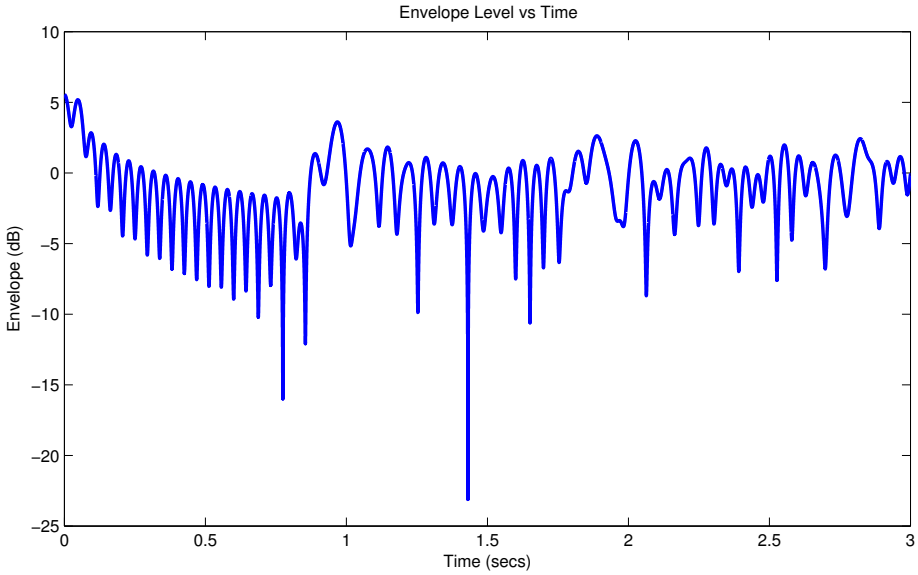

Figure 2.4: Faded envelope generated by Jakes' fading simulator with 16 low-frequency oscillators, a maximum Doppler shift of 11.43 Hz.

2.2 A Single-Link MIMO

As mentioned previously, multipath propagation was traditionally considered an undesired effect. In MIMO systems, however, multipath propagation is exploited to improve the spectral efficiency.

In this section, we will review MIMO basics, including how MIMO systems benefit from multipath propagation. Our focus will be on single-link MIMO systems. Later, in Section 2.3, we will broaden our focus to consider multiple interfering MIMO links and discuss techniques used to optimize the performance of these networks.

2.2.1 System Model

Consider a single MIMO link with n_t antennas at the transmitter (t) and n_r antennas at the receiver (r), as depicted in Figure 2.5. Here, the n_t transmit antennas serve as the multiple inputs to the channel, while the n_r receive antennas serve as the

multiple outputs from the channel. Hence, the system is referred to as a multiple-input multiple-output system.

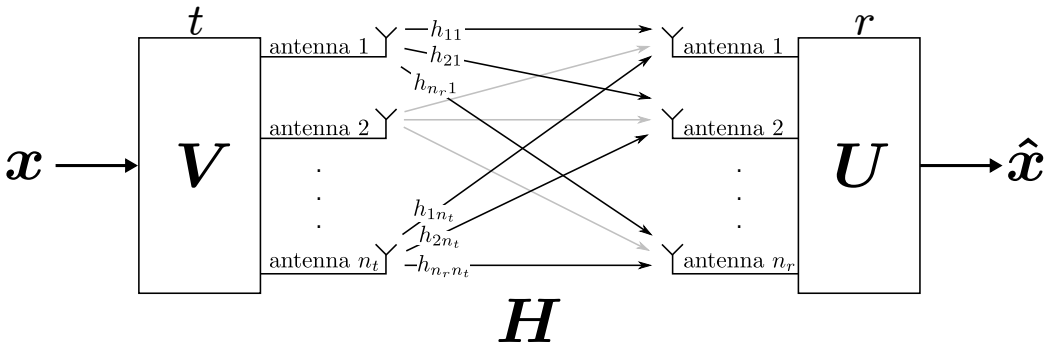

Figure 2.5: System block diagram for a single MIMO link.

We will assume a narrowband channel and no mobility so that the channel causes frequency-non-selective fading and behaves in a quasi-static fashion, in which the channel is assumed constant for the duration of a burst, but random between bursts. This assumption represents an indoor channel where the coherence time is large compared to the time of a burst of data [37].

The channel, denoted as $\boldsymbol{H}$ in Figure 2.5, is an $n_r \times n_t$ matrix with complex elements given by

$$\boldsymbol{H} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1N_t} \\ h_{21} & h_{22} & \cdots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{N_r2} & \cdots & h_{N_rN_t} \end{pmatrix}, \tag{2.6}$$

where each entry $h_{ij} \in \mathbb{C}$ denotes the attenuation and phase shift between the j^{th} transmit antenna and the i^{th} receive antenna. The channel essentially "mixes up" the signals from each transmit antenna element so that each receive antenna sees a linear combination of them. If the transmit and receive antenna elements are spaced at distance of at least $\lambda_c/2$, where λ_c is the wavelength of the carrier frequency, and the channel is rich in multipath components, then fading from each transmit antenna to each receive antenna will be independent [70]. We can model this channel using

the flat-fading Rayleigh model, so that the MIMO channel element at the i^{th} row and j^{th} column is given by [37]:

$$h_{ij} = \sqrt{\frac{\rho}{2}} \left(Z_1 + \mathfrak{i} Z_2 \right), \tag{2.7}$$

where Z_1 and Z_2 are i.i.d. Gaussian random variables with a zero mean and unit variance, and ρ is the SNR of the link. Notice that $\mathbb{E}\left[|h_{ij}|^2\right] = \rho$.

With independent fading, the channel matrix $\boldsymbol{H}$ is full rank with probability one. Because of this, the received signals will then be composed of rank($\boldsymbol{H}$) linearly independent combinations of the signals sent on each of the transmit antennas. Since rank($\boldsymbol{H}$) $\leq \min(N_r, N_t)$, the MIMO link can support up to $\min(N_r, N_t)$ streams in parallel, a capability called *spatial multiplexing*.

In the event that a line-of-sight component is present at the receiver, the MIMO channel is no longer Rayleigh faded. Instead, it is Rician faded. Discussions on how to generate Rician faded MIMO channels can be found in [41, 56, 71]. At the cost of complexity, MIMO channels can also be calculated so as to model a particular physical environment. The channel matrix in such a physical model would depend on physical parameters such as the angle of arrival (AOA), angle of departure (AOD), and time of arrival (TOA) [56]. In [71], Molisch presents a physical MIMO model and provides typical physical parameters for both macro and microcell environments. The *Kronecker Model* for MIMO channels, which provide a way to introduce correlation between the antenna elements, can provide a balance between complexity and accuracy [41, 56, 71].

If we assume that the transmitter in Figure 2.5 is sending d streams, the received signal is given by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{V}\boldsymbol{x} + \boldsymbol{n}, \tag{2.8}$$

where $\boldsymbol{x} \in \mathbb{C}^d$ is the desired signal vector to communicate; $\boldsymbol{V} \in \mathbb{C}^{n_t \times d}$ is called the beamforming matrix and is used by the transmitter to send a linear combination of

$\boldsymbol{x}$ through its antennas; and $\boldsymbol{n} \in \mathbb{C}^{n_r}$ is the noise vector. The receiver can take this signal $\boldsymbol{y}$ and apply its combining weights $\boldsymbol{U} \in \mathbb{C}^{n_r \times d}$ to form an estimate of $\boldsymbol{x}$, which we denote as $\hat{\boldsymbol{x}}$. The received signal after combining is given by

$$\hat{\boldsymbol{x}} = \boldsymbol{U}^\dagger \boldsymbol{y} = \boldsymbol{U}^\dagger \boldsymbol{H}\boldsymbol{V}\boldsymbol{x} + \boldsymbol{U}^\dagger \boldsymbol{n}, \tag{2.9}$$

where $(\cdot)^\dagger$ is the conjugate transpose of $(\cdot)$.

In the following sections, we will describe multiple methods for obtaining the beamforming weights $\boldsymbol{V}$ and the combining weights $\boldsymbol{U}$. We will assume that $\boldsymbol{x}$ is composed of independently encoded Gaussian codebook symbols with unit-energy so that $\mathbb{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix of appropriate size. Also, we will assume that the elements of $\boldsymbol{n}$ are circularly symmetric additive white Gaussian noise so that $\mathbb{E}[\boldsymbol{n}\boldsymbol{n}^\dagger] = \boldsymbol{I}$. Finally, we assume that the transmitter has a limited transmit power p and so the beamforming weights for the transmitter must satisfy $\operatorname{tr}\left(\boldsymbol{V}\boldsymbol{V}^\dagger\right) \leq p$.

2.2.2 Zero-Forcing (ZF)

One of the simplest techniques for computing both the beamforming and combining weights is to use zero-forcing (ZF), which simply aims to invert the channel [6]. If the beamforming weights $\boldsymbol{V}$ are given, then the ZF combining weights are given by

$$\boldsymbol{U}^{\mathrm{ZF}} = \boldsymbol{H}\boldsymbol{V}\left(\boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{H}\boldsymbol{V}\right)^{-1}. \tag{2.10}$$

Notice that the combining weights in (2.10) are the pseudo-inverse of the cascade of the channel matrix $\boldsymbol{H}$ and the beamforming matrix $\boldsymbol{V}$. Using (2.10), the received signal after combining becomes

$$\begin{aligned}
\hat{\boldsymbol{x}} &= \left(\boldsymbol{U}^{\mathrm{ZF}}\right)^\dagger \boldsymbol{y} \\
&= \left(\boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{H}\boldsymbol{V}\right)^{-1} \boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{H}\boldsymbol{V}\boldsymbol{x} + \left(\boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{H}\boldsymbol{V}\right)^{-1} \boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{n} \\
&= \boldsymbol{x} + \left(\boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{H}\boldsymbol{V}\right)^{-1} \boldsymbol{V}^\dagger \boldsymbol{H}^\dagger \boldsymbol{n}.
\end{aligned} \tag{2.11}$$

Although simple, the ZF combining weights can perform poorly because it inverts the matrix, forcing the interference between the streams to zero, even if noise is amplified. This is obvious in the second term of (2.11), where $\left(\boldsymbol{U}^{\mathrm{ZF}}\right)^{\dagger}$ might have some amplifying effect on $\boldsymbol{n}$.

If we assume that the combining weights are given, we can also compute a set of ZF beamforming weights as follows:

$$\boldsymbol{V}^{\mathrm{ZF}} = \boldsymbol{H}^{\dagger}\boldsymbol{U}\left(\boldsymbol{U}^{\dagger}\boldsymbol{H}\boldsymbol{H}^{\dagger}\boldsymbol{U}\right)^{-1}\boldsymbol{\Theta}, \tag{2.12}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{d\times d}$ is a diagonal matrix that ensures that the ZF beamforming weights satisfy the transmit power constraint $\mathrm{tr}\left(\boldsymbol{V}^{\mathrm{ZF}}\left(\boldsymbol{V}^{\mathrm{ZF}}\right)^{\dagger}\right) \leq p$.

2.2.3 Minimum Mean-Squared-Error (MMSE)

The minimum mean-squared-error (MMSE) criteria relaxes the channel inverse constraint so that the receiver can collect more signal energy [6]. The MMSE objective function can be formulated as

$$\begin{aligned}(\boldsymbol{V}^{\mathrm{MMSE}}, \boldsymbol{U}^{\mathrm{MMSE}}) = \arg\min_{\boldsymbol{V},\boldsymbol{U}} \mathrm{tr}\left(\mathbb{E}[(\hat{\boldsymbol{x}}-\boldsymbol{x})(\hat{\boldsymbol{x}}-\boldsymbol{x})^{\dagger}]\right), \\ \text{such that } \mathrm{tr}\left(\boldsymbol{V}\boldsymbol{V}^{\dagger}\right) \leq p.\end{aligned} \tag{2.13}$$

If the beamforming weights are assumed fixed, we can optimized with respect to the combining weights to find $\boldsymbol{U}^{\mathrm{MMSE}}$. To perform this optimization, we can either complete the square [6], or use matrix differential calculus (see [67, 79]) to compute the gradient of $\mathrm{tr}\left(\mathbb{E}[(\hat{\boldsymbol{x}}-\boldsymbol{x})(\hat{\boldsymbol{x}}-\boldsymbol{x})^{\dagger}]\right)$ with respect to $\boldsymbol{U}^{\dagger}$, set it equal to $\boldsymbol{0}$, and solve for $\boldsymbol{U}$. Performing the latter, we get

$$\boldsymbol{H}\boldsymbol{V} = \boldsymbol{H}\boldsymbol{V}\boldsymbol{V}^{\dagger}\boldsymbol{H}\boldsymbol{U} + \boldsymbol{U}. \tag{2.14}$$

To get (2.14), we have used the fact that $\nabla_{\boldsymbol{X}}\,\mathrm{tr}\left(\boldsymbol{X}\boldsymbol{A}\right) = \boldsymbol{A}^{T}$, where $(\cdot)^{T}$ is the transpose of $(\cdot)$. Solving for $\boldsymbol{U}$, we get the well known MMSE combining solution, as

given by [6]

$$\boldsymbol{U}^{\mathrm{MMSE}} = \left(\boldsymbol{H}\boldsymbol{V}\boldsymbol{V}^{\dagger}\boldsymbol{H}^{\dagger} + \boldsymbol{I}\right)^{-1}\boldsymbol{H}\boldsymbol{V}. \tag{2.15}$$

Similarly, we can assume that the combining weights are fixed, and optimize (2.13) with respect to the MMSE beamforming weights. The optimization, however, requires some mathematical manipulation to satisfy the transmit power constraint $\mathrm{tr}\left(\boldsymbol{V}\boldsymbol{V}^{\dagger}\right) \leq p$. The solution can be found in [55].

2.2.4 Singular Value Decomposition (SVD)

For a single MIMO link, we can compute the beamforming and combining weights that maximize the rate on the link. The solution is related to the singular value decomposition (SVD) of the channel [41, 56, 70, 101], as given by

$$\boldsymbol{A}\boldsymbol{S}\boldsymbol{B}^{\dagger} = \boldsymbol{H}, \tag{2.16}$$

where $\boldsymbol{A} \in \mathbb{C}^{n_r \times n_r}$ and $\boldsymbol{B} \in \mathbb{C}^{n_t \times n_t}$ are unitary matrices and $\boldsymbol{S} \in \mathbb{R}^{n_r \times n_t}$ is a diagonal matrix containing the singular values of $\boldsymbol{H}$. The optimal beamformer and combiner chooses $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{\Theta}$ and $\boldsymbol{U} = \boldsymbol{A}$, respectively, where $\boldsymbol{\Theta} \in \mathbb{R}^{n_t \times n_t}$ is a diagonal matrix that distributes the available power across the streams. The optimal power allocation strategy is achieved by allocating the maximum power using a strategy originally suggested by Shannon [87], which is commonly known as waterfilling. This waterfilling technique distributes the power across the streams in such a way as to allocate more power to those streams with the highest gains. Let $\mathrm{diag}\left(\boldsymbol{\Theta}\right) = [\theta_1, \theta_2, \ldots, \theta_{n_t}]^T$. Then, assuming that $\mathrm{diag}\left(\boldsymbol{S}\right) = [\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_N}]^T$ are sorted such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, the values of $\mathrm{diag}\left(\boldsymbol{\Theta}\right)$ can be computed using the procedure in Figure 2.6.

Interestingly, if beamforming weights are given by $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{\Theta}$, the ZF combiner in (2.10) and the MMSE combiner in (2.15) become

$$\boldsymbol{U}^{\mathrm{ZF}} = \boldsymbol{A}\boldsymbol{\mathcal{D}}_1, \tag{2.17}$$

$$\boldsymbol{U}^{\mathrm{MMSE}} = \boldsymbol{A}\boldsymbol{\mathcal{D}}_2, \tag{2.18}$$

1: Initialize $k = N$;
2: **repeat**
3: Compute waterfilling level $\mu = (p + \sum_{j=1}^{k} \frac{1}{\lambda_j})/k$;
4: Compute the power level $\theta_j = \mu - \frac{1}{\lambda_j}$ for all $j \in \{1, \ldots, k\}$;
5: Set $\theta_j = 0$ for all $j \in \{k+1, \ldots, n_t\}$;
6: $k = k - 1$;
7: **until** $\theta_j \geq 0$ for all $j \in \{1, \ldots, n_t\}$
8: **return** $\boldsymbol{\Theta}$;

Figure 2.6: Waterfilling pseudocode.

respectively, where $\boldsymbol{\mathcal{D}}_1 = \boldsymbol{S}^{-1}\boldsymbol{\Theta}^{-1}$ and $\boldsymbol{\mathcal{D}}_2 = \left(\boldsymbol{S}^2\boldsymbol{\Theta}^2 + \boldsymbol{I}\right)^{-1}\boldsymbol{S}\boldsymbol{\Theta}$ are diagonal, real matrices. Since multiplying (2.9) by a diagonal matrix has no effect on its performance (both signal and noise are scaled by the same value), the ZF and MMSE combiners are optimal for the single link case, whenever the beamforming weights are given by the optimal $\boldsymbol{V} = \boldsymbol{B}\boldsymbol{\Theta}$.

Besides the optimization criteria for the beamforming and combining weights described in Section 2.2.2 to Section 2.2.4, another criterion is to compute the beamforming and combining weights that minimizes the weighted mean-squared-error. In [84], the authors present the structure of the joint beamforming and combining weights that minimize the weighted sum mean-squared-error (MSE) for the case of a single link in the absence of interference. The authors show that minimizing the weighted sum MSE can also maximize the sum rate. Additionally, the authors use the weighted mean-squared-error solution to design beamforming and combining weights that achieve any set of relative SNRs across the streams, thereby providing quality-of-service (QoS). We will explore the weighted mean-squared-error criterion in Chapter 4, where we derive the beamforming and combining weights based on this criterion for a MIMO link in the presense of interference.

2.3 The MIMO Interference Channel

In this section, we extend our discussion to the MIMO interference channel in which as set of MIMO links are interfering. Thanks to the degrees-of-freedom provided by the multiple antennas, MIMO link systems can manage interference so that the receivers of multiple interfering MIMO links can receive their desired signal reliably and independently. In this section, we will explore techniques for performing interference suppression. We will also discuss how performance of a network of MIMO links can be greatly improved by using a combination of spatial multiplexing and interference suppression.

This section is organized as follows. In Section 2.3.1, we will provide the system model for a link in the presence of other interfering MIMO links. In Section 2.3.2, we discuss the capacity of this system. Then, in Section 2.3.3 we will discuss some techniques that can be used to compute the beamforming and combining weights that suppress interference an support the streams on the links. Finally, in Section 2.3.4, we explain the concept of MIMO degrees-of-freedom and discuss how counting degrees-of-freedom is used in the networking literature to predict the performance of a network composed of MIMO links.

2.3.1 System Model

Consider a set of M interfering half-duplex MIMO links as depicted on Figure 2.7. Let d_k be the number of multiplexed streams on link k, and let n_{t_k} and n_{r_k} be the number of antenna elements at the transmitter and receiver of link k, respectively. Let $\boldsymbol{H}_{kl} \in \mathbb{C}^{n_{r_k} \times n_{t_l}}$ be the matrix of complex channel gains between the antennas of transmitter l and those of receiver k.

The received vector at receiver k is given by

$$\boldsymbol{y}_k = \boldsymbol{H}_{kk}\boldsymbol{V}_k\boldsymbol{x}_k + \underbrace{\sum_{l=1,l\neq k}^{M} \boldsymbol{H}_{kl}\boldsymbol{V}_l\boldsymbol{x}_l + \boldsymbol{n}_k}_{\boldsymbol{z}_k}, \tag{2.19}$$

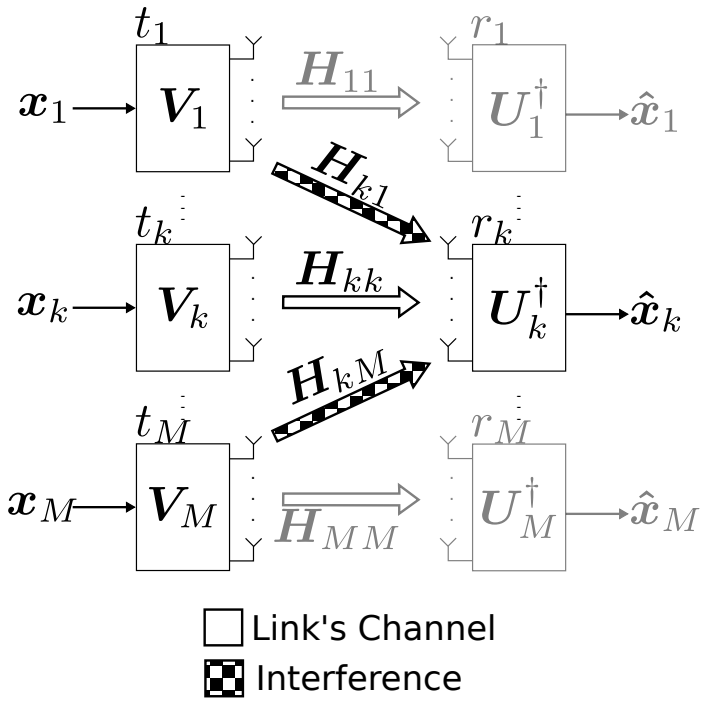

Figure 2.7: System model diagram for a network of interfering MIMO links.

where $\boldsymbol{V}_k \in \mathbb{C}^{n_{t_k} \times d_k}$ is the beamforming matrix of transmitter k; $\boldsymbol{x}_k \in \mathbb{C}^{d_k}$ is the transmit signal vector from transmitter k, assumed to be independently encoded Gaussian codebook symbols with unit-energy so that $\mathbb{E}[\boldsymbol{x}_k \boldsymbol{x}_k^\dagger] = \boldsymbol{I}$; $\boldsymbol{n}_k \in \mathbb{C}^{n_{r_k}}$ is a vector of Gaussian noise elements with covariance matrix $\mathbb{E}[\boldsymbol{n}_k \boldsymbol{n}_k^\dagger] = \boldsymbol{R}_{\boldsymbol{n}_k}$; and $\boldsymbol{z}_k \in \mathbb{C}^{n_{r_k}}$ is the total received interference plus noise with covariance matrix

$$\boldsymbol{R}_{\tilde{k}} = \mathbb{E}[\boldsymbol{z}_k \boldsymbol{z}_k^\dagger] = \sum_{l=1, l \neq k}^{M} \boldsymbol{H}_{kl} \boldsymbol{V}_l \boldsymbol{V}_l^\dagger \boldsymbol{H}_{kl}^\dagger + \boldsymbol{R}_{\boldsymbol{n}_k}. \tag{2.20}$$

In order to meet a power constraint of p_k, the beamforming weights for transmitter k must satisfy $\text{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger\right) \leq p_k$.

After combining, the received signal at receiver k is given by

$$\begin{aligned} \hat{\boldsymbol{x}}_k &= \boldsymbol{U}_k^\dagger \boldsymbol{y}_k, \\ &= \boldsymbol{U}_k^\dagger \boldsymbol{H}_{kk} \boldsymbol{V}_k \boldsymbol{x}_k + \boldsymbol{U}_k^\dagger \sum_{l=1, l \neq k}^{M} \boldsymbol{H}_{kl} \boldsymbol{V}_l \boldsymbol{x}_l + \boldsymbol{U}_k^\dagger \boldsymbol{n}_k \end{aligned} \tag{2.21}$$

where $\boldsymbol{U}_k \in \mathbb{C}^{n_{r_k} \times d_k}$ is the combining matrix of receiver k.

Notice in (2.21) that interference caused by the l^{th} link to the k^{th} link is a function of both the beamforming weights at l and the combining weights at k. This interdependency makes determining these weights a difficult problem.

Next, we briefly discuss the capacity of this system.

2.3.2 System Capacity

The instantaneous capacity of the system in (2.19) can be computed in several different ways, depending on the assumptions at hand. If we treat the residual interference after combining as interference, the instantaneous capacity in bits/sec/Hz of link k is given by as

$$\dot{C}_k = \sum_{i=1}^{d_k} \log_2 \left(1 + \mathrm{SINR}_{ki}\right), \tag{2.22}$$

where the signal-to-interference-plus-noise ratio (SINR) of stream i in link k is given by [19]

$$\mathrm{SINR}_{ki} = \frac{(\boldsymbol{U}_k)_{*i}^{\dagger} \boldsymbol{H}_{kk} (\boldsymbol{V}_k)_{*i} (\boldsymbol{V}_k)_{*i}^{\dagger} \boldsymbol{H}_{kk}^{\dagger} (\boldsymbol{U}_k)_{*i}}{(\boldsymbol{U}_k)_{*i}^{\dagger} \boldsymbol{R}_{\bar{k}} (\boldsymbol{U}_k)_{*i}}, \tag{2.23}$$

where $(\cdot)_{*i}$ is the i^{th} column of $(\cdot)$ and takes precedence over the $(\cdot)^{\dagger}$ operator so that $(\cdot)_{*i}^{\dagger} = ((\cdot)_{*i})^{\dagger}$.

If, on the other hand, we assume that the interference between the streams can be subtracted after combining, the instantaneous capacity in bits/sec/Hz of link k is given by [72]

$$\hat{C}_k = \log_2 \left| \boldsymbol{I} + \left(\boldsymbol{U}_k^{\dagger} \boldsymbol{R}_{\bar{k}} \boldsymbol{U}_k \right)^{-1} \boldsymbol{U}_k^{\dagger} \boldsymbol{H}_{kk} \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_{kk}^{\dagger} \boldsymbol{U}_k \right|. \tag{2.24}$$

Finally, if the combiner is information lossless and all inter-stream interference can be subtracted, the instantaneous capacity in bits/sec/Hz of link k is give by [11]

$$C_k = \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_{\bar{k}}^{-1} \boldsymbol{H}_{kk} \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_{kk}^{\dagger} \right|. \tag{2.25}$$

In the following section, we describe simple techniques for managing interference based on ZF and MMSE.

2.3.3 Interference Management

The problem of determining the beamforming and combining weights that suppress interference and support the streams on the link is difficult due to the inherent interdependency between the beamforming and combining weights. For this reason, most existing algorithms that compute these weights are iterative in nature [4, 20, 32–34, 43, 44, 72, 73, 78, 83, 108]. In this section, we will only discuss techniques that can be used by a receiver to choose its combining weights, assuming that the beamforming weights of all links are fixed. Later, in chapters 3 and 4, we will discuss techniques for computing both the beamforming and the combining weights for all links.

To compute the combining weights given the beamforming weights, we can use the ZF and MMSE criteria. A ZF combiner on link k, for example, will pick its column vectors as $(\boldsymbol{U}_k)_{*i} = \boldsymbol{h}_{ki} - \boldsymbol{\hbar}_{ki}$, where $\boldsymbol{h}_{ki} = \boldsymbol{H}_{kk}(\boldsymbol{V}_k)_{*i}$, and where $\boldsymbol{\hbar}_{ki}$ is the projection of $\boldsymbol{h}_{ki}$ onto the span of all other vectors $\boldsymbol{\hbar}_{kj}$ for $j \neq i$ and all column vectors of $\boldsymbol{H}_{kj}\boldsymbol{V}_j$ for all $j \neq k$. This strategy will, for each stream i at link k, eliminate all interference between the desired streams ($\boldsymbol{\hbar}_{kj}$ for $j \neq i$) and all interference from the other interfering links ($\boldsymbol{H}_{kj}\boldsymbol{V}_j$ for all $j \neq k$).

An MMSE combiner, on the other hand, relaxes the zero interference constraint and, instead, tries to capture a larger portion of the desired signal. The MMSE combining weights for link k can be obtained similarly to the single-link case as discussed in Section 2.2.3, and are given by [72, 78]:

$$\boldsymbol{U}_k^{\text{MMSE}} = \left(\boldsymbol{H}_{kk}\boldsymbol{V}_k\boldsymbol{V}_k^{\dagger}\boldsymbol{H}_{kk}^{\dagger} + \boldsymbol{R}_{\bar{k}}\right)^{-1}\boldsymbol{H}_{kk}\boldsymbol{V}_k. \tag{2.26}$$

The ZF and MMSE combiners can be used as part of a strategy for computing the beamforming and combining weights for all links. Several techniques for computing both the beamforming and combining weights include [4, 20, 21, 32–34, 43, 44, 48, 52, 58, 72, 73, 76, 78, 83, 88–91, 108]. In [43, 44], Gomadam et al. present the Max-SINR algorithm that, given a stream allocation, computes the beamforming and

combining weights that maximize the SINR of each link iteratively, assuming equal power allocation across the streams. This Max-SINR algorithm is related to MMSE since maximizing the SINR also minimizes the mean-squared-error [19]. In [78], the Peters and Heath present an algorithm based on iteratively minimizing the MSE for the MIMO interference channel. In [108], Ye and Blum present an algorithm called the Gradient Projection (GP) algorithm that optimizes the transmit covariance matrices $\boldsymbol{Q}_k = \boldsymbol{V}_k \boldsymbol{V}_k^\dagger$ to find a local optimum for the sum rate of interfering links. In [88], Shen et al. propose an algorithm that finds the transmit covariance matrices that approximately maximize the weighted sum rate using a linear approximation. In [20], Christensen et al. formulate the weighted sum MSE problem for the MIMO broadcast channel, where a single AP transmits data to several clients simultaneously. Christensen et al. relate the weighted sum MSE problem to the weighted sum rate problem and propose an algorithm whose goal is to maximize the sum rate of the MIMO broadcast channel. In [72, 73] and [90], Negro et al. and Shi et al. present similar results to [20] but for the MIMO interference channel and the MIMO interference broadcast channel, respectively. In the MIMO interference broadcast channel, multiple interfering access points transmits data to several of their associated clients simultaneously. Finally, in [83], the Razaviyayn et al. propose an algorithm based on semi-definite programming whose goal is to maximize the sum rate of the MIMO interference channel.

It is important to note that performance of both the ZF combiner and MMSE combiners will be poor if many interfering signals are present. For example, if a receiver with $n_r = 4$ antennas is trying to receive a single stream and it is interfered by $i = 4$ other streams, it is likely that the set of interfering signals will span $\mathbb{C}^4$, preventing the receiver from collecting the desired signal free of interference. Low performance, however, can be avoided by limiting the number of streams on each link so that interfered-by links are able to suppress its interference [12]. Setting how

many streams should each link support so as to achieve high performance is called the *stream allocation* problem.

In the next section, we will discuss a technique used by the networking community to estimate the number of interference-free streams that can be supported in a given network with the goal of estimating whether a given network setup will have good performance or not.

2.3.4 MIMO Degrees of Freedom and Feasibility

The preferred metrics for evaluating the performance of a network is its capacity. However, due to its decentralized nature, the capacity of interference networks remains a long-standing open problem [3, 14, 35]. Another metric that can be used to evaluate the performance of a network is to is the number of interference-free streams. In terms of capacity, these interference-free streams represent the rate of growth of network capacity with the log of the signal-to-noise ratio (SNR) [14].

A technique to determine the number of interference-free streams in the network is to count and keep track of the number of degrees-of-freedom that are used for interference suppression and spatial multiplexing at every node [7, 9, 10, 46, 65, 75, 92, 95, 99]. In general, it is said that a node with n antennas has n degrees-of-freedom. With these degrees-of-freedom, a receiver node can receive d streams and suppress interference from i interfering streams as long as $d + i \leq n$. Similarly, a transmitter node can send d streams and suppress its interference at i interfering streams as long as $d + i \leq n$. These constraints are known as the *degrees-of-freedom constraints* and are inherently based on ZF linear processing.

Whenever two nodes interfere, it is not necessary that both nodes use their degrees-of-freedom to suppress the same interference. One interference suppression strategy, which we will refer to as unilateral interference suppression in Chapter 3, assigns only one of the two nodes that interfere to use their degrees-of-freedom to perform the

interference suppression while the other node safely ignores this interference.

In large networks, it is useful to keep track of which node assumes the responsibility of suppressing interference for which other node. For this, we define a pair of matrices: one ($\boldsymbol{A}^t$) for the transmitters and one ($\boldsymbol{A}^r$) for the receivers. These matrices contain the *interference suppression assignment* that specifies, for each node, which nodes' interference must be suppressed and which nodes' interference can be safely ignored. The entry $a^t_{kl} \in \{0, 1\}$ at the k^{th} row and the l^{th} column of $\boldsymbol{A}^t$ is set to one if the transmitter of link k is assigned to suppress its interference at the receiver of link l and zero otherwise. Similarly, the entry $a^r_{kl} \in \{0, 1\}$ at the k^{th} row and the l^{th} column of $\boldsymbol{A}^r$ is set to one if the receiver of link k is assigned to suppress the interference from the transmitter of link l and zero otherwise. If interference from one node to another most be suppressed by one of the two nodes, then $a^r_{kl} = 1 - a^t_{lk}$ must hold. For convenience, we define $a^r_{kk} = a^t_{kk} = 1$ for all $k \in \{1, \ldots, M\}$.

In the context of an interference suppression strategy that assigns who suppresses who, we define a *stream allocation vector* $\boldsymbol{d} = [d_1, d_2, \ldots, d_M]$ as *feasible* if and only if there exist interference suppression assignment matrices $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$ as defined above such that the degrees-of-freedom constraints

$$\sum_{l=1}^{M} a^t_{kl} d_l \leq n_{t_k}, \tag{2.27}$$

for each transmitter node $k \in \{1, \ldots, M\}$, and

$$\sum_{l=1}^{M} a^r_{kl} d_l \leq n_{r_k}, \tag{2.28}$$

for each receiver node $k \in \{1, \ldots, M\}$, are satisfied.

As mentioned earlier, this interference suppression strategy that assigns which node suppresses which is particularly useful to determine how many interference-free streams can be active in the network. In [94], Srinivasan et al. considered the case in which all nodes in the network is equipped with n antenna elements and found that

the maximum number of interference-free streams was the largest integer not greater than $2n/(M+1)$.

As an example, consider the three link example shown in Figure 2.8, where t_k and r_k denote the transmitter and receiver of link k. For a unilateral interference suppression strategy, there are two interference suppression assignments in which two streams per link are feasible. Figure 2.8 depicts one of the two interference suppression assignments, namely

$$\boldsymbol{A}^t = \boldsymbol{A}^r = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \tag{2.29}$$

The other assignment can be obtained by transposing (2.29).

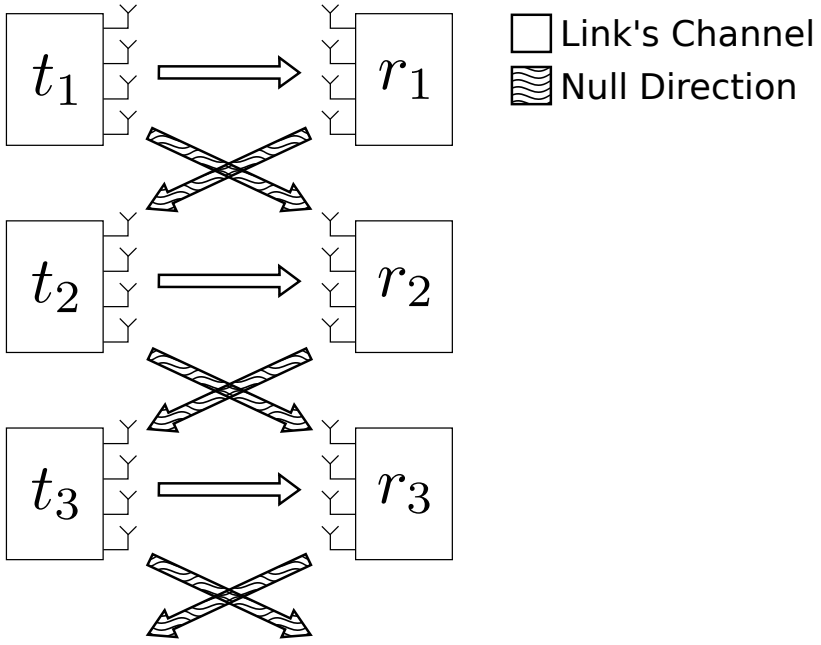

Figure 2.8: Three-link network where each node has 4 antenna elements and each link carries 2 streams. Here, t_3 suppresses r_1 and r_3 suppresses t_1, i.e. the nulls wrap around the figure.

In Figure 2.8, each node is using two degrees-of-freedom for supporting its two streams and two degrees-of-freedom for suppressing interference from one other link that is also carrying two streams. Also notice that interference between every pair

of interfering nodes is suppressed by either the transmitter of the interference or the receiver of the interference; thus, all six streams are free of interference.

The solution the three-link problem in Figure 2.8 has another interpretation based on *interference alignment*. The idea behind interference alignment is that interfering transmitters align their signals so that they overlap within a subspace of the undesired receivers' signal spaces, leaving the remaining signal subspace available for those receivers' desired signals [14]. Figure 2.9 shows a visual representation of the interference alignment solution to the three link network. In Figure 2.9, the interference from t_1 and the interference from t_3, for example, are aligned so that r_2 sees only the same wavy-red interference that occupies two dimensions. The other two dimensions are free for r_2's desired signal. For the case in which interference suppression is done using the assignment in (2.29), the same alignment of interference is achieved. For example, as shown in Figure 2.8, assigning t_1 to suppress its interference at r_2 and assigning r_2 to suppress the interference from t_3, causes the interference from t_1 and the interference from t_3 to be aligned at r_2.

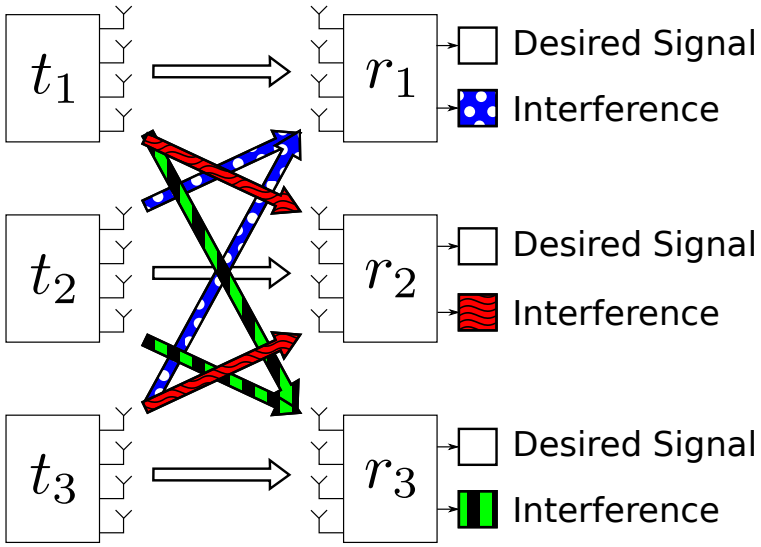

Figure 2.9: Visual representation of the interference alignment solution that maximizes the number of interference-free streams of the three-link network.

Another technique to determine if a stream allocation can be free of interference in

a given network is to compute the beamforming and combining weights and determine if the interference on each stream is below some threshold. In [109, 110], for example, Yetis et al. use the algorithms in [43, 44] to numerically determine if stream allocations can be free of interference for various network scenarios. Yetis et al. propose a rule of thumb that can almost surely determine if a stream allocation can be free of interference for the case where all links carry d streams, all transmitters have n_t antennas, and all receivers have n_r antennas. For scenarios in which the number of antennas is different and each link carries a single stream, Yetis et al. in [109, 110] also provide a method that can estimate if a stream allocation can be free of interference by counting the number of equations and the number of variables.

2.4 Single-Hop Wireless Networks

We are interested in studying the network-wide optimization of unplanned and/or dense single-hop wireless networks with MIMO links. In this scenario, each single-hop wireless network is composed of an access point (AP) and several associated clients, where the APs typically provide their clients access to a wired network, such as the internet, to provide services such as data and voice.

When the deployment of these single-hop networks is unplanned and/or dense, interference from one network can severely degrade the performance of the other networks. In the context of WLANs, single-hop networks are formed when WLANs operate in infrastructure mode and are commonly referred to as basic service sets (BSS). In current WLAN deployments, interference between these BSS is managed by contending within non-overlapping channels (three in 802.11b/g [50]). However, as the number of interfering BSS increases, the number of BSS per non-overlapping channel increases, thereby increasing interference and reducing the overall performance of the network. In cellular networks, cell towers can, in general, be carefully planned so as to minimize intercell interference. However, the emergence of femtocells, which

are typically deployed by the end user, adds an unplanned aspect in which a similar interference problem can occur. In this dissertation, we will focus our discussion of WLAN scenarios.

This section is organized as follows. In Section 2.4.1, we provide our network model. Then, in Section 2.4.2, we discuss relevant multiple access mechanisms. In Section 2.4.3, we discuss issues that arise when trying to optimize single-hop wireless networks and also state our assumptions. Finally, in Section 2.4.4, we will discuss several strategies from the literature that used for network-wide optimization of networks with MIMO links.

2.4.1 Network Model

We assume a dense deployment of overlapping single-hop networks, where each orthogonal channel is serving two to six such networks. Our discussion applies to each orthogonal channel independently. These dense deployment situations arise frequently in dense unplanned WLAN deployments, e.g. apartment buildings and commercial districts. The target scenarios could also include smaller planned deployments, e.g. within a small building.

Consider a set of A interfering single-hop wireless networks, where the k^{th} AP services C_k clients. Let M be the total number of links, where $M = \sum_{k=1}^{A} C_k$. Figure 2.10 depicts an example of four single-hop wireless networks for a WLAN scenario, where the dashed arrows between an AP and a client represent that the corresponding AP and client are associated and form a link.

Notice in the example of Figure 2.10 that links can share a node. For example, links 1, 2, and 3 share AP 1 as a common node. With MIMO links, more than one of these links that share a node can be active simultaneously. AP 1, for example, can transmit (or receive) data reliably and independently to (or from) a set of its clients. This technique known as multi-user MIMO (MU-MIMO). In this dissertation,

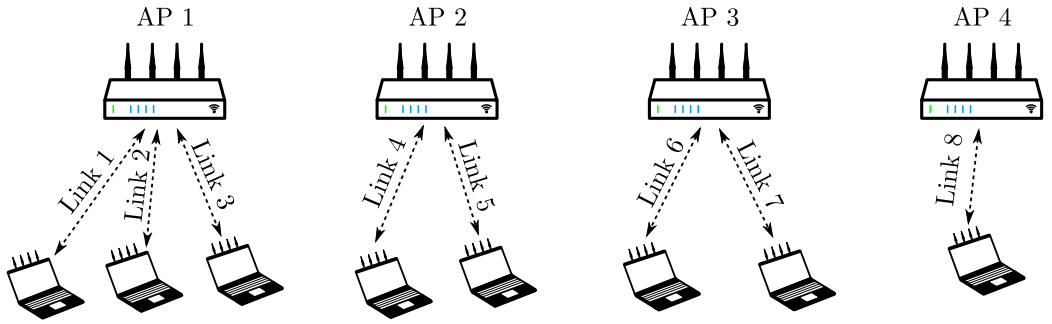

Figure 2.10: Example of four single-hop wireless networks.

however, we will not consider MU-MIMO techniques. Instead, we will assume that at most one link can be active for each network and leave the addition of MU-MIMO as a possible future research direction.

2.4.2 Multiple Access Techniques

We consider two multiple access mechanisms to share the wireless medium as follows:

- *TDMA* – With time division multiple access (TDMA), time is divided into time slots, which are generally of fixed duration. TDMA is a reservation-based multiple access scheme since links are assigned time slots in which they may communicate [40]. To avoid interference, links that interfere are assigned to separate time slots. The assignment of which links are assigned to which time slot is called a schedule. Scheduling can done in such a way as to maximize a performance metric, such as throughput and/or a fairness criteria.

- *CSMA/CA* – Carrier sense multiple access with collision avoidance (CSMA/CA) is a random multiple access scheme used by the 802.11 standard when operating in distributed coordination function (DCF) mode. With this scheme, a node wishing to transmit senses the channel first and waits until the channel becomes idle (i.e. power is below a threshold) for a random period of time. Then, the node acquires the channel by transmitting its packet. If received successfully, the destination of the packet will acknowledge the packet by transmitting an ACK.

If the source does not receive the ACK within a timeout interval, it assumes that a collision occurred and generates a new random waiting time and waits until the channel has been idle for this period of time before attempting to transmit again. The random waiting time is composed of a fixed waiting time plus a random contention window (CW). The CW is generated by an exponential backoff algorithm, which chooses the CW between a minimum (CWmin) and a maximum value (CWmax). For a given packet, the algorithm doubles CWmax (up to a maximum value) every time the transmission is unsuccessful and resets the CW to CWmin whenever the transmission is successful (see [50] for details). The exponential backoff algorithm helps resolve contention among nodes that are trying to acquire the channel.

When operating in infrastructure mode, the 802.11 standard also defines a multiple access scheme called point coordination function (PCF) that operates on top of DCF (which uses CSMA/CA). The idea of PCF is that an AP acquires the channel for a fixed period of time in which the AP can transmit to its clients and poll transmissions from its clients without contention. In this way, the AP essentially reserves the channel to communicate with its clients. We will mainly make use of the TDMA and CSMA/CA mechanisms for sharing the wireless medium in this dissertation. However, PCF will become useful later when we discuss strategies for coordinating the APs.

In the following section, we discuss several networking problems encountered while trying to benefit from the interference suppression capability of MIMO links.

2.4.3 Network Problems and Assumptions

In this dissertation, we will assume that all wireless devices operate in half-duplex mode, so that, at any given time, a node can act either as a transmitter or as a receiver, but not both. Therefore, AP 1 in Figure 2.10 will not be able to receive data

from, say, Link 1, while it is transmitting data to another of its links (say Link 2).

For our network analysis, we do not assume that the data rate is given by Shannon's capacity formulas (discussed in Section 2.3.2). Instead, we take a more practical approach and assume that a transmitter has a set of modulation and coding schemes to choose from, which will determine the data rate of a given transmission.

We obtain the probability that a packet can be decoded successfully by multiplying the probabilities that each stream of the packet is decoded successfully and independently. The probability that each stream is decoded successfully and independently is a function of the SINR of the stream, which we compute using (2.23). Note that by computing the SINR using (2.23), we are assuming that all interference, no matter how small, affects the reception of the packet.

We say that two nodes are within communication range of each other if one of the nodes, taking the role of a transmitter, can send a single stream of data at the lowest data rate and the other node can, with high probability, receive and decode this data in the absence of interference.

A problem with performing interference suppression in addition to spatial multiplexing with MIMO links is that the channel state information (CSI) between interfering nodes must be measured. If any combination of links can be chosen at a given time, complete CSI knowledge is required, which can cause high overheads depending on the size of the network. In the topology of Figure 2.10, for example, more than 250 measurements are needed to collect all CSI.

For measuring the CSI from a given node, we assume a sounding packet must be successfully received from that node. A *sounding packet* is any packet that contains training symbols in its preamble so that the receiver of the packet can estimate all channel dimensions from the source. Therefore, the only interference channels that can be measured and considered for interference suppression at a node is that which comes from interferers within its communication range. However, as mentioned

earlier, we account for all interference when measuring performance.

We assume that the majority of wireless devices are equipped with multiple antennas. Also, we assume that all clients are capable of transmitting sounding packets, and that all APs are capable of measuring CSI from a sounding packet that was successfully decoded.

We assume that the channels are symmetric so that CSI measured at a node can be used as an estimate for the reverse channel. The case of asymmetric channels can be handled by a priori executing an extra synchronization step so that the channels can be treated as symmetric [45, 57].

Additionally, we assume that channels do not change rapidly, so that beamforming and combining weights that are calculated at one time can be reused for some period of time before they must be recalculated. This assumption is consistent with the scenarios we consider, in which the APs are in fixed locations and cover environments like an office, a home, or a coffee house, where users are mobile but often stay in one location for a moderate amount of time in between movements. Detecting when channel states have changed sufficiently to necessitate a new round of measurements is beyond the scope of this dissertation. Herein, we simply assume that this is done periodically and that measurements remain valid in between these measurement times.

We have previously mentioned that most techniques for computing the beamforming and combining weights that perform interference suppression are iterative in nature. A problem that must be tackled is that of how to go about computing these weights. If the beamforming and combining weights are computed in a distributed fashion, some sort of coordination is needed so that nodes can take turns updating their weights. Additionally, depending on the complexity of the algorithm used for computing the beamforming and combining weights, the weight computation could cause high overheads if the number of packets that must be exchanged is large. In

a practical scenario, a low-overhead technique for computing the beamforming and combining weights is necessary to achieve and maintain high performance.

Another issue that arises when taking advantage of interference suppression between single-hop wireless networks is that of cooperation. Interfering APs must be able to cooperate to schedule transmissions and suppress interference. With products from companies such as Aerohive Networks, APs already cooperate to perform tasks such as channel selection and transmission power control [1]. These technologies could easily be extended to incorporate the type of cooperation proposed herein. Alternatively, many enterprise wireless configurations employ centralized network controllers, which connect to APs through a wired network and could be used to facilitate AP cooperation. This has, in fact, been done in [62], which uses a centralized controller architecture to provide a proof of concept for performing spatial multiplexing and simple interference suppression using beamforming and combining weight computed in a non-iterative fashion based on ZF (similar to the OBIC strategy discussed in Chapter 3.2.2) for the downlink only. In this work, we assume that APs cooperate either via the centralized controller or directly with each other through the wireless medium. We do not, however, allow APs to communicate with each other through clients.

When considering MIMO links that can perform both spatial multiplexing and interference suppression, scheduling to achieve high performance while maintaining fairness can be a challenging problem. The reason is because the performance of a set of links is unknown until the beamforming and combining weights are computed. Moreover, since the beamforming and combining weights for a given link set are different from those of another link set, even if the difference between the link sets is a single node, determining whether to add or remove a link to or from a time slot requires that the beamforming and combining weights be recomputed, which can be an expensive operation. These problems complicate the design of a scheduling

algorithm, specially if fairness is considered.

Finally, in this dissertation, we assume that APs are loosely synchronized. There are a variety of ways in which this can be achieved. For example, [62] had the APs run the network time protocol (NTP) for this purpose. We will discuss several techniques that can be used to achieve synchronization for our problem at hand in Chapter 5.3

In the following section, we will briefly discuss several techniques proposed for using MIMO links to improve the network-wide performance of wireless networks.

2.4.4 Related Work on Scheduling and Network-Wide Optimization

Traditional approaches to dealing with the problem of network-wide optimization tackle the problem by efficiently sharing the available bandwidth between overlapping networks, rather than to reduce or eliminate interference [30, 36, 47, 66, 69, 111, 112]. Other approaches propose to use transmission power control [5, 15, 54], directional antennas [16], or smart antennas with pre-selected beam patterns [22].

As mentioned earlier, scheduling to maintain fairness when both spatial multiplexing and interference suppression are considered can be a difficult problem since the performance of a set of links is unknown until the beamforming and combining weights are computed, which can be an expensive operation.

Several works have simplified the problem of network-wide optimization by either assuming that interference suppression is done at the receiver side only, that interference is suppressed in a sequential order, that each link supports only one stream, and/or that raw data can be shared among the receivers or the transmitters (requiring high-bandwidth wired connections). The works of [76, 98] propose a MAC protocol for ad hoc networks that exploit stream multiplexing and interference suppression at the receiver side only. In [75], Park et al. propose a MAC protocol in which transmitters and receivers of links wanting to communicate must suppress interference to/from links that are already communicating. In [61], Kim et al. propose a MAC

protocol whose goal is to maximize the throughput of multi-hop networks where each link carries at most one stream. In [42], Gollakota et al. propose to suppress some interference using the interference-suppression capability of MIMO links, then use the Ethernet to share decoded packets so that the receivers can suppress the remaining interference. In [64], Lin et al. propose a protocol for networks of MIMO links with varying number of antenna elements at every node. The goal of this protocol is to enable as many concurrent transmissions as would the transmitter with the maximum number of antenna elements. The protocol achieves this by having links join in on the transmission of other links if the new links can avoid causing and receiving interference. In [81], Rahul et al. present a strategy to coordinate multiple access points connected by a backbone in such a way as to operate as one large MIMO transmitter. In [62], Kumar et al. demonstrated limited interference-suppression capabilities using ZF on the downlink of a network of 20 nodes composed of six access points using firmware-modified off-the-shelf network interface cards and 14 unmodified clients.

Other works that aim to optimize the network-wide performance of MIMO links have tackled the problem using network optimization formulations based on satisfying the MIMO degrees-of-freedom constraints (see Section 2.3.4), but assume that the rates on the links are equal or fixed, and known a priori. In [46], Hamdaoui and Shin present a linear programming formulation that maximizes the aggregate throughput in multi-hop networks of MIMO links and that assumes that average link rates are known a priori. In [7], Bhatia and Li present an optimization formulation for multi-hop networks of MIMO links that assumes interference is suppressed at the receiver side only and that the data rates are known. The works of [65, 92, 93] formulate a linear optimization formulation for multi-hop networks of MIMO links using a link layer scheduling scheme called OBIC. The optimization formulation of [65, 92, 93] also assumes that average stream rates are known a priori. In [9], Blough et al. propose a linear programming formulation that determines which nodes form links

that maximize the aggregate capacity when scheduled together in a single time slot. The linear programming formulation of [9] accounts for streams of varying rates, but assumes that they are known a priori.

Later, in Chapter 6, we tackle the problem of scheduling and network-wide optimization by computing the beamforming and combining weights for different link sets and selecting the subset of link sets that maximize the sum rate while also meeting the desired fairness criteria.

2.5 Chapter Summary

In this chapter, we have reviewed basic wireless communication and MIMO concepts. Specifically, we have reviewed the effect of fading and how it can be used by MIMO links to improve their performance. We have also illustrated basic strategies for computing the beamforming and combining weights for a single link based on ZF and MMSE. Also, we have examined how to use the ZF and MMSE criterion to compute the combining weights of a receiver in the presence of interference from other MIMO links. We have also studied the trade-offs between spatial multiplexing and interference suppression. Additionally, we have defined the stream allocation problem. Finally, we have discussed difficulties encountered when interference suppression is considered for network-wide performance optimization and briefly discussed related works and their approaches.

CHAPTER 3

THE PERFORMANCE LOSS OF UNILATERAL INTERFERENCE SUPPRESSION

In this chapter, we will classify any strategy for determining the beamforming and combining weights in a network of interfering MIMO links as either *unilateral* or *bilateral*. We define an interference suppression strategy as *unilateral* whenever the responsibility of suppressing interference is preassigned to either the transmitter or the receiver of this interference, but not both. Many existing strategies for managing interference in a network of MIMO links adopt a unilateral approach. Examples of unilateral strategies include the SRP/SRMP-CiM [46], SPACE-MAC [75], CAS [99], OBIC [65, 92], ExtendedGreedy [95], OSTM [9], and CLOM [7].

In contrast, we define an interference suppression strategy as *bilateral* whenever the responsibility for suppressing interference is not preassigned to one of the two involved nodes, but is instead shared by both nodes in the process of determining the beamforming and combining weights. Examples of bilateral strategies include the SRMP-NiM [46], Max-SINR [44], IMMSE [52], incremental-SNR algorithm [85], and the MMSE and Max-SINR from [78].

There is a key distinction between the two categories: with unilateral interference suppression, the interference from one node to another is ignored by one of the two nodes, whereas with bilateral interference suppression, neither node ignores the interference.

In this chapter, we show that, for a three-link network, a proposed bilateral interference suppression approach performs better than all known unilateral interference

suppression approaches, even after exhaustively searching for the best unilateral solution [25]. For larger networks where an exhaustive search is infeasible, we use heuristics to search for the best possible unilateral and bilateral solution and we show that the sum capacity using bilateral interference suppression can be significantly higher than the sum capacity using unilateral interference suppression. We also illustrate that by handling the cyclic interdependencies of the beamforming and combining weights, unilateral strategies can support a higher number of streams and achieve better performance.

In this chapter, we will use the notion of interference suppression assignment matrices $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$, as defined previously in Section 2.3.4, to characterize a unilateral interference suppression strategy. We will set the entries for $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$ according to the following rules. We set $a^r_{kl} = a^t_{lk} = 0$ for $k \neq l$ if $\rho_{kl} < \varrho$ for some threshold ϱ, where ρ_{kl} is the interference-to-noise ratio (INR) caused by the transmitter of link l to the receiver of link k. For $\rho_{kl} \geq \varrho$ and $k \neq l$, however, interference is either suppressed at the transmitter or the receiver, but not both, and so $a^r_{kl} = 1 - a^t_{lk}$.

In this chapter only, we will assume that noise is white so that $\mathbb{E}[\boldsymbol{n}_k \boldsymbol{n}_k^\dagger] = \boldsymbol{R}_{\boldsymbol{n}_k} = \boldsymbol{I}$ for all $k \in \{1, \ldots, M\}$. Additionally, we are not concerned with the optimization of the power allocation through the streams, and so, in this chapter only, we will assume that each stream gets allocated equal power and that $\text{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger\right) = 1$ for all $k \in \{1, \ldots, M\}$.

This chapter is organized as follows. In Section 3.1, we describe strategies for computing the beamforming and combining weights locally for a single node for both unilateral interference suppression and bilateral interference suppression. In Section 3.2, we present an algorithm that computes the beamforming and combining weights globally for every node for both suppression strategies. In Section 3.3, we present numerical results. Finally, in Section 3.4, we provide a brief summary of this chapter.

3.1 Locally Calculating the Beamforming and Combining Weights

The computation of weights in a network is complicated by the fact that the transmitter beamforming weights and receiver combining weights are interdependent: the beamforming weights that suppress interference depend on the corresponding combining weights, while the combining weights that suppress interference depend on the corresponding beamforming weights. In Section 3.2, we propose an iterative algorithm that deals with this problem. For now, as a stepping stone, we show in this section how to compute the combining weights as a function of the relevant beamforming weights, and how to compute the beamforming weights as a function of the relevant combining weights.

For convenience we normalize the combining weights at the receiver of link k according to

$$\boldsymbol{U}_k = \hat{\boldsymbol{W}}_k \sqrt{\frac{1}{d_k}}, \tag{3.1}$$

where $\hat{\boldsymbol{W}}_k$ is the matrix formed after dividing each column vector of $\boldsymbol{W}_k$ with its corresponding Euclidean norm. In the following, we specify different ways of computing $\boldsymbol{W}_k$.

3.1.1 Zero-Forcing Combining for Unilateral Suppression

The zero-forcing (ZF) combining weights eliminate all interference, despite the penalty of reducing its signal energy [6]. In the context of a unilateral strategy, for which the suppression responsibilities are preassigned according to $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$, the ZF combining weights at the receiver of link k are

$$(\boldsymbol{W}_k)_{*i} = \boldsymbol{h}_{ki} - \boldsymbol{\hbar}_{ki}, \tag{3.2}$$

for all $i \in \{1, \cdots, d_k\}$, where $\boldsymbol{h}_{ki} = \boldsymbol{H}_{kk} (\boldsymbol{V}_k)_{*i}$, and $\boldsymbol{\hbar}_{ki}$ is the projection of $\boldsymbol{h}_{ki}$ onto the span of all columns of $\boldsymbol{h}_{kj}$ for $j \neq i$ and all columns of $a^r_{kl} \boldsymbol{H}^{kl} \boldsymbol{V}_l$ for $l \neq k$. Notice

that these ZF weights eliminate interference from the transmitter of link l only if $a^r_{kl} = 1$, and ignores it otherwise.

3.1.2 Minimum Mean-Squared-Error Combining for Unilateral and Bilateral Suppression

A minimum mean-squared-error (MMSE) receiver relaxes the zero interference constraint with the advantage of allowing more signal to be collected [6]. In the context of a unilateral strategy, for which the suppression responsibilities are preassigned according to $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$, the MMSE combining weights at the receiver of link k are

$$\boldsymbol{W}_k = \left(\boldsymbol{B}_k + \boldsymbol{I}\right)^{-1} \boldsymbol{H}_{kk}\boldsymbol{V}_k, \tag{3.3}$$

where

$$\boldsymbol{B}_k = \sum_{l=1}^{M} a^r_{kl}\boldsymbol{H}_{kl}\boldsymbol{V}_l\boldsymbol{V}_l^{\dagger}\boldsymbol{H}_{kl}^{\dagger}. \tag{3.4}$$

The presence of a^r_{kl} in (3.4) ensures that these weights suppress only the interference that they are assigned to suppress.

MMSE can also be defined in the context of bilateral interference suppression. To do so, we set $a^t_{lk} = a^r_{kl} = 1$ for $\rho_{kl} \geq \varrho$ and $k \neq l$, i.e. both the receiver of link k and the transmitter of link l include this interference in the computation of their weights. With this modification, we can use (3.3) and (3.4) to compute the MMSE combining weights at the receiver of link k in the context of bilateral interference suppression.

The MMSE weights for bilateral interference suppression are equal to the Max-SINR weights from [44] since weights that minimize the mean-squared-error also maximize the SINR [19]. We prefer to use MMSE weights instead of Max-SINR weights because MMSE weights can be computed with lower complexity than Max-SINR weights, since all weight columns of MMSE can be computed after one matrix inversion instead of computing a matrix inversion for each weight column of Max-SINR.

3.1.3 Beamforming Via a Virtual Network

To compute the beamforming weights for a transmitter that performs interference suppression, we follow [82] and reverse the channel, creating a virtual network in which we compute the beamforming weights assuming the transmitter is a virtual receiver. Specifically, to compute the beamforming weights, add a $\leftarrow$ to all variables in (3.1) to (3.4), then compute $\overleftarrow{\boldsymbol{H}}_{lk} = \boldsymbol{H}_{kl}^{\dagger}$, $\overleftarrow{\boldsymbol{V}}_k = \boldsymbol{U}_k$, $\overleftarrow{\boldsymbol{U}}_k = \boldsymbol{V}_k$, and $\overleftarrow{a}^r_{kl} = a^t_{kl}$. The resulting beamforming weights allocate equal power to each stream since $(\boldsymbol{V}_k)^{\dagger}_{*i}(\boldsymbol{V}_k)_{*i} = (\overleftarrow{\boldsymbol{U}}_k)^{\dagger}_{*i}(\overleftarrow{\boldsymbol{U}}_k)_{*i} = \frac{1}{d_k}$ for all $i \in \{1, \ldots, d_k\}$.

The combination of this virtual procedure, and the MMSE receiver weights (3.3) results in a set of beamforming weights that we will loosely refer to as MMSE, even though strictly speaking they do not minimize the sum mean-squared error at the receivers.

3.2 Globally Calculating the Beamforming and Combining Weights

The interdependency of the beamforming and combining weights can create dependency cycles that significantly complicate their optimization. Let us revisit the three link example, as shown in Figure 3.1, where each node has four antenna elements and each link carries two streams. Figure 3.1 depicts the interference suppression assignment given by

$$\boldsymbol{A}^t = \boldsymbol{A}^r = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{3.5}$$

Given the interference suppression assignment in (3.5), the ZF and MMSE weights of r_1 are dependent on the weights at t_1 and t_3. Furthermore, t_3's weights depend on the weights of r_3 and r_2. If the nulling assignment is followed, this sequence traverses every node and completes a cycle when calculating the weights of t_2, which depend on the weights of r_1, the initial node.

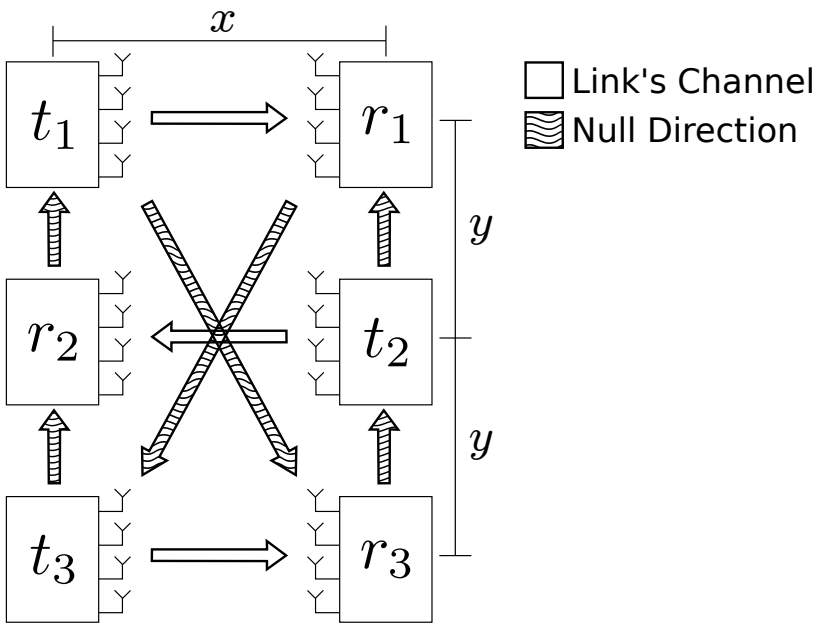

Figure 3.1: Topology of example three-link network. At least one high interfering node is located at a distance y from every receiver.

Our solution to the problem of dependency cycles is to iteratively compute the beamforming and combining weights. Similar iterative approaches were taken in the previous reported bilateral algorithms of [44, 52, 78, 85].

Our algorithm, called ComputeWeights, computes the set of all beamforming weights $\{\boldsymbol{V}\}$ and the set of all combining weights $\{\boldsymbol{U}\}$ by initializing the beamforming and combining weights according to the highest eigenmodes of the desired channel's singular value decomposition (SVD) [70] and computing the interference suppression weights iteratively until the weights converge (within a threshold ϵ) or a maximum number of iterations N_{max} is reached. We provide a summary of algorithm ComputeWeights in Figure 3.2.

For convenience, we define algorithm ComputeWeights as a general algorithm so that we can reuse it in the context of unilateral or bilateral interference suppression. As we will show in the next section, the inputs of ComputeWeights vary depending on the interference suppression strategy. The possible inputs for ComputeWeights are: the set of all channels $\{\boldsymbol{H}\}$; the interference suppression assignment matrices

$\boldsymbol{A}^t$ and $\boldsymbol{A}^r$; a stream allocation $\boldsymbol{d}$; a *node schedule* $\boldsymbol{s} = [s_1, s_2, \ldots, s_S]$ that defines the order in which weights are computed (important since different node orderings produce different results); and a flag F that indicates ZF ($F = 0$) or MMSE ($F = 1$) weights.

Next, we describe two unilateral interference suppression algorithms and a bilateral interference suppression algorithm that compute the particular weights of all links based on algorithm ComputeWeights.

Input: $(\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s}, F)$
Output: $(\{\boldsymbol{V}\}, \{\boldsymbol{U}\})$

1: **for** $k \leftarrow 1$ **to** M **do**
2: Initialize r_k's and t_k's weights to link k's SVD corresponding to the highest d_k eigenmodes;
3: **if** link k performs interference suppression **then**
4: Allocate equal power $\sqrt{\frac{1}{d_k}}$;
5: Remove any node in link k from $\boldsymbol{s}$ that does not perform interference suppression;
6: **else**
7: Allocate optimal power on the d_k highest eigenmodes via waterfilling;
8: Remove t_k and r_k from $\boldsymbol{s}$;
9: **end if**
10: **end for**
11: **for** *iteration* $\leftarrow 1$ **to** N_{max} **do**
12: **for** $i \leftarrow 1$ **to** S **do**
13: Set k equal to the link number of node s_i
14: **if** s_i is a receiver **then**
15: Compute r_k's weights using (3.2) if $F = 0$, or (3.3) and (3.4) if $F = 1$;
16: **else**
17: Reverse the communication link ;
18: Compute $\overleftarrow{r}_k$'s weights using (3.2) if $F = 0$, or (3.3) and (3.4) if $F = 1$;
19: **end if**
20: **end for**
21: Stop if the maximum absolute value of the difference of elements between the previous weights and the newly computed weighs is less than ϵ for all s_i;
22: **end for**
23: **return** $(\{\boldsymbol{V}\}, \{\boldsymbol{U}\})$;

Figure 3.2: Algorithm ComputeWeights for computing the weights of each node.

3.2.1 Global Weights for Unilateral Interference Suppression

We use ComputeWeights to create two instances that can compute the unilateral interference suppression weights, namely ComputeWeights with ZF unilateral interference suppression, and ComputeWeights with MMSE unilateral interference suppression. ComputeWeights with ZF unilateral interference suppression can be obtained using the inputs $(\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s}, F = 0)$, and ComputeWeights with MMSE unilateral interference suppression can be obtained using the inputs $(\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s}, F = 1)$. Here, $\boldsymbol{d}$ must be feasible, and $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$ must be the corresponding interference suppression assignment.

3.2.2 OBIC: A Cycle-Free Unilateral Strategy

Another unilateral strategy is the *order-based interference cancellation (OBIC)* strategy from [65, 92], which is based on the rule that nodes being scheduled must suppress interference from previously scheduled interfering nodes. OBIC is inherently unilateral since each node ignores interference to and from all nodes that are scheduled after it. The entries of $\boldsymbol{A}^r$ and $\boldsymbol{A}^t$ are populated as nodes are scheduled. We define a node schedule as *feasible under OBIC* if for every scheduled node, the node can suppress the interference from all previously scheduled nodes without violating the degrees-of-freedom constraints.

OBIC will not generally consider all stream allocations that are feasible. For example, in the context of the three-link example in Figure 3.1, the stream allocation $\boldsymbol{d} = [2, 2, 2]$ is not feasible under OBIC. The reason is that the OBIC scheduling mechanism specifically excludes cycles.

In [65], the authors propose that interference suppression under OBIC be done with ZF. However, we also define an MMSE-based OBIC strategy called *OBICmmse* that uses MMSE instead of ZF to perform interference suppression on the previously scheduled interfering nodes. We implement OBIC using ComputeWeights with the

inputs ($\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s}, F = 0$) and OBICmmse using ComputeWeights with inputs ($\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s}, F = 1$). These instances of ComputeWeights execute a single iteration ($N_{max} = 1$). Also, the input node schedule $\boldsymbol{s}$ must be feasible under OBIC and it defines $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$.

3.2.3 Global Weights for Bilateral Interference Suppression

We reuse Algorithm ComputeWeights to iteratively compute the beamforming and combining weights for the case of bilateral interference suppression. For this instance of ComputeWeights, we fix the node schedule to $\boldsymbol{s}_\circ = [r_1, r_2, \ldots, r_M, t_1, t_2, \ldots, t_M]$, i.e., all receivers first, followed by all transmitters. We obtain ComputeWeights with bilateral interference suppression using the inputs ($\{\boldsymbol{H}\}, \boldsymbol{A}^t, \boldsymbol{A}^r, \boldsymbol{d}, \boldsymbol{s} = \boldsymbol{s}_\circ, F = 1$). For this instance of ComputeWeights, the definition of $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$ are modified for bilateral interference suppression, so that $a^t_{l,k} = a^r_{k,l} = 1$ for $\rho_{kl} \geq \varrho$.

3.3 Numerical Results

This section is organized as follows. In Section 3.3.1, we present a specific example where the performance of bilateral interference suppression is significantly better than that of the best unilateral interference suppression. In Section 3.3.2 and Section 3.3.3, we present results comparing the bilateral versus the unilateral strategies in a three-link network and in a random eight link network, respectively. In Section 3.3.4, we show the advantage of overcoming dependency cycles over avoiding them for unilateral interference suppression. For all simulations, we fix $\varrho = -2.9$ dB, we fix $\epsilon = 0.0001$, we set the reference SNR and INR at one meter to 57.1 dB, and, unless otherwise stated, the SNR and INR vary inversely proportional to the distance cubed. Except for the example on Section 3.3.1, we assume a "quasi-static" flat-fading Rayleigh model where the channel is assumed constant for the duration of a burst, but random between bursts, and the channel elements are independent and identically distributed, complex Gaussian with zero mean and unit variance [37]. Finally, we assume that noise

at each receiver is white, satisfying $\boldsymbol{R}_{\boldsymbol{n}_k} = \boldsymbol{I}$ for all $k \in \{1, \ldots, M\}$.

For the following discussion, we compute the capacity by determining the SINR of each stream and computing the capacity $\dot{C}_k$, as given by (2.22).

3.3.1 An Example

Consider three links where each node has two antenna elements spaced at half-wavelength and each link carries a single stream. For this example only, we consider a channel without fading. We locate the transmitters and receivers as shown in Figure 3.1 with $y = 25$, and $x = 50$. For t_1, t_2, and t_3, the angles as measured counter-clockwise from the horizontal axis to the line through the two antennas are 131.1°, 136°, and 29.8°, respectively. For r_1, r_2, and r_3, these angles are 23.4°, 134.1°, and 135°, respectively. Taking t_3 as the origin, we place a reflector with 0.9 attenuation at $(25, 70)$. We set $N_{max} = 1000$, and exhaustively search for the node schedule and interference suppression assignment that produces the highest aggregate capacity for the unilateral approaches. In this scenario, the sum capacity using ComputeWeights with bilateral interference suppression is 10.22 bits/sec/Hz, while the sum capacity using the best unilateral strategy (ComputeWeights with MMSE unilateral interference suppression) is 3.12 bits/sec/Hz. The bilateral strategy thus outperforms the best unilateral strategy by 227%.

3.3.2 Three-Link Results

Unlike the previous Section 3.3.1, we return to Rayleigh fading. We consider the three-link example of Figure 3.1, where every node has four antennas. For each link, we allocate zero to four streams. Where applicable, we calculate weights for all stream allocations, interference suppression assignments, and node schedules. We set $N_{max} = 1000$, and we record only the highest capacity of every interference suppression assignment and node schedule that converge.

We show how the aggregate capacity varies with interference for this three-link

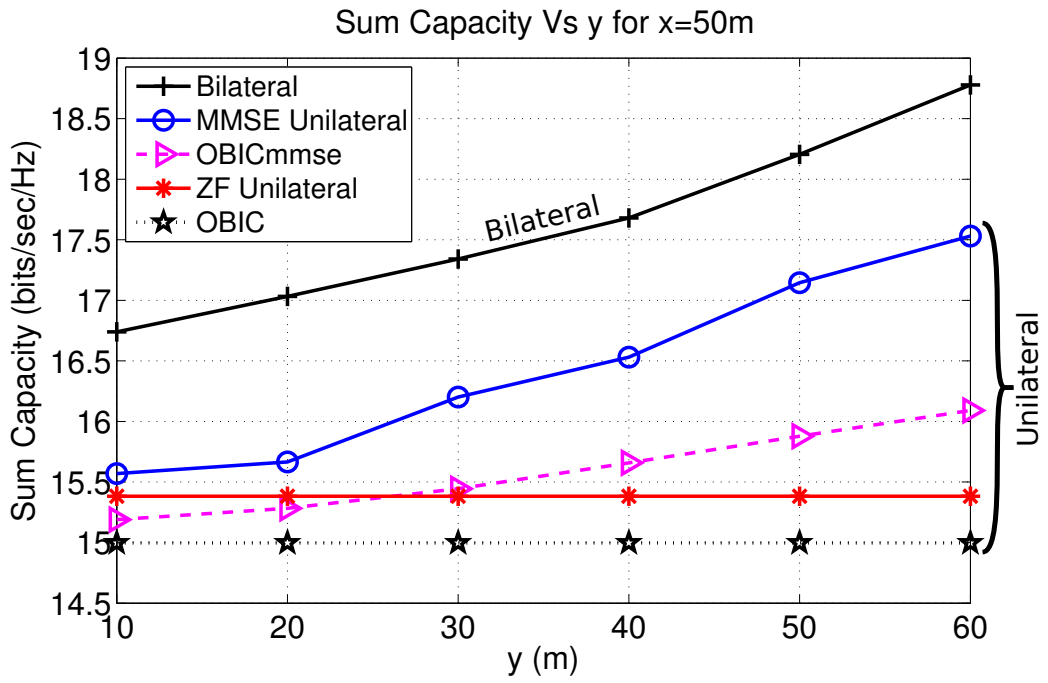

Figure 3.3: Maximum capacity of all stream allocations, node schedules, and interference suppression assignments for the three-link network of Figure 3.1.

example. Figure 3.3 shows the maximum capacity of all stream allocations, averaged over 100 trials, plotted as a function of the distance y for fixed $x = 50$. As can be observed, as interference decreases (y increases), the capacities of ComputeWeights with bilateral interference suppression, ComputeWeights with MMSE unilateral interference suppression, and OBICmmse increase. The slope, however, is larger for ComputeWeights with bilateral interference suppression and ComputeWeights with MMSE unilateral interference suppression than for OBICmmse. The sum capacity of ComputeWeights with MMSE unilateral interference suppression is between 5.8% and 8% less than the sum capacity of ComputeWeights with bilateral interference suppression for all values of y. Figure 3.3 also shows that the OBIC, ComputeWeights with ZF unilateral interference suppression, OBICmmse, and ComputeWeights with MMSE unilateral interference suppression strategies had at worst a 20%, 18%, 14%, and 8% capacity loss as compared to ComputeWeights with bilateral interference suppression.

3.3.3 Larger Network Results

In this section we will present results for two experiments with eight MIMO links. In Section 3.3.3.1, we will fix the eight-link topology and vary the interference path loss exponent. Then, in Section 3.3.3.2, we will fix the path loss exponent and vary the topology of the eight links.

For all simulations in this section, we fix all nodes to have four antenna elements, and we allocate zero to four streams at each link. Because of the size of the network we considered, the computation time required to test all possible stream allocations, all possible node schedules, and all possible interference suppression assignments is excessive. For this reason, we use the feasibility heuristic ExtendedGreedy from [95] to find a stream allocation space that is feasible for all simulations. Also, we use $\boldsymbol{A}^t$ and $\boldsymbol{A}^r$ provided by ExtendedGreedy as input to ComputeWeights with ZF/MMSE unilateral interference suppression. Additionally, we heuristically determine the node schedule for ComputeWeights with ZF/MMSE unilateral interference suppression by scheduling nodes that depend the least on other nodes first. For OBIC, we average the results over a maximum of five random OBIC feasible node schedules for each stream allocation. Finally, we fix $N_{max} = 10000$ and we only record data for stream allocations that converge.

3.3.3.1 Fixed Eight-Link Topology

We consider the fixed topology composed of eight MIMO links shown in Figure 3.4 in which each receiver is 50 meters from its transmitter. Let α_S be the desired signal's path-loss exponent, and α_I be the path-loss exponent between every interfering transmitter-receiver pair. We fix $\alpha_S = 3$ and vary α_I to vary the interference. We let $\varrho = -2.9$ dB and so $\alpha_I = 2.5$, $\alpha_I = 2.7$, $\alpha_I = 3$, $\alpha_I = 3.2$, and $\alpha_I = 3.5$ corresponds to 100%, 87%, 48%, 21%, and 13% of all interference satisfying $\rho_{kl} \geq \varrho$, respectively.

Figure 3.5 depicts the maximum sum capacity as a function of α_I averaged over

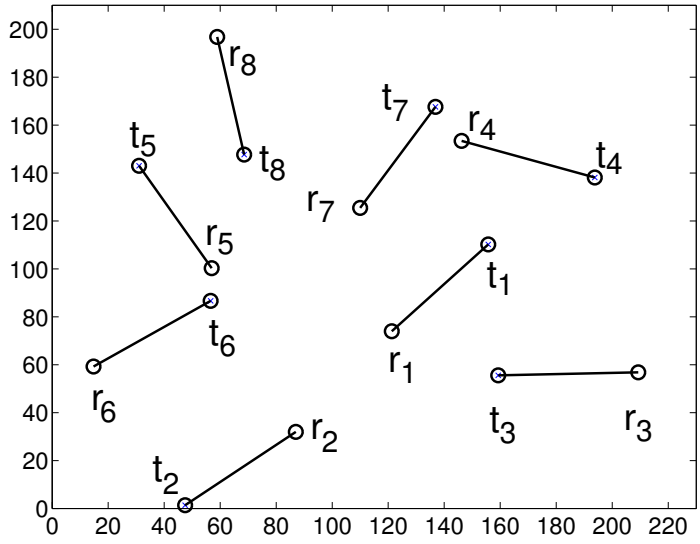

Figure 3.4: Topology of simulated eight link network.

50 random channel realizations. OBIC based strategies performed poorly at high interference ($\alpha_I = 2.5$) possibly due to their limited stream allocation space. ComputeWeights with bilateral interference suppression outperformed the best unilateral interference suppression method (ComputeWeights with MMSE unilateral interference suppression) by 26%, 8%, and 4% at high, medium, and low interference, respectively. Also, ComputeWeights with bilateral interference suppression outperformed OBIC and OBICmmse by 71% and 60%, 42% and 14%, and 15% and 5% at high, medium, and low interference, respectively. It is possible that other stream allocations exist in which ComputeWeights with bilateral interference suppression performs better than that of the results shown in Figure 3.5 since we have constrained the stream allocation space to be feasible. Clearly, Figure 3.5 shows that deviating from ComputeWeights with bilateral interference suppression in large networks, where many links are scheduled concurrently, can result in large penalties in the aggregate throughput.

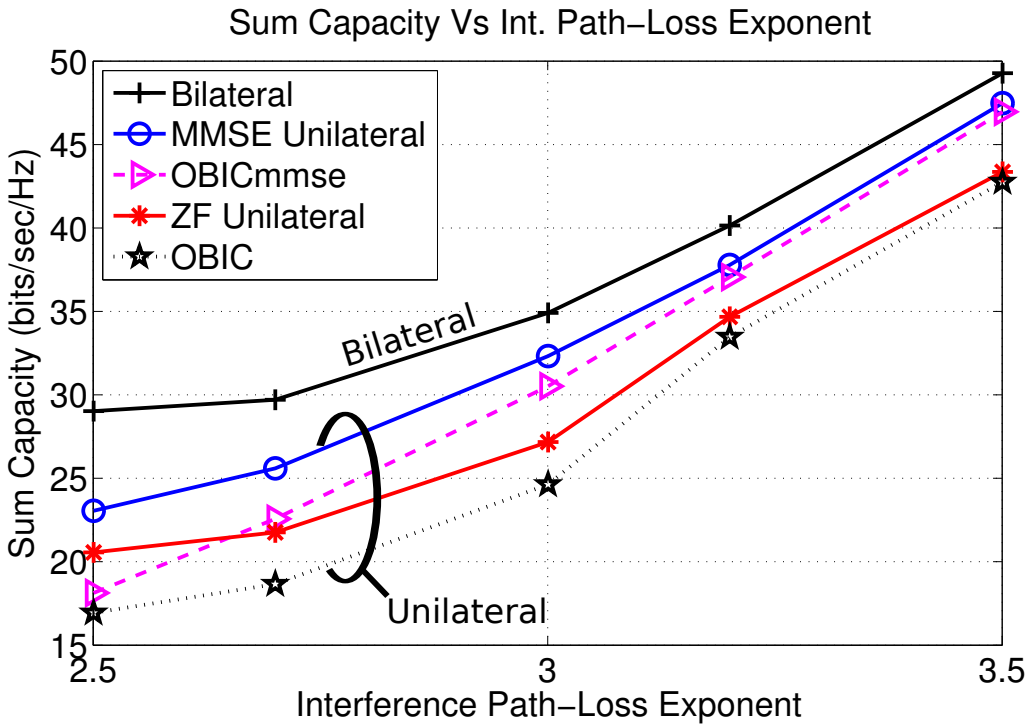

Figure 3.5: Plot of maximum sum capacity versus interference path-loss exponent α_I for fixed signal path-loss exponent $\alpha_S = 3$.

3.3.3.2 Random Eight-Link Topology

We now present results for eight MIMO links in which the eight transmitters are uniformly distributed in a square with a given area. Each receiver is 50 meters from its transmitter, with an angle chosen uniformly such that the receiver is located inside the square area.

Figure 3.6 depicts the maximum sum capacity versus area averaged over 50 random trials. When the area is small, medium (near $5 \times 10^4 m^2$), and large, nearly 100%, 43%, and 8% of all interference satisfy $\rho_{kl} \geq \varrho$, respectively. OBIC had the lowest aggregate capacity at all areas tested. OBIC was followed by OBICmmse in the small area region, and ComputeWeights with ZF unilateral interference suppression in the larger area region. ComputeWeights with bilateral interference suppression outperformed the best unilateral method, ComputeWeights with MMSE unilateral interference suppression, by 26%, 8.6%, and 3.3% at the small, medium, and large area

regions, respectively. Also, ComputeWeights with bilateral interference suppression outperformed OBIC and OBICmmse by 68% and 60%, 31% and 16%, and 9% and 4% at small, medium, and large area regions, respectively.

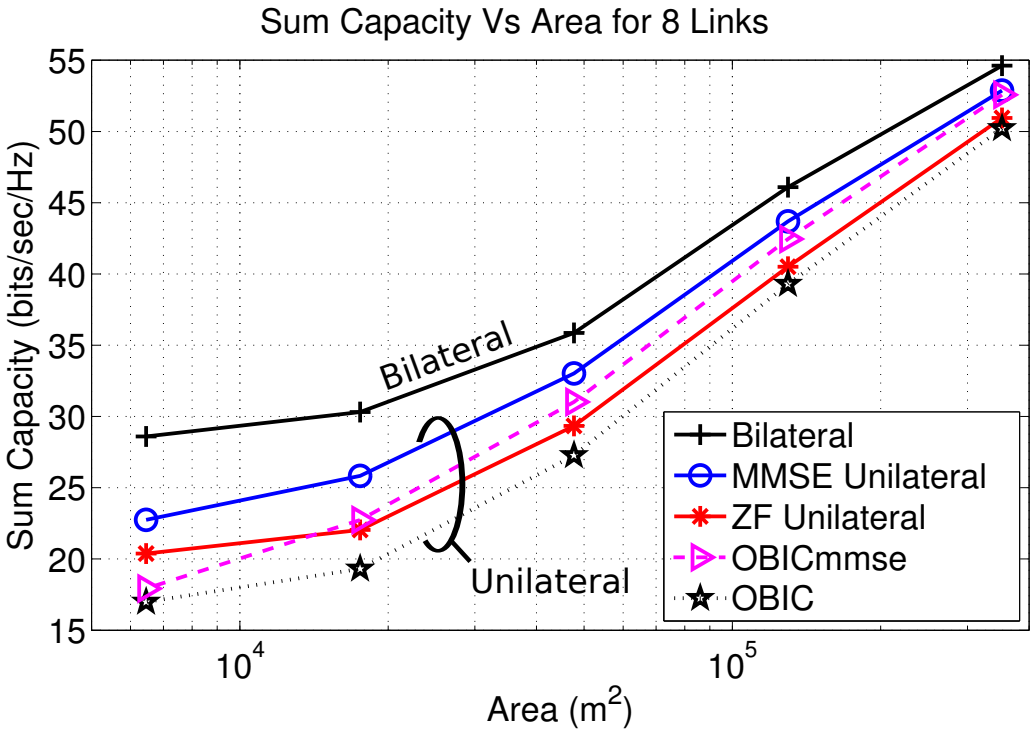

Figure 3.6: Average maximum capacity of networks with 4 antenna elements at every node, and up to 8 active links.

3.3.4 The Advantage of Overcoming Cycles

Using the same simulation setup from Section 3.3.2, we now look at how the aggregate capacity varies with different stream allocations to show the benefit of overcoming cycles. We compare only between ComputeWeights with ZF/MMSE unilateral interference suppression and OBIC/OBICmmse, but we also show results for ComputeWeights with bilateral interference suppression. Figure 3.7 shows the average capacity of 100 random channel realizations for the most relevant stream allocations and $x = y = 50$. The stream allocations $\boldsymbol{d} = [1, 1, 2]$, $\boldsymbol{d} = [1, 2, 1]$, $\boldsymbol{d} = [2, 1, 2]$, and $\boldsymbol{d} = [2, 2, 2]$ are the allocations in which OBIC/OBICmmse, ComputeWeights

with ZF unilateral interference suppression, ComputeWeights with MMSE unilateral interference suppression, and ComputeWeights with bilateral interference suppression achieved the highest capacity, respectively. Notice that for $\boldsymbol{d} = [2, 1, 2]$ and $\boldsymbol{d} = [2, 2, 2]$ an OBIC feasible node schedule does not exist (cycles cannot be avoided) and so we show no results for OBIC/OBICmmse for these allocations. Figure 3.7 shows that ComputeWeights with MMSE unilateral interference suppression for $\boldsymbol{d} = [2, 1, 2]$ outperforms the best of OBIC and OBICmmse by 14% and 8%, respectively. For a unilateral interference suppression strategy, these results show that overcoming cycles result in higher sum capacity than avoiding cycles because more streams can be allocated per link when cycles are present.

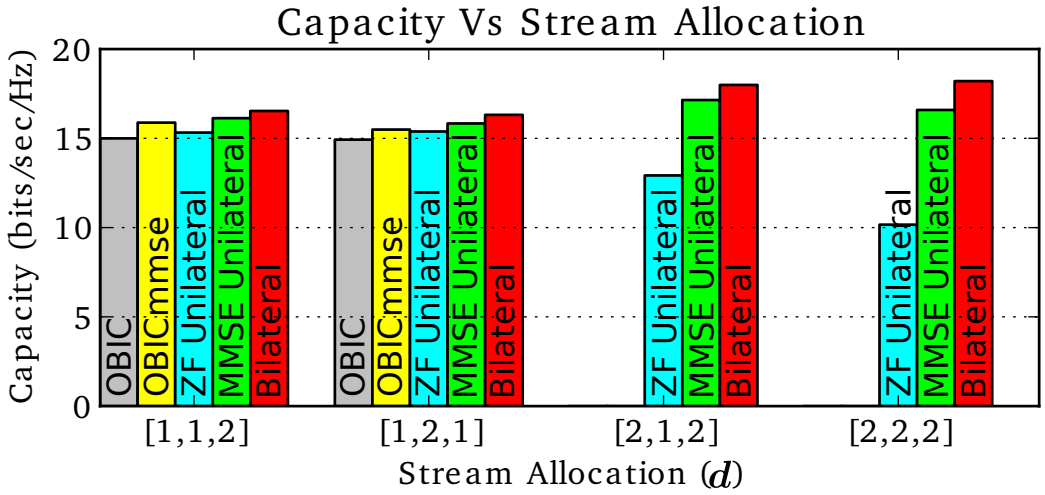

Figure 3.7: Capacities of most relevant stream allocations for the three-link topology of Figure 3.1.

3.4 Chapter Summary

In this chapter, we showed that, for a three-link example, a bilateral interference suppression strategy performs better than the best unilateral interference suppression strategy even after considering all node schedules and all interference suppression assignments for the unilateral interference suppression strategy. We showed that the performance loss of unilateral strategies can be greater in larger networks. Using

the three-link example, we showed that overcoming dependency cycles leads to a higher number of streams in the network than preventing cycles, which improves the performance of the network. We conclude that while the unilateral interference suppression strategy can aid network designers in determining the feasibility of a stream allocation in the network, it is ultimately the weight algorithm that determines the performance of the network. Thus, the weights should be computed bilaterally to find the best weights for network operation.

CHAPTER 4

JOINT OPTIMIZATION OF STREAM ALLOCATION AND BEAMFORMING AND COMBINING WEIGHTS FOR THE MIMO INTERFERENCE CHANNEL

In the previous Chapter 3, we presented a simple bilateral strategy that, given a set of interfering MIMO links and the number of streams to allocate to each link, determined the beamforming and combining weights of all links in the network. In this chapter, we design an algorithm, based on bilateral interference suppression, whose goal is to maximize the sum rate of a set of interfering MIMO links by jointly optimizing which subset of transmitters should transmit, the number of streams for each transmitter (if any), and the beamforming and combining weights that support those streams.

Most prior work has focused on a subset of the problems of determining which transmitters should transmit, how many streams each should transmit, and their corresponding beamforming and combining weights. The Max-SINR algorithm from [43, 44], the Max-SINR algorithm from [78], and the bilateral algorithm from Chapter 3 are examples of algorithms that only compute the beamforming and combining weights. These algorithms require a priori specification of which transmitters should transmit as well as how many streams each should transmit.

Other work has focused on the joint problem of determining the beamforming and combining weights (or transmit covariance matrices), and the number of streams that each transmitter should transmit. Examples include the MWSR algorithm from [72], the GP algorithm from [108], the greedy algorithm from [33], and the BR algorithm from [4]. All of these works require a separate algorithm to determine which transmitters should transmit.

Other work has focused on determining which transmitters should transmit, and how many streams each should transmit. The works of [7, 9, 21, 46, 65, 92, 95], for example, propose either linear programming formulations or heuristic algorithms that determine which transmitters should transmit and how many streams each should transmit. The proposed solutions of these works require a separate algorithm to compute the beamforming and combining weights and do not necessarily maximize the sum rate, since the decisions are not based on the beamforming and combining weights of the links, which ultimately determine the performance.

Our main contributions in this chapter are as follows:

- As a stepping stone, we first extend the seminal results of Sampath et al. [84] to design, for a single link in the presence of interfering links, the joint optimal beamforming and combining weights that minimize the sum weighted MSE across all streams in the network given the beamforming and combining weights of the interfering links. The transceiver has the following characteristics:
  - it has the ability to deactivate itself if doing so minimizes the weighted sum MSE;
  - it diagonalizes the MIMO channel; and
  - it reduces to the optimal eigen-mode transmission with power allocated through waterfilling for the special case where the transmitter causes no interference to any receiver.
- We provide an interpretation to the structure of the transceiver and show that the optimal combiner is a minimum mean-squared-error (MMSE) combiner, and that the optimal beamformer can be viewed as the MMSE combiner of a virtual network.
- Using our transceiver and the results from [20, 72, 84] that relate the minimum

weighted sum MSE to the maximum weighted sum rate, we design an algorithm whose goal is to maximize the sum rate. The algorithm has the following characteristics:

- it determines which subset of transmitters should transmit;
- uses only local information, a required property for any distributed implementation;
- it outperforms previously reported algorithms at high interference; and
- it achieves comparable performance to the top-performing previously reported algorithms at medium and low interference.

This chapter is organized as follows. In Section 4.1, we design the joint beamforming and combining weights for a single link. In Section 4.2, we review the relationship between the sum rate and the minimum weighted sum MSE. In Section 4.3, we propose our algorithm for computing the beamforming and combining weights for all links. In Section 4.4, we present numerical results. Finally, in Section 4.5, we provide a brief summary of this chapter.

4.1 Jointly Computing the Beamforming and Combining Weights for a Single Link

We begin by optimizing the beamforming and combining weighs for a single link in the presence of interfering links as a stepping stone towards computing the beamforming and combining weights for all links. In this section, we design the joint transceiver that minimizes the weighted sum MSE across all streams in the network. We choose the weighted sum MSE criterion because, as we will review in Section 4.2, it reduces to maximizing the sum rate as a special case [20, 72, 84]. Later, in Section 4.3, we will use the joint transceiver to design an algorithm that computes the beamforming and combining weights of all links.

4.1.1 The Minimum Weighted Sum MSE Problem

We formulate the weighted sum MSE optimization for the beamforming and combining weights of link k as

$$(\boldsymbol{V}_k^*, \boldsymbol{U}_k^*) = \arg \min_{\boldsymbol{V}_k, \boldsymbol{U}_k} \sum_{l=1}^{M} \mathrm{tr}\left(\boldsymbol{W}_l \boldsymbol{E}_l\right)$$
$$\text{such that } \mathrm{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger\right) \leq p_k, \tag{4.1}$$

where

$$\boldsymbol{E}_k = \mathbb{E}[(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^\dagger] \tag{4.2}$$

is the error covariance matrix of link k and contains the MSE of the streams of link k in the diagonal. In (4.1), the error weight matrix $\boldsymbol{W}_k \in \mathbb{R}^{d_k \times d_k}$ is a diagonal matrix of nonnegative weights associated with the MSE of the streams of link k.

Whereas previous works solve for $\boldsymbol{V}_k$ as a function of $\boldsymbol{U}_k$ in (4.1) and vice versa, we perform a joint optimization so as to find $\boldsymbol{V}_k$ and $\boldsymbol{U}_k$ simultaneously. Due to both the joint transceiver design and the inequality power constraint $\mathrm{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger\right) \leq p_k$ in (4.1), the resulting beamforming and combining weights have the capability to deactivate (or activate) its link if doing improves the overall performance. As we will see using numerical results in Section 4.4, deactivating links is desirable at high interference since interference caused by one link highly affects the performance of all other links in the network.

4.1.2 The Minimum Weighted Sum MSE Solution

The solution to (4.1) can be expressed in terms of the following compact singular-value decomposition (SVD):

$$\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} \boldsymbol{P}_{\bar{k}}^{-1/2} = \boldsymbol{F}_k \boldsymbol{D}_k \boldsymbol{G}_k^\dagger, \tag{4.3}$$

where $\boldsymbol{D}_k \in \mathbb{R}^{d_k \times d_k}$ is a diagonal matrix containing the nonzero singular values of $\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} \boldsymbol{P}_{\bar{k}}^{-1/2}$ ordered in decreasing order from top left to bottom right; $\boldsymbol{F}_k \in$

$\mathbb{C}^{n_{r_k} \times d_k}$ and $\boldsymbol{G}_k \in \mathbb{C}^{n_{t_k} \times d_k}$ have orthonormal column vectors that correspond to the left and right eigenvectors of $\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} \boldsymbol{P}_{\bar{k}}^{-1/2}$ with nonzero singular values, respectively; and $\boldsymbol{P}_{\bar{k}}$ is

$$\boldsymbol{P}_{\bar{k}} = \underbrace{\sum_{l=1, l \neq k}^{M} \boldsymbol{H}_{lk}^{\dagger} \boldsymbol{U}_l \boldsymbol{W}_l \boldsymbol{U}_l^{\dagger} \boldsymbol{H}_{lk}}_{\boldsymbol{\Pi}_k} + \mu_k \boldsymbol{I}, \tag{4.4}$$

where $\mu_k \geq 0$ is a Lagrange multiplier that must satisfy the Karush-Kuhn-Tucker (KKT) conditions for the optimization of (4.1) (see Appendix A).

The SVD in (4.3) requires that $\boldsymbol{P}_{\bar{k}}$ be invertible, which is clearly true whenever $\mu_k > 0$. In the following, we present the beamforming and combining weights that solve (4.1) by assuming that we already know μ_k and that $\boldsymbol{P}_{\bar{k}}$ is invertible. Later in this section, we show how to obtain μ_k, and show that $\boldsymbol{P}_{\bar{k}}$ is always invertible.

Theorem 1. *The joint beamforming and combining weights that solve* (4.1) *are given by*

$$\boldsymbol{V}_k = \boldsymbol{P}_{\bar{k}}^{-1/2} \boldsymbol{G}_k \boldsymbol{\Theta}_k, \tag{4.5}$$

$$\boldsymbol{U}_k = \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{F}_k \boldsymbol{\Phi}_k, \tag{4.6}$$

where

$$\boldsymbol{\Theta}_k = \left(\boldsymbol{W}_k^{1/2} \boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2} \right)_+^{1/2}, \tag{4.7}$$

$$\boldsymbol{\Phi}_k = \boldsymbol{W}_k^{-1/2} \boldsymbol{\Theta}_k, \tag{4.8}$$

and $(\cdot)_+$ *is the matrix* $(\cdot)$ *with the negative entries replaced with zeros.*

Proof. See Appendix A on page 122.

Using Theorem 1, we can rewrite the received signal after combining (2.21) as

$$\hat{\boldsymbol{x}}_k = \boldsymbol{\Phi}_k \left(\boldsymbol{D}_k \boldsymbol{\Theta}_k \boldsymbol{x}_k + \hat{\boldsymbol{n}}_k \right), \tag{4.9}$$

where $\hat{\boldsymbol{n}}_k = \boldsymbol{F}_k^{\dagger} \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{z}_k$ is a vector of white Gaussian noise satisfying

$$\mathbb{E}[\hat{\boldsymbol{n}}_k \hat{\boldsymbol{n}}_k^{\dagger}] = \boldsymbol{F}_k^{\dagger} \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{R}_{\bar{k}} \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{F}_k = \boldsymbol{I}.$$

Therefore, the beamforming and combining weights in (4.5) and (4.6) diagonalize the MIMO channel.

To complete the solution to (4.1), we must find μ_k. Because a closed-form solution to μ_k is unknown, and because $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger)$ is a decreasing function of μ_k [78, 88], we search for the value of μ_k as follows.

First, let us consider the case in which $\boldsymbol{\Pi}_k$ in (4.4) is invertible. For this case, we first test for $\mu_k = 0$. If the beamforming weights satisfy $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) \leq p_k$, then the search is done because all KKT conditions are satisfied. If $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) > p_k$, then we search for the $\mu_k > 0$ such that $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) = p_k$, thereby satisfying all KKT conditions.

Now, let us consider the case in which $\boldsymbol{\Pi}_k$ is singular. For this case, we state the following lemma.

Lemma 1. *If* $\boldsymbol{\Pi}_k$ *in* (4.4) *is singular, then the limit of* $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger)$ *using* $\boldsymbol{V}_k$ *in* (4.5) *as* μ_k *approaches zero from the right is*

$$\lim_{\mu_k \to 0^+} \mathrm{tr}\left(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\right) = \infty. \tag{4.10}$$

Proof. See Appendix B on page 128.

Lemma 1 together with the fact that $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger)$ is a decreasing function of μ_k suggest that whenever $\boldsymbol{\Pi}_k$ is singular, there exist a $\mu_k > 0$ such that $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger) = p_k$, which satisfies all KKT conditions. Therefore, $\boldsymbol{P}_k$ is always invertible.

Figure 4.1 summarizes the algorithm for finding μ_k for the general case. In our simulations, we use the bisection method to perform the search for $\mu_k > 0$.

4.1.3 Interpreting the Solution

The beamforming and combining weights in (4.5) and (4.6) have three components that can be inter-related through the use of a virtual network in which receivers become virtual transmitters and transmitters become virtual receivers. The concept of a virtual network has been previously used to aid the design of the transmitter's beamforming weights in the works of [25, 43, 44, 82].

1: **if** $\boldsymbol{\Pi}_k$ is invertible **then**
2: Set $\mu_k = 0$;
3: **if** $\mathrm{tr}(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger) \leq p_k$ **then**
4: **return** μ_k;
5: **end if**
6: **end if**
7: Find $\mu_k > 0$ such that $\mathrm{tr}(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger) = p_k$;
8: **return** μ_k;

Figure 4.1: Pseudocode for finding μ_k.

To build the virtual network that relates (4.5) and (4.6), let us define $\overleftarrow{\boldsymbol{H}}_{lk} = \boldsymbol{H}_{kl}^\dagger$ as the virtual MIMO channel between the virtual transmitter of link k and the virtual receiver of link l; $\overleftarrow{\boldsymbol{V}}_k = \boldsymbol{U}_k \boldsymbol{W}_k^{1/2}$ as the virtual beamforming weights of link k; $\overleftarrow{\boldsymbol{U}}_k = \boldsymbol{V}_k \boldsymbol{W}_k^{-1/2}$ as the virtual combining weights of link k; and $\overleftarrow{\boldsymbol{R}}_{n_k} = \mu_k \boldsymbol{I}$ as the virtual noise covariance of the virtual receiver of link k.

Figure 4.2 shows a block diagram of the transmit and receive structure for link k and highlights the three components of the joint transceiver using dotted boxes. The three components and their functions are as follows:

- *Whitening Component* – The first component of the transceiver is a whitening component. At the receiver side, the receiver whitens the interference plus noise of the received signal. At the transmitter side, this whitening component performs a similar function by whitening a "virtual" interference plus noise with covariance matrix given by

$$\overleftarrow{\boldsymbol{R}}_{\bar{k}} = \sum_{l=1, l\neq k}^{M} \overleftarrow{\boldsymbol{H}}_{kl} \overleftarrow{\boldsymbol{V}}_l \overleftarrow{\boldsymbol{V}}_l^\dagger \overleftarrow{\boldsymbol{H}}_{kl}^\dagger + \overleftarrow{\boldsymbol{R}}_{\boldsymbol{n}_k} = \boldsymbol{P}_{\bar{k}}. \tag{4.11}$$

- *Rotating Component* – The second component is a rotating component. Using this component, both the transmitter and the receiver rotate their signal so as to diagonalize their MIMO channel. The rotating matrices are chosen based on the SVD of the cascade of the whitening components and the MIMO channel $\left(\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} \boldsymbol{P}_{\bar{k}}^{-1/2}\right)$.

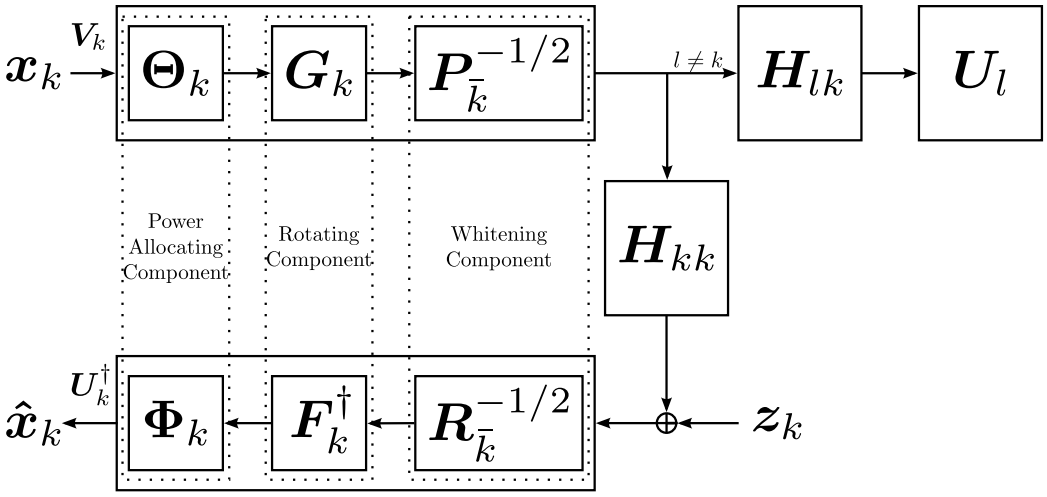

Figure 4.2: Block diagram of the components of the joint beamforming and combining weights.

- *Power Allocating Component* – The third component is a power allocating component that scales each element of the signal vector. Due to the $(\cdot)_+$ operator in (4.7), the power allocating component at the transmitter can potentially prevent some, if not all, streams from being transmitted. On the virtual network, the receiver's power allocating component acts similarly to the transmitter's power allocating component by scaling some signal elements and even reducing the number of streams on the virtual link.

The three components operate jointly to determine the number of streams to allocate to the link (if any). After whitening the channel, the rotating components provide the appropriate rotating matrices to use for communication through the MIMO channel. Then, the power allocating components allocate non-zero power to the link's i^{th} stream if $(\boldsymbol{W}_k)_{ii}^{1/2} > (\boldsymbol{D}_k)_{ii}^{-1}$ and allocate zero power otherwise, where $(\cdot)_{ij}$ is the element at the i^{th} row and j^{th} column of $(\cdot)$. Clearly, the higher the values of $(\boldsymbol{W}_k)_{ii}^{1/2}$ and $(\boldsymbol{D}_k)_{ii}$ are, the more likely that $(\boldsymbol{W}_k)_{ii}^{1/2} > (\boldsymbol{D}_k)_{ii}^{-1}$ will be satisfied. Note that the values of $\boldsymbol{D}_k$ in (4.3) are determined according to the whitening components and that decreasing the values of $\boldsymbol{R}_{\bar{k}}$ and $\boldsymbol{P}_{\bar{k}}$ results in higher values of $\boldsymbol{D}_k$. Therefore, the threshold values of $\boldsymbol{D}_k$ used to determine whether or not to allocate zero power

are determined by the received interference plus noise at the receiver ($\boldsymbol{R}_{\bar{k}}$), the interference caused by the transmitter to other receivers ($\boldsymbol{\Pi}_k$), and the power allocated through the beamformer (determined by μ_k).

Using the MMSE combining weights for link k, as given by [72, 78, 88]

$$\boldsymbol{U}_k^{\text{MMSE}} = \left(\boldsymbol{H}_{kk}\boldsymbol{V}_k\boldsymbol{V}_k^{\dagger}\boldsymbol{H}_{kk}^{\dagger} + \boldsymbol{R}_{\bar{k}}\right)^{-1}\boldsymbol{H}_{kk}\boldsymbol{V}_k, \tag{4.12}$$

we can further relate the joint beamforming and combining weights, as described by the following lemmas.

Lemma 2. *If the beamforming weights for link k are given by* (4.5)*, then the MMSE combining weights for link k are equal to* (4.6)*.*

Lemma 3. *If the combining weights for link k are given by* (4.6)*, then the MMSE combining weights for link k of the virtual network are $\overleftarrow{\boldsymbol{U}}_k^{MMSE} = \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k$, and so the beamforming weights of the real network are given by* (4.5)*.*

Proof. See Appendix C on page 130 for the proof of Lemma 3. We omit the proof of Lemma 2 because of its similarity to the proof of Lemma 3.

Remark 1. *By the data processing theorem from information theory, we know that $C_k \geq \hat{C}_k$ for any choice of $\boldsymbol{U}_k$. Using the matrix inversion lemma, it is easy to show that the MMSE combining weights in* (4.12) *are information lossless, so that $C_k = \hat{C}_k$ [72]. Therefore, from Lemma 2, it is clear that the combining weights given by* (4.6) *are also information lossless.*

4.2 The Weighted Sum MSE and the Sum Rate

We have chosen the weighted sum MSE as our objective function because, with a proper choice of the error weight matrix $\boldsymbol{W}_k$, minimizing the weighted sum MSE also maximizes the sum rate. This relationship was exploited for the single MIMO link in the absence of interference by Sampath et al. in [84], for the MIMO broadcast

channel by Christensen et al. in [20], and for the MIMO interference channel by Negro et al. in [72]. In the following subsections, we discuss how to choose the error weight matrices that maximize the sum rate for two separate cases.

4.2.1 Case 1: The MIMO Link That Causes No Interference, but Receives Interference

In [84], Sampath et al. showed that for a single MIMO link in the absence of interference, minimizing the weighted MSE will also maximize the rate on the link when the error weights $\boldsymbol{W}$ are chosen appropriately. In this section, we generalize this result to include the case in which the receiver is subject to interference from other links.

Consider the case where we wish to maximize the rate on a MIMO link k in which the transmitter causes no interference to any receiver, but in which the receiver is interfered by other transmitters. For this case, the solution is to transmit through the eigen modes of the whitened channel ($\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}$) and allocate power through waterfilling the same way that a greedy transmitter does [33, 108]. We find that for this scenario, the $\boldsymbol{W}_k$ that maximizes the rate on the link when optimizing (4.1) is given by $\boldsymbol{W}_k = \alpha_k \boldsymbol{\Lambda}_k$, where $\boldsymbol{\Lambda}_k^{1/2} \in \mathbb{R}^{d_k \times d_k}$ is a diagonal matrix containing the singular values of $\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}$ in decreasing order from top left to bottom right and α_k is any positive scalar. The following proposition summarizes this result.

Proposition 1. *For the case when the transmitter of link k causes no interference to other receivers but the receiver of link k is subject to interference from other transmitters and* $\boldsymbol{W}_k = \alpha_k \boldsymbol{\Lambda}_k$ *for any scalar* $\alpha_k > 0$*, the beamforming and combining weights in* (4.5) *and* (4.6) *reduce to the optimal eigen-mode transmission with power allocated through waterfilling.*

Proof. See Appendix D on page 132.

4.2.2 Case 2: The MIMO Link that Causes and Receives Interference

Consider the general case of the MIMO interference channel where all transmitters interfere with all receivers. In [72], the authors find that the gradient of the sum rate and the gradient of the weighted sum MSE are equal if

$$\boldsymbol{W}_k = \boldsymbol{I} + \boldsymbol{B}_k^{\dagger}\boldsymbol{V}_k^{\dagger}\boldsymbol{H}_{kk}^{\dagger}\boldsymbol{R}_{\bar{k}}^{-1}\boldsymbol{H}_{kk}\boldsymbol{V}_k\boldsymbol{B}_k, \tag{4.13}$$

where $\boldsymbol{B}_k$ is an arbitrary unitary matrix.

Notice that $\boldsymbol{W}_k$ in (4.13) is a function of $\boldsymbol{V}_k$, which is itself one of the variables to optimize. To solve this interdependency, the authors of [72] propose to compute $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ in separate steps in an iterative algorithm. We follow the same approach.

In our formulation of (4.1), we require that $\boldsymbol{W}_k$ be diagonal. To guarantee that (4.13) is always diagonal, we choose $\boldsymbol{B}_k$ in (4.13) from the following SVD:

$$\boldsymbol{A}_k\boldsymbol{S}_k\boldsymbol{B}_k^{\dagger} = \boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_k\boldsymbol{V}_k, \tag{4.14}$$

where $\boldsymbol{S}_k \in \mathbb{R}^{d_k \times d_k}$ is a diagonal matrix containing the singular values of $\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_k\boldsymbol{V}_k$ ordered in decreasing order from top left to bottom right; $\boldsymbol{B}_k \in \mathbb{C}^{d_k \times d_k}$ is a unitary matrix; and $\boldsymbol{A}_k \in \mathbb{C}^{n_{r_k} \times d_k}$ has orthonormal column vectors. This way, (4.13) becomes

$$\boldsymbol{W}_k = \boldsymbol{I} + \boldsymbol{S}_k^2. \tag{4.15}$$

This choice of $\boldsymbol{B}_k$ was used by Christensen et al. in [20] to design the WSRBF-WMMSE-D algorithm with diagonal weighting matrix for the MIMO broadcast channel.

4.3 Computing the Beamforming and Combining Weights for all Links

We now propose an algorithm whose goal is to maximize the sum rate of a set of interfering MIMO links by jointly optimizing the number of streams (if any) on each link as well as their beamforming and combining weights.

The proposed algorithm is summarized in Figure 4.3. Our algorithm begins by activating all links and allocating the maximum number of streams at every transmitter, as shown on Line 1 of Figure 4.3, where $\boldsymbol{I}_{n_{t_k} \times d_k}$ is an $n_{t_k} \times d_k$ matrix with ones on its diagonal and zeros elsewhere. Then, on lines 2-6, the algorithm computes the beamforming and combining weights iteratively. In Line 3, the receivers compute their interference-plus-noise covariance, error weights, and combining weights using the previously computed beamforming weights. In Line 4, the transmitters compute their beamforming weights using the previously computed interference-plus-noise covariance, error weights, and combining weights.

1: Initialize $d_k = \text{rank}\,(\boldsymbol{H}_{kk})$, $\boldsymbol{V}_k = \sqrt{\frac{p_k}{d_k}} \boldsymbol{I}_{n_{t_k} \times d_k}$ for all $k \in \{1, \ldots, M\}$;
2: **for** *iteration* $\leftarrow 1$ **to** N_{max} **do**
3: Compute $\boldsymbol{R}_{\bar{k}}$ using (2.20), $\boldsymbol{W}_k$ using (4.15), and $\boldsymbol{U}_k^{\text{MMSE}}$ using (4.12) for all $k \in \{1, \ldots, M\}$;
4: Compute $\boldsymbol{V}_k$ using (4.5) for all $k \in \{1, \ldots, M\}$;
5: Stop if the maximum absolute value of the difference of elements between the previous $\boldsymbol{Q}_k = \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}$ and the newly computed $\boldsymbol{Q}_k$ is less than ϵ for all $k \in \{1, \ldots, M\}$;
6: **end for**

Figure 4.3: Pseudocode for computing the beamforming and combining weights of all links.

The proposed algorithm has three desirable properties. First, the proposed algorithm will disable, or re-enable, links if it determines that doing so improves the overall performance. Second, the computation of the beamforming or combining weights at a node requires only local information. That is, the computation of the beamforming weights at a node requires information only from the desired receiver and from those receivers it causes interference to. Similarly, the computation of the combining weights at a node requires information only from the desired transmitter and from those transmitters that it receives interference from. Requiring only local information is a desired property for any distributed implementation. Third, it is

easy to prove using the technique from [20] that our proposed algorithm is guaranteed to converge, since at every iteration the algorithm moves monotonically towards a bounded objective.

For the MIMO interference channel, it is well known that the problem of maximizing the sum rate is non-convex [108]. Therefore, the proposed algorithm in Figure 4.3 can only guarantee finding a local maximum on the sum rate. Additionally, different initial conditions on the beamforming weights in Line 1 of Figure 4.3 will lead to different solutions. Similar to [78], we find that random initializations are just as good as any smart initialization we have tried, such as initializing the beamforming weights to the optimal SVD weights in the absence of interference. In the results presented in the following section, we choose the initialization stated in Line 1 of Figure 4.3.

A similar algorithm to the proposed algorithm can be found in [90], which was independently developed for a more general problem. However, the algorithm in [90] derives the beamforming weights as a function of the combining weights and vice versa. Thus, the algorithm from [90] is unable to deactivate links.

4.4 Numerical Results

In this section, we present numerical results comparing the sum rate and time complexity of the proposed algorithm to those of the previously reported algorithms in various levels of interference. For all simulations, we assume a "quasi-static" flat-fading Rayleigh channel model where the channel is assumed constant for the duration of a burst, but random between bursts [37]. Also, we assume that noise at each receiver is white, satisfying $\boldsymbol{R}_{\boldsymbol{n}_k} = \boldsymbol{I}$ for all $k \in \{1, \ldots, M\}$. We set the reference signal-to-noise ratio (SNR) and interference-to-noise ratio (INR) at one meter to 65.3 dB and assume a path-loss exponent of three. We set each node to have four antenna elements. Unless otherwise specified, we fix the distance between the transmitter and its corresponding receiver to 50 meters. We uniformly distribute the center of each

link within a circle of a given radius. Also, we uniformly distribute the angles from the horizontal axis to the line that goes through the transmitter and receiver of every link from zero to 2π.

For all algorithms, we initialize the beamforming weights as shown on Line 1 of Figure 4.3, and use the convergence criterion as shown on Line 5 of Figure 4.3 with $\epsilon = 0.0001$. Additionally, we set the maximum number of iterations to $N_{max} = 10000$. If an algorithm reaches the maximum number of iterations, the algorithm stops and we record the sum rate.

In the following, we show results for the MWSR algorithm from [72] and the GP algorithm from [108]. We also show results for the MMSE algorithm from [78, 88], since, as we will see in Section 4.4.2, it has good performance when the number of links is high. We do not include a comparison against the SDP algorithm from [83], because on the few sample runs we attempted, we found the execution time of the SDP algorithm to be about two to three orders of magnitude higher than other methods. Additionally, we do not show a comparison with the linear-approximation-based algorithm from [60] since it was shown in [90] to have higher complexity and comparable sum rate than approaching the problem through the weighted sum MSE criterion.

The rest of this section is organized as follows. In Section 4.4.1, we fix the number of links and vary the radius of the circle in which the links are placed. In Section 4.4.2, we fix the radius of the circle and vary the number of links within the circle. Finally, in Section 4.4.3, we compare the complexity of our proposed algorithm in terms of CPU time.

4.4.1 Sum Rate Versus Circle Radius

We consider ten MIMO links and vary the radius of the circle in which these links are placed. In Figure 4.4, we show the sum rate, averaged over 100 trials, plotted as

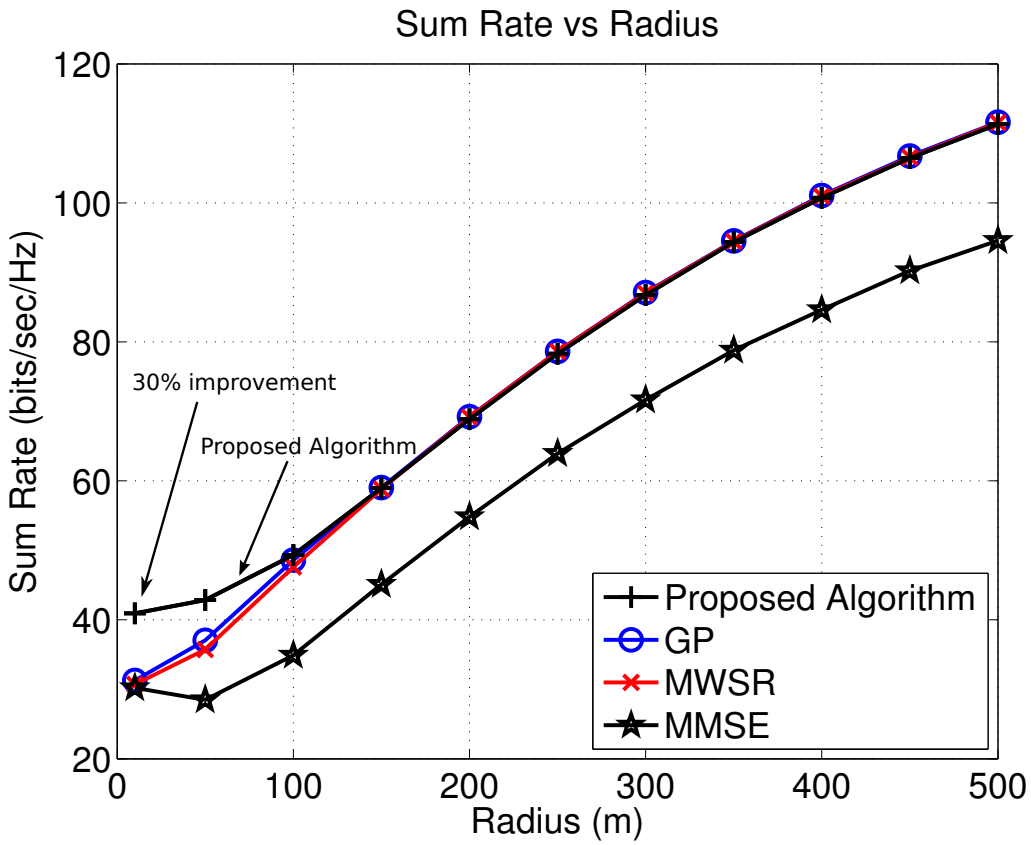

Figure 4.4: Sum rate as a function of the radius of the circle where the center of the ten links are placed.

a function of the radius of the circle. In this experiment, a large radius corresponds to a sparse scenario where interference from other links is low (resulting in low INR between links). In contrast, a small radius corresponds to a dense scenario where interference from other links is high (resulting in high INR between links). Note that because the distance between a transmitter and its corresponding receiver is fixed at all times, the SNR is equal and fixed for every link, independent of the radius of the circle.

The results of Figure 4.4 show that our proposed algorithm achieves at least 30% higher sum rate at the lowest radius tested (10 meters), as compared to the other algorithms. In this range, interference is high and the degrees-of-freedom available from the multiple antennas on the nodes are not enough to support high performance

on all links. The proposed algorithm achieves higher performance at high interference because, not only does it optimize the number of streams on each transmitter and the corresponding beamforming and combining weights for each stream, but it also optimizes which subset of transmitters should transmit. On the average, the proposed algorithm activated slightly more than half the links within the circle at high interference and almost all links at low interference. The MMSE algorithm from [78, 88] minimizes the unweighted sum MSE and, therefore, does not necessarily achieve a high sum rate.

4.4.2 Sum Rate Versus Number of Links at High Interference

In this section, we fix the radius of the circle to ten meters and vary the number of links placed within this circle. Note again that the SNR is equal and fixed for every link, independent of the number of links placed within the circle. Figure 4.5 shows the sum rate, averaged over 100 trials, plotted as a function of the maximum number of active links when the radius of the circle is fixed to ten meters. These results show that as the number of links increases, the sum rate of the GP, MWSR, and MMSE algorithms decrease while the sum rate of the proposed algorithm increases. Our algorithm achieves a sum rate that is at least 65% better than the sum rate of the other algorithms at the largest number of links tested. When the number of links is large, the proposed algorithm achieves a sum rate that is at least 90% better than that of the GP and MWSR. The proposed algorithm achieves high performance, and its performance increases as the number of links increases, since the algorithm has diversity on which links to activate and deactivate. On the average, for the results shown on Figure 4.5, our proposed algorithm activates between six to eight links.

When the number of antenna elements at every node is increased, the performance gap between the proposed algorithm and both the GP and MWSR is narrowed, especially when the number of links in the circle is low. This is expected since

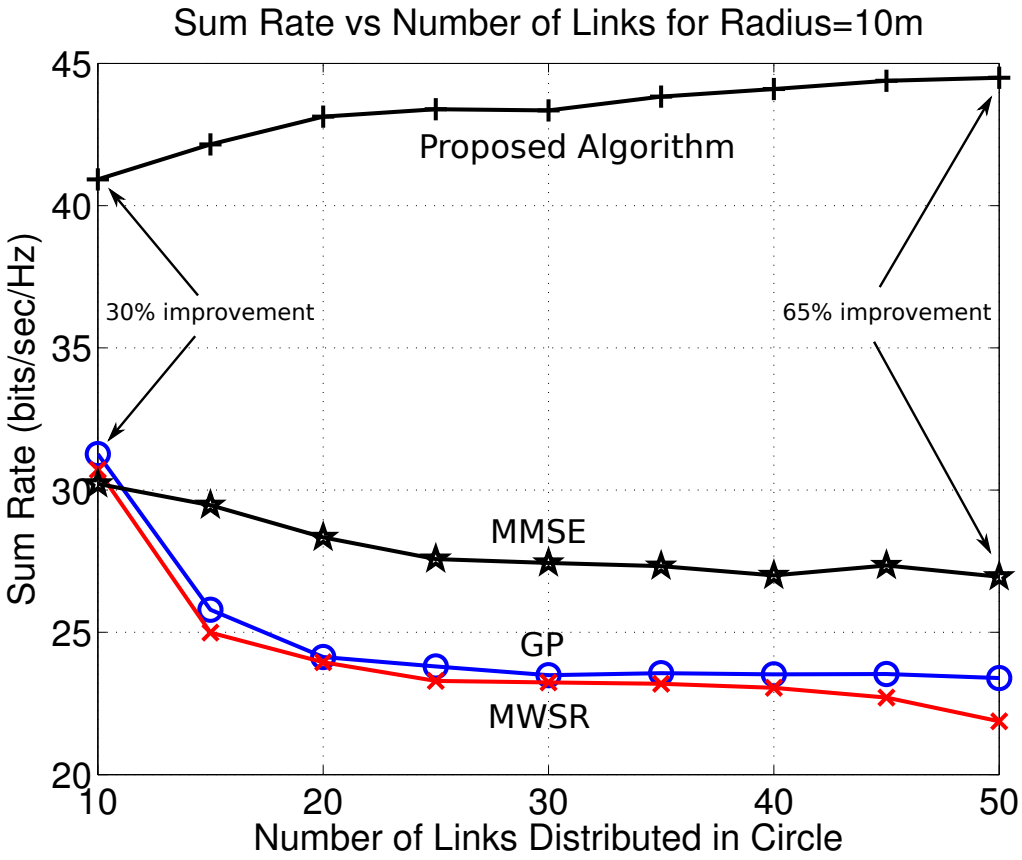

Figure 4.5: Sum rate as a function of the number of links placed within the circle when the radius of the circle is fixed to ten meters. All links placed within the circle are considered by the algorithms, but an algorithm may choose to deactivate some links.

increasing the number of antenna elements at every node increases the degrees of freedom available to support more links. For example, when the number of antenna elements at every node is increased from 4 to 8 and the number of links is high (50 links), the proposed algorithm achieves a sum rate that is 73% (as opposed to 90% when every node has 4 antennas) higher than that of the GP and the MWSR. However, when the number of antenna elements at every node is increased from 4 to 8 and the number of links is low (10 links), the proposed algorithm no longer has a 30% performance improvement over the other algorithms and instead, all algorithms except for the MMSE algorithm achieve comparable performance. The MMSE algorithm achieved a lower sum rate than all other algorithms when all nodes are equipped with

8 antenna elements.

4.4.3 Complexity

We now compare the complexity of our proposed algorithm in terms of the CPU running time for the experiment of Section 4.4.2 in which we fix the radius of the circle to ten meters and vary the number of links placed within this circle. Figure 4.6 shows the average CPU time by the algorithms as implemented in MATLAB and run on a i7-2700K Intel CPU rated at 3.5GHz. These results show that the running time of the GP and the MWSR algorithms increases greatly as the number of active links increases. The running times of the MMSE algorithm and the proposed algorithm, however, increases only slightly as the number of active links increases. The MMSE algorithm has the shortest running time of all algorithms and the running time of our proposed algorithm was consistently about four times that of the MMSE algorithm.

We also analyze the complexity of the proposed algorithm in terms of sum rate versus number of iterations. Figure 4.7 shows the sum rate achieved by the proposed algorithm as a function of the number of iterations for two trials of the experiment of Section 4.4.2 for the cases when $M = 10$ and $M = 50$ links are placed within a circle whose radius is ten meters. The results show that, for Trial 1, $M = 10$, the proposed algorithm is within 6% and 1% of its final sum rate after 10 and 20 iterations, respectively. For Trial 2, $M = 10$, the proposed algorithm requires slightly more iterations to reach a high-sum-rate solution; however, after 25 iterations, the proposed algorithm is already within 2% of the final sum-rate. Similar results can be shown when the number of links is increased to $M = 50$, with the exception that more iterations are required to reach a high-sum-rate solution. For Trial 1, $M = 50$ and Trial 2, $M = 50$, the sum rate after 25 iterations is within 6% and 10% of the final sum-rate, respectively.

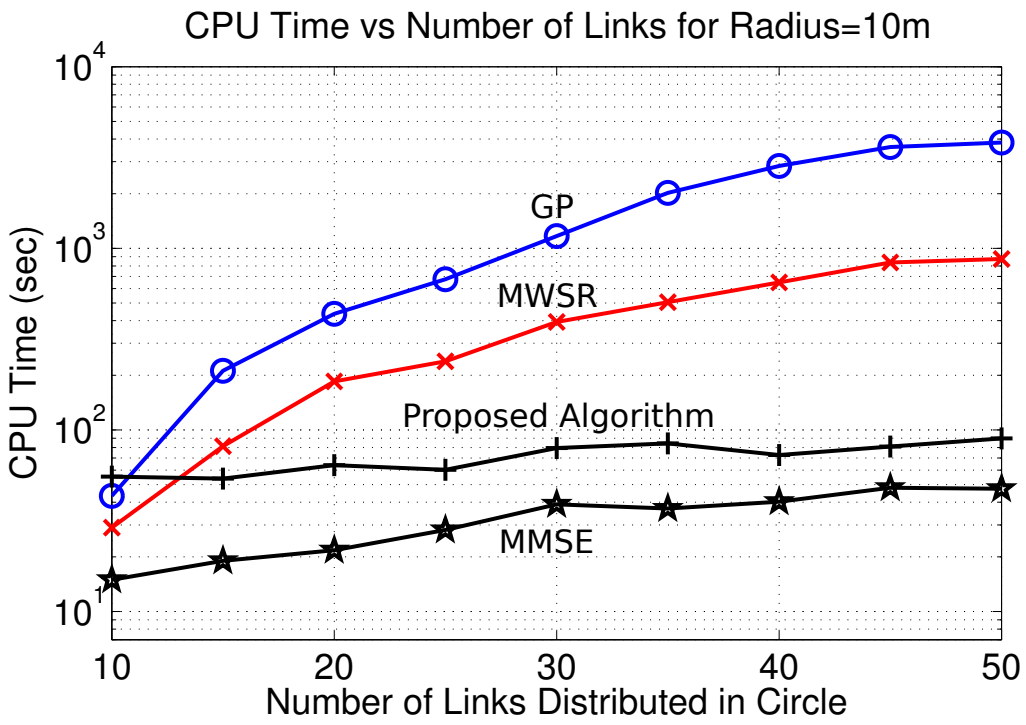

Figure 4.6: Average CPU time as a function of the number of links placed within the circle when the radius of the circle is fixed to ten meters. All links placed within the circle are considered by the algorithms, but an algorithm may choose to deactivate some links.

In terms of complexity per iteration of a particular link, we perform the following analysis. For simplicity of notation, we drop the subscript k and assume that each transmitter is equipped with n_t transmit antennas and that each receiver is equipped with n_r receive antennas. For each iteration, the proposed algorithm requires that each receiver computes $\boldsymbol{R}$, $\boldsymbol{W}$, and $\boldsymbol{U}^{\mathrm{MMSE}}$. Computing $\boldsymbol{R}$ requires M matrix additions of size $n_r \times n_r$, which is $O(Mn_r^2)$. Computing $\boldsymbol{W}$ requires computing $\boldsymbol{R}^{-1/2}$ (an inverse and a square root) and the singular values of the $n_r \times d$ matrix in (4.14). Assuming that computing the SVD of an $m \times n$ matrix is $O(m^2n + n^3)$ and that computing the square root and inverse of an $n \times n$ matrix is $O(n^3)$, then computing $\boldsymbol{W}$ is $O(n_r^2 d + d^3 + n_r^3)$. Computing $\boldsymbol{U}^{\mathrm{MMSE}}$ requires computing the inverse of an $n_r \times n_r$ matrix, which is $O(n_r^3)$. At the transmitter side, computing $\boldsymbol{V}$ requires a search for μ as well as computing $\boldsymbol{P}^{-1/2}$ and computing the SVD of the

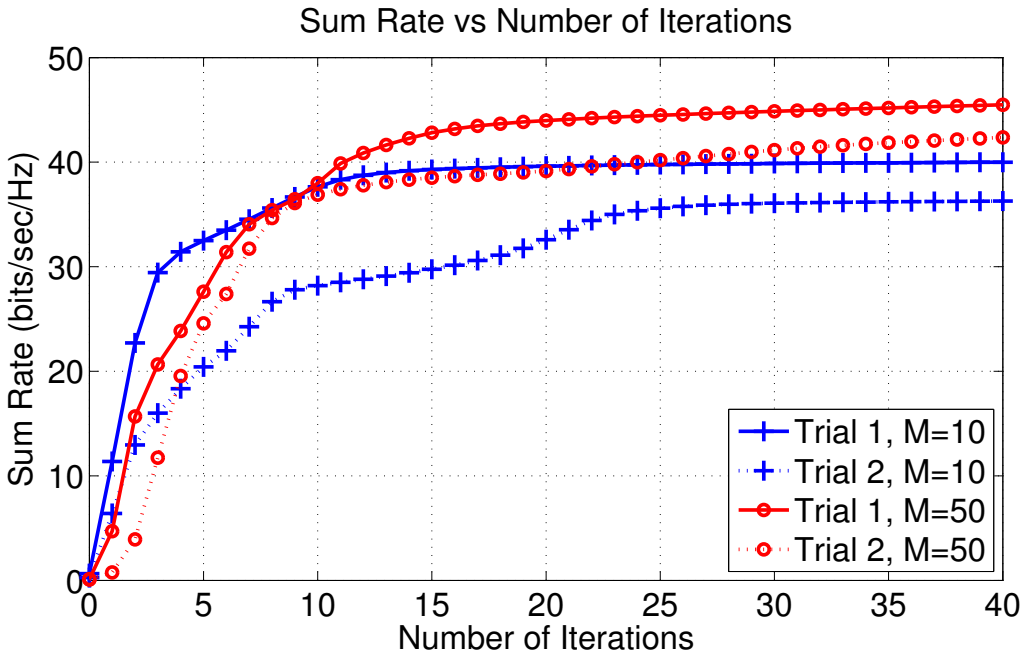

Figure 4.7: Sum rate achieved by proposed algorithm as a function of the number of iterations for four different trials.

$n_r \times n_t$ matrix in (4.3). Computing $\boldsymbol{P}^{-1/2}$ requires roughly M matrix additions of size $n_t \times n_t$ and a square root and inverse operation. The overhead of searching for μ can be reduced by searching around the value of μ obtained from the previous iteration. Ignoring the search for μ, computing $\boldsymbol{P}^{-1/2}$ is $O(Mn_t^2 + n_t^3)$ and the SVD operation in (4.3) is $O(n_t^2 n_r + n_r^3)$. Since $n_r \geq d$, the complexity of the proposed algorithm per iteration of the beamforming and combining weights for all links becomes $O(M^2 n_r^2 + Mn_r^3 + M^2 n_t^2 + Mn_t^3 + Mn_t^2 n_r)$. As the number of links M becomes large, the complexity of the algorithm becomes $O(M^2)$, which is comparable to that of previously reported algorithms [72, 78, 88, 90].

4.5 Chapter Summary

In this chapter, we have presented an algorithm for maximizing the sum rate of a set of interfering MIMO links. Our algorithm jointly optimizes which link should be

active, the number of streams (if any) on each link, and their corresponding beamforming and combining weights. Our simulation results showed that, in terms of sum rate, the proposed algorithm is able to achieve higher sum rate at high interference and comparable sum rate at medium and low interference than previously reported algorithms. Also, our simulation results showed that at high interference, the sum rate of our proposed algorithm increases as the number of links increases, because the proposed algorithm can deactivate links and has diversity on which links to deactivate. Finally, our results showed that the proposed algorithm also has lower time complexity than most algorithms tested at high interference with large number of links.

CHAPTER 5

FRAMEWORK FOR OPTIMIZING THE PERFORMANCE OF DENSE NETWORKS WITH MIMO LINKS

In this chapter, we investigate a number of practical issues that arise when attempting to implement any high-performing algorithm for computing the beamforming and combining weights (such as the ones presented in [4, 14, 20, 25, 33, 43, 72, 78, 83, 108] and the one presented in Chapter 4) into an unplanned and/or dense single-hop network, such as a WLAN or femtocell, where each single-hop network is composed of an access point (AP) serving several associated clients. An example topology is given in Figure 5.1

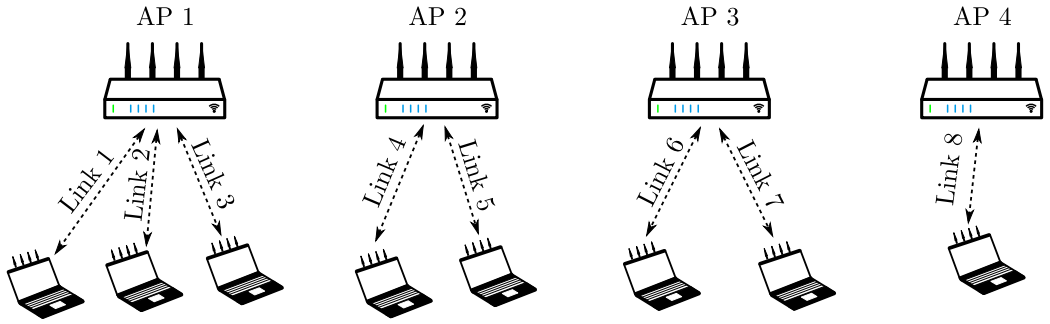

Figure 5.1: Example of four single-hop wireless networks.

We focus on the following issues:

- reducing the overheads in measuring and communicating the channel state information (CSI) required for computing the beamforming and combining weights;
- reducing the overhead of computing the beamforming and combining weights; and

- overcoming the overheads related to cooperation.

Based on our proposed solutions, we design a framework for taking advantage of the spatial-multiplexing and interference-suppression capabilities of MIMO links. We have implemented the MIMO framework using the ns-3 network simulator [102]. The implementation accounts for all overheads (e.g., CSI measurement and feedback, beamforming and combining weight computation and distribution, etc.). Performance evaluation in ns-3 show that our proposed solutions can significantly reduce the overhead of collecting CSI with negligible loss in performance. We show that with spatial multiplexing only, our framework has comparable performance to that of 802.11n. Later, in Chapter 6, we will also consider interference suppression within our framework and present several scheduling approaches for our framework that can greatly improve the performance of single-hop wireless networks as compared against that of 802.11n.

This chapter is organized as follows. In Section 5.1, we discuss the issues with taking advantage of the interference suppression capabilities of MIMO links, and discuss our proposed solutions. In Section 5.2, we describe our framework that implements our solutions. In Section 5.3, we discuss the issue of synchronization of transmitters to perform interference suppression. In Section 5.5, we present simulation results and analysis. Finally, in Section 5.6, we provide a brief summary of this chapter.

5.1 Core Issues of Performing MIMO Spatial Multiplexing and Interference Suppression

Although MIMO links can potentially achieve large gains in terms of sum rate by performing a combination of spatial multiplexing and interference suppression, making use of the capabilities of MIMO links in real network scenarios is made difficult by several core issues. In the following sections, we discuss three main issues and propose several techniques for addressing them.

5.1.1 Computation Overhead of Beamforming and Combining Weights

At the core of most of these issues is the interdependency between the beamforming and combining weights. Due to this interdependency, methods for computing these weights that achieve the best performance are iterative and often require many iterations and packet exchanges between participating nodes. Although non-iterative methods exist, we showed in Section 3.3.3 that iterative algorithms achieve 70% higher sum rate than these non-iterative methods in high-interference scenarios for eight-link networks. Instead of using a non-iterative algorithm with low performance, we propose to use an iterative algorithm with high performance.

To reduce the overhead incurred while computing weights for a given set of links, we propose to compute all weights at a single node, either at a designated AP or at a centralized network controller, and have this node collect all CSI from the participating links. In this dissertation, we will focus on the case that the CSI is collected by an AP, which we will refer to as the *Worker AP*. However, our general approach could just as easily be implemented using a centralized network controller. Collecting CSI at a single node eliminates the overhead of nodes exchanging beamforming and combining weights at each iteration of the algorithm, as would be required in a distributed execution of the algorithm. This approach works well for the scenarios targeted herein, i.e. with two to six APs operating on one channel. As discussed earlier, this covers many cases that commonly occur in practice.

Another issue that arises because of the interdependency between the beamforming and combining weights is that the beamforming and combining weights computed for a given link set will be different from those computed for other link sets, even if the difference between the link sets is just a single node. To avoid the overhead of having to compute weights for all possible link sets, we propose to reuse, as much as possible, link sets for which beamforming and combining weights have already been computed. When a link set is reused, the beamforming and combining weights

are already available, thereby avoiding the relatively expensive weight calculation operation.

To further reduce the impact of computing the beamforming and combining weights, we propose to compute weights in the background and use only those link sets for which weights have already been computed until new weights become available. Additionally, we initialize the candidate link sets with the set of *single-link link sets*, in which each link gets a dedicated link set so that it can use the channel free of interference. In the absence of interference, the optimal beamforming and combining weights are easily computed through singular-value decomposition (SVD) and water-filling [101] (see also Chapter 2.2.4). In our approach, we will use only the subset of link sets that maximize the performance. Therefore, although the single-link link sets will be available, they will be used less frequently as weights for new link sets are computed.

5.1.2 CSI Measurement Overhead

To optimize performance for a given set of interfering MIMO links, the nodes must collect all CSI. This includes the channels between every transmitter and their corresponding receivers and between every transmitter and all other receivers with which they interfere. If any combination of links (one link per AP) is allowed to be selected at a given time, the channels between every pair of interfering nodes must be measured. If we have A APs and C clients per AP and every node is within the communication range of every other node, then we have $A + AC$ nodes and approximately $(A + AC)^2 = A^2 + 2AC + (AC)^2$ measurements to collect. This means that with $A = 5$ APs and $C = 10$ clients per AP, more than $2,500$ CSI measurements are needed. This produces a huge amount of CSI that must be communicated between nodes in the network.

We propose a novel approach, called *Consistent AP Orientation with Channel*

Symmetrization, or CAPOCS, to reduce the number of channel measurements and CSI communication overhead. To understand our approach, note that the number of channel measurements is dominated by the $(AC)^2$ term. Here, AC is the number of clients and the $(AC)^2$ term corresponds to each client measuring its channel with every other client. Since channel measurements are only needed on actual links and between nodes that interfere, which are transmitter-receiver pairs, we propose to coordinate active APs in a slot so that they are either all transmitting (downlink direction) or all receiving (uplink direction). Thus, we prevent one AP from activating a downlink and another AP from activating an uplink in the same time slot. In this situation, clients do not interfere with each other, because they are consistently receivers or transmitters at one time. As a result, clients do not need to measure channels with other clients and the dominant term in the channel measurements expression disappears. Similarly, APs do not need to measure channels with other APs. What is left is to measure channels for every AP-client link, which requires A^2C measurements. In the above example, this produces more than an order of magnitude reduction in the number of channel measurements, from more than $2,500$ down to 250.

5.1.3 Cooperation Overhead

To overcome the overhead of coordinating the participating nodes, we propose to have transmitters aggregate as many packets as can fit within a fixed duration τ_{data} and have the receivers simultaneously acknowledge the packets using a BlockAck. This differs from the aggregate-packet mechanism in 802.11n in that the packets in 802.11n are aggregated up to a maximum size [50], whereas we propose to aggregate packets up to a maximum duration. Note that by fixing the data-transmission duration, we also account for the fact that each active link can have different data rates.

To avoid having to compute and distribute beamforming and combining weights

for communicating the BlockAcks, we propose to have the destination nodes use the vector that achieves the highest gain from their combining weights for the current link set, normalized to maximize the transmit power, as their beamforming weights for sending the BlockAck. Similarly, we propose to have the source nodes use the vector that corresponds to the highest gain from their beamforming weights for the current link set as their combining weights for receiving the BlockAcks. This technique of reversing the roles, but reusing the weights, is commonly used to aid the computation of the transmitter weights [25, 43, 78, 82] (see also Section 3.1.3). Although this technique is suboptimal, we expect it to be sufficient since BlockAcks are sent using a lower data rate, which also has a lower SINR requirement, than data packets.

5.2 Framework Description

In this section we present a framework in which we implement our ideas to realize spatial multiplexing and interference suppression within dense wireless networks. We reiterate that, although this presentation assumes a designated AP collects CSI and computes the beamforming and combining weights, these functions could also be implemented on a centralized network controller.

A flow chart of the MIMO framework is shown in Figure 5.2. In the first step, the APs discover each other to choose the Worker AP. During this step, any leader election protocol can be used for choosing the Worker AP (e.g., [13, 68, 77, 103, 104]). In the second step, the Worker AP requests CSI from all APs. During this step, each AP takes a turn asking each of its associated clients to transmit a sounding packet, and each AP listens for sounding packets and records the CSI for each. By taking turns, APs avoid interference within measurements. Once all the CSI is measured at the APs, the APs forward their measurements to the Worker AP. The Worker AP then constructs and distributes the first schedule using the single-link link sets. Note that the beamforming and combining weights for the single-link link sets do not

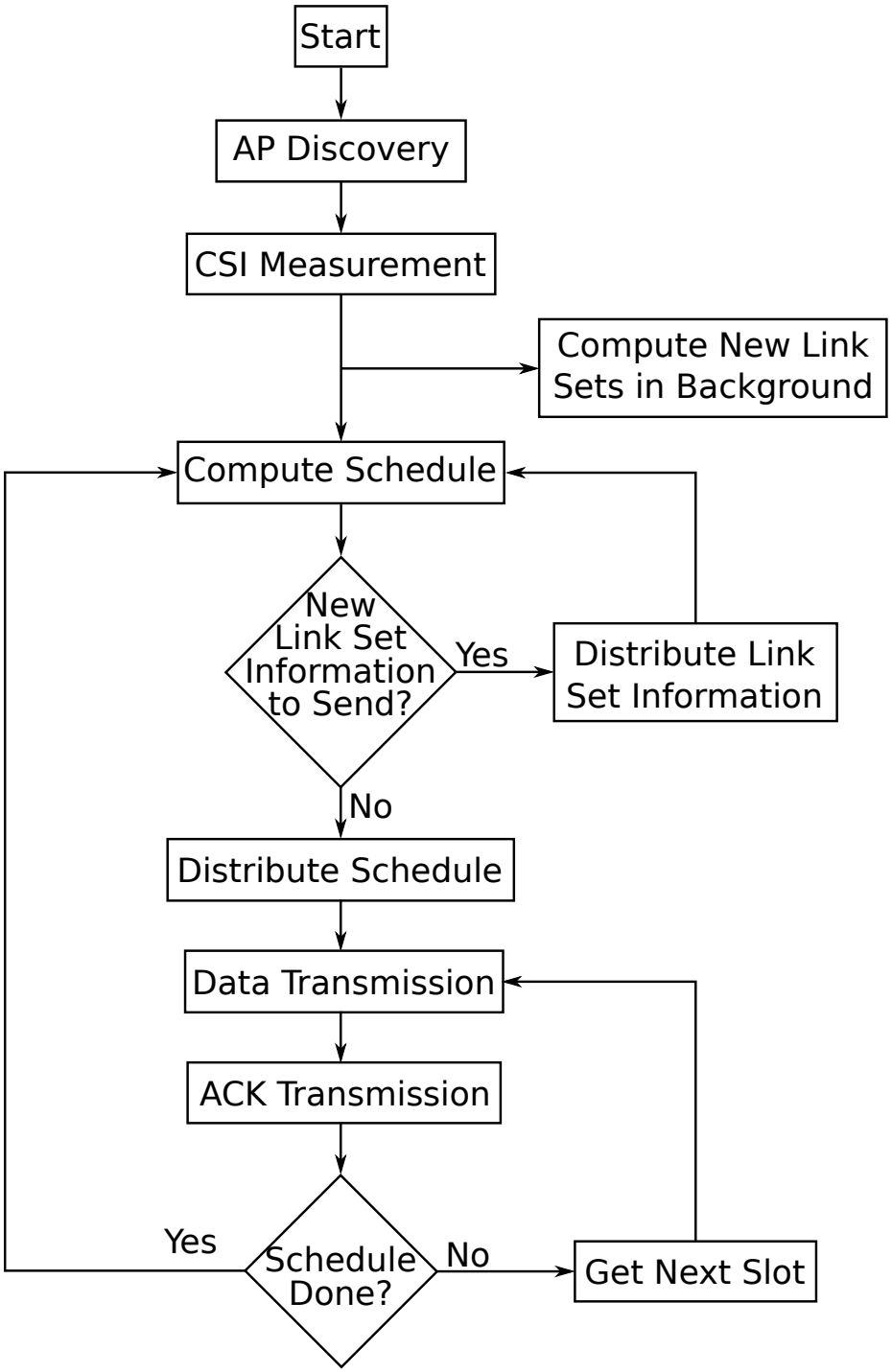

Figure 5.2: High-level flowchart of framework.

need to be distributed since they are easily computed locally by the corresponding nodes. In the background, the Worker AP begins computing the link sets and the corresponding beamforming and combining weights sequentially for all subsequent link sets once all CSI is received.

Once each node has the schedule, the transmitters of the first time slot transmit for a duration equal to τ_{data}. Then, the receivers wait for a short time before sending their BlockAcks in the reverse channel. We label the total time allocated for the BlockAck as τ_{ack}, so that the duration of a time slot is $\tau_{\text{slot}} = \tau_{\text{data}} + \tau_{\text{ack}}$. This process repeats for every time slot in the schedule.

Upon termination of the last time slot, the Worker AP regains control and computes a new schedule by considering all link sets for which beamforming and combining weights have been computed. The Worker AP then distributes the weights for all link sets that were selected in the schedule that have not already been distributed. Finally, the Worker AP distributes the desired schedule and all nodes follow the new schedule until completion, and then the process repeats.

5.3 Synchronization

The beamforming and combining weight strategies of [4, 14, 20, 25, 33, 43, 72, 78, 83, 108], the one presented in Chapter 4, and even the SVD strategy discussed in Chapter 2.2.4 were designed under the assumption that the channels are narrowband. To apply these strategies to wideband channels, a technique known as orthogonal frequency-division multiplexing (OFDM) is used when deploying MIMO [18, 97].

With OFDM, a wideband channel can be transformed into a large number of closely spaced orthogonal narrowband sub-carriers. To achieve this, OFDM modulator divides the incoming data into blocks of N symbols and computes the inverse fast Fourier transform (iFFT) of each block individually, which produces N modulated symbols for each block. The transmitter then inserts a cyclic prefix (CP) or guard

interval to each block and sends each symbol sequentially. The CP is inserted to each OFDM block to eliminate inter-symbol-interference between blocks, which is caused by multi-path copies of the signal. The number of symbols used as CP must have a duration longer than the delay spread of the multi-path components. Finally, at the receiver side, the receiver removes the CP and demodulates each block using a fast Fourier transform (FFT) to obtain the desired symbols.

To take advantage of the interference-suppression capability of MIMO links, both the desired signal and the interference must arrive at all corresponding receivers within the CP [100]. This can be achieved if the transmitters synchronize their transmissions. Until now, we have assumed that this synchronization is achieved by a TDMA multiple access scheme. However, notice that the only phases of the proposed framework that requires synchronization are the data-transmission phase and BlockAck-transmission phase. Therefore, the proposed strategy can also be implemented using CSMA/CA with minimum modification if the data transmissions and BlockAck transmissions can be synchronized. Next, we present a strategy that can be used to synchronize the data-transmission phase in our framework based on CSMA/CA.

The basic idea of our proposed synchronization technique is to synchronize on carrier sense. To begin, the APs loosely synchronize by agreeing when to begin carrier sensing. To prevent neighboring nodes from trying to access the channel during this carrier sensing phase, APs can broadcast beacons initiating a contention-free period using the point coordination function (PCF) of 802.11 [50]. Once this period begins, a designated transmitter can begin communicating its data to its desired receiver. Upon sensing the channel busy, the remaining transmitters begin their own transmissions, thereby achieving a tighter synchronization.

To analyse the feasibility of this synchronization strategy, let us analyse a worst case scenario using the timing parameters from 802.11n [50], which are listed in Table

5.1. Initially, the transmitter that triggers the tight synchronization begins its transmission. This signal takes a time equal to τ_{prop} to propagate and reach its desired receiver and the next transmitter. The next transmitter must then sense the channel busy, which can take a time τ_{cca}. This transmitter then takes a time τ_{RxTx} to switch from receiving to transmitting. Finally, the new signal propagates to the medium, reaching its desired and unintended receivers, which take another time τ_{prop}. The total time required for this one-hop is $\tau_{\text{err1hop}} = \tau_{\text{cca}} + \tau_{\text{RxTx}} + 2\tau_{\text{prop}}$. If we assume a path-loss exponent of three, we can estimate the maximum propagation delay to be $\tau_{\text{prop}} \approx 0.73$ μs, corresponding to a distance of about 220 meters, for practical values of transmit power and carrier sense threshold. Therefore, for each hop, the timing error can be as large as $\tau_{\text{err1hop}} = 7.5$ μs, and so the CP must compensate for this error.

Table 5.1: Timing parameters for 802.11n.

Parameter	Value
τ_{cca}	< 4 μs
τ_{RxTx}	< 2 μs
τ_{prop}	$\ll 1$ μs

If we add 800 ns to account for the typical indoor delay spread [50] and assume that we want to synchronize with transmitters that can be as far away as two hops, then we need CP to compensate for $\tau_{\text{err}} = 15.8$ μs of synchronization error. This can easily be accomplished by scaling the OFDM block size N and the CP until the duration of the CP is greater than τ_{err} [64, 100]. By scaling both N and CP by the same factor, the ratio of CP symbols to usable data stays the same (about 0.25 for 802.11 [50]), thereby incurring no overhead [100]. With an OFDM block size of $N = 1264$ and a bandwidth of 20 MHz, the CP has exactly the desired duration of 15.8 μs.

To synchronize the transmission of the BlockAcks, a similar strategy can be implemented by having the transmitters initiate another synchronize-on-carrier-sense phase a short duration after the end of the allocated time for data transmission.

The synchronization mechanism above works only for dense networks of up to two carrier-sense hops. For interfering networks that span more than two carrier-sense hops, the CP and OFDM block size N will need to be scaled accordingly. For larger networks, another synchronization strategy might be needed, such as using the network time protocol (NTP) as done in [62] or simply implementing the framework on top of a TDMA multiple access scheme.

In the next section, we describe how the link sets are computed and we define several scheduling algorithms that can be integrated with this framework.

5.4 Extending the Algorithm in Chapter 4 To Consider Links that Share a Node

Given a set of links where no two links share a node, the MIMO weight algorithm proposed in Chapter 4 achieves a local maximum of the sum rate by optimizing which links should be active, the number of streams on each active link, and the corresponding MIMO weights that support those streams. This algorithm first initializes the beamforming weights. Then, the algorithm iteratively computes the combining weights, followed by the beamforming weights.

In single-hop wireless networks, links can share a node (e.g., links 1, 2, and 3 share a AP 1 in Figure 5.1). Using the analysis of [20], we generalize the algorithm from Chapter 4 to also consider links that share a node. The modifications can be summarized as follows. We compute a pair of beamforming and combining weights for each link, even if the link shares a node with another link. For the case that an AP is associated to multiple nodes and is acting as a receiver, we compute the combining weights for each link independently, so that interference between its clients is treated as noise. For the case that this AP is acting as a transmitter, we compute

the beamforming weights as detailed in Chapter 4, except that the variables μ_k in the virtual whitening component of all beamforming weights k for this AP must be equal and are chosen such that the total transmit power through all beamforming weights satisfy the links' transmit power constraint (similarly to [20, 90]). With this modification, all links can serve as input to the algorithm so that the algorithm can determine the best subset of links that maximize the sum rate. We further modify the algorithm so that the final link set contains at most one link for each AP (for fair comparison against 802.11n). To achieve this, midway and at the end of the iterations, we remove the worst performing links that share a node so that they are no longer considered for the final link set.

5.5 Simulation Results

In this section, we present numerical results to quantify the benefits of using CAPOCS. Also, we show results on the performance of the proposed framework with no interference suppression and compare it with that of 802.11n.

The rest of this section is organized as follows. In Section 5.5.1, we discuss the ns-3 simulation setup, which we will reuse in Chapter 6. In Section 5.5.2, we quantify the overhead avoided by using CAPOCS and also quantify the performance loss of using CAPOCS. Finally, in Section 5.5.3, we perform an initial evaluation of our framework to show that the framework with no interference suppression suffers no significant penalty in performance as compared to 802.11n.

5.5.1 Simulation Setup

We implement the MIMO framework in the ns-3 simulator [102] and compare against 802.11n, which performs spatial multiplexing only. We have added support for the physical-layer MIMO model described in Chapter 2.3.1. To simulate 802.11n, we have modified the basic 802.11 protocol within the ns-3 simulator to support 802.11n capabilities such as Greenfield preamble, support for sounding packets, and support

for MIMO spatial multiplexing.

In the simulator, a transmitter chooses the highest data rate for which the bit-error rate is less than 10^{-6} based on the expected signal to interference-plus-noise ratio (SINR). Also, packets are decoded on a stream-by-stream basis and a packet is received successfully only if all streams are decoded successfully. Table 5.2 shows the data rates and their corresponding SINRs as reported by ns-3.

Table 5.2: Data rates and their SINR threshold for a BER of 1×10^{-6} as reported by ns-3.

Data Rate (Mbps)	SINR
6	2.46851
9	4.80368
12	4.93702
18	9.60737
24	22.2137
36	45.4008
48	135.384
54	181.051

The 802.11n strategy always tries to maximize the rate on its link by computing the optimal SVD weights, as described earlier in Section 2.2.4. In our simulations of 802.11n, we enable the aggregate MAC service data unit (A-MSDU) support in ns-3, which enables aggregate packets of up to 7935 bytes and enables their appropriate BlockAcks [50].

We account for the overhead of computing the MIMO weights within our simulation by measuring the CPU time consumed by the weight computation function and scaling it so as to estimate the running time of a multi-core capable AP running at 3 GHz. Also, we limit the number of iterations executed by the MIMO weight algorithm to ten.

For the simulation of the MIMO framework, we account for the overhead of 30 OFDM subcarriers at all times. Additionally, we assume that CSI and MIMO weights

are transmitted uncompressed when collecting CSI and when distributing the MIMO weights, respectively. This tends to overestimate the overhead of our approach for exchanging CSI and MIMO weights.

For all simulations, we assume a flat-fading Rayleigh MIMO channel [37], and assume that every wireless node has four antenna elements. We assume a path-loss exponent of three. Also, we assume that all radios operate using a 20 MHz bandwidth and that the carrier frequency is 5 GHz. Unless otherwise stated, the MIMO channels are fixed throughout the duration of the simulation. Also, we set the packet generation rate high enough so that each AP has data to send to each of its clients and each client has data to send to its AP. For the simulations of the MIMO framework, the Worker AP is randomly chosen among the APs. We assume that the fraction of downlink traffic to the total traffic is $p_{\text{downlink}} = 0.6$. Finally, we set the schedule duration to $\tau_{\text{schedule}} = 3$ secs, the data duration to $\tau_{\text{data}} = 10$ ms, and acknowledgment duration to $\tau_{ack} = 210$ μs.

5.5.2 Evaluation of CAPOCS

We begin by comparing the overhead of collecting the CSI at a single node for the case in which APs can arbitrarily become transmitters or receivers (requiring almost all CSI) and for the case that CAPOCS is used. We simulate four APs in a line at 50 m intervals. Each AP has C associated clients that are uniformly distributed within a radius of 80 meters from the AP.

Figure 5.3 shows the time to collect the relevant CSI as a function of the number of clients per AP (C). It is clear from Figure 5.3 that as the number of clients increases, collecting all CSI results in a rapid increase in collection time, whereas collecting CSI as required by CAPOCS produces only a linear time increase. At the highest C tested, the time to collect all CSI is at least eight times greater than for collecting only all uplink or all downlink CSI. This extra overhead can lead to low performance,

especially if the channels must be remeasured periodically.

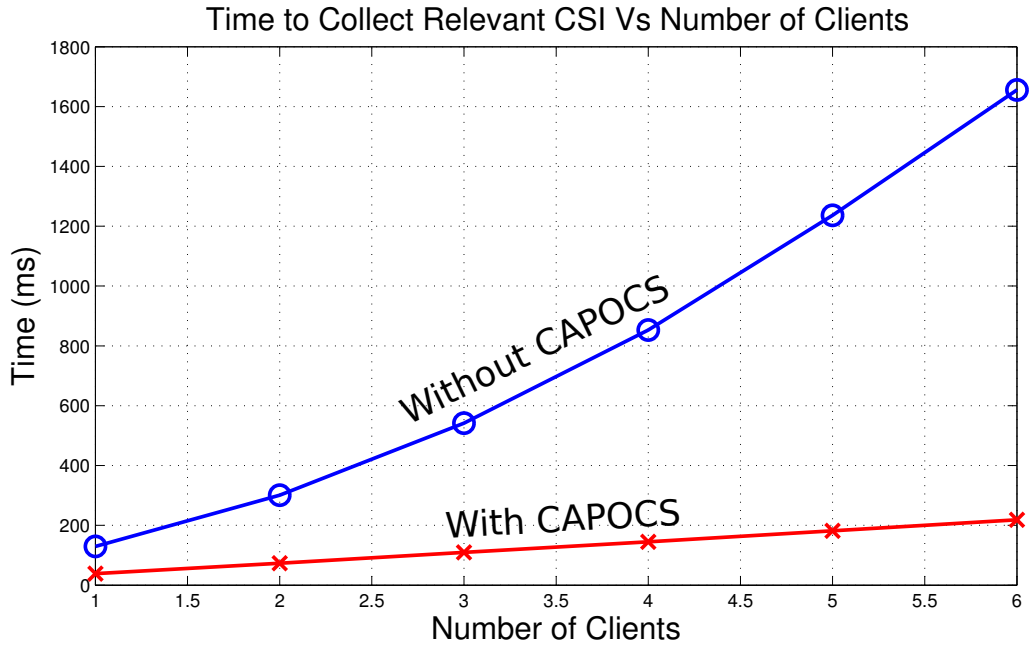

Figure 5.3: Comparison of time to collect the required CSI as a function of the number of clients.

We also perform a different simulation to quantify the potential performance loss of using CAPOCS to coordinate the APs. In this experiment, we place a varying number of APs uniformly within a circle of 50 meters. For each AP, we place four associated clients that are uniformly distributed within 80 meters of that AP. For each random network and all possibilities of AP roles (either transmitter or receiver for each AP), we compute the capacity using the MIMO weights computed by the modified algorithm for determining the beamforming and combining weights described in Section 5.4. For each random network, we test every possibility of AP roles and save the minimum, maximum, and average capacity. When CAPOCS is considered, only two AP roles are considered: the case in which all APs are transmitters and the case in which all APs are receivers. The results, averaged over 100 random networks, are shown in Figure 5.4.

In Figure 5.4, it can be observed that the penalty in terms of capacity is negligible

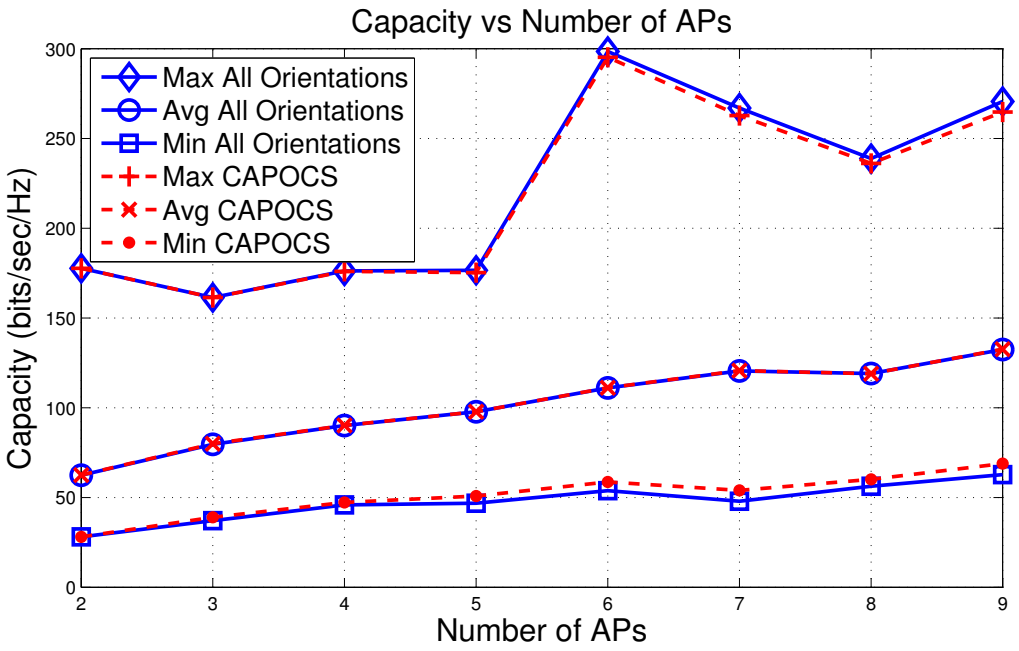

Figure 5.4: Minimum, maximum, and average sum capacity as a function of the number of APs for all possibilities of AP orientations and when CAPOCS is used.

on the average case. In the maximum case, CAPOCS experiences a maximum loss of approximately 2% when the number of APs is large. In the minimum case, however, CAPOCS achieved a higher capacity on the average. This last result is consistent since the minimum of a superset will be smaller or equal than the minimum of a subset.

5.5.3 Evaluation of MIMO Framework with Spatial Multiplexing Only

In this section, we compare the performance of our framework against that of 802.11n for the case that no interference suppression is performed (only spatial multiplexing is considered for both mechanisms). We will show that, for this scenario, the two mechanisms have comparable performance, and so both mechanisms encounter similar overheads. Later, in Chapter 6, we will consider the case that interference suppression is performed within our framework and so the majority of the gains that we observe are due to the interference suppression capability of MIMO links and not by the framework itself.

The simulation setup is as follows. We simulate four APs in a line, placed every 50 m. We uniformly distribute two clients per AP within an 80 m radius of their associated AP. We vary the aggregate-packet duration τ_{data}, and perform 100 trials for each. For this simulation, we set the simulation time to 15 seconds.

Figure 5.5 shows the average sum goodput as a function of τ_{data} for our framework. Figure 5.5 also shows the average sum goodput achieved by 802.11n, which does not depend on the aggregate-packet duration τ_{data}. The data on Figure 5.5 shows that the proposed MIMO framework achieves, on the average, 10.6% improvement on the sum goodput as compared to 802.11n. When the aggregate-packet duration is $\tau_{\text{data}} = 10$ ms, the MIMO framework achieved a goodput that is 12% better than that of the 802.11n.

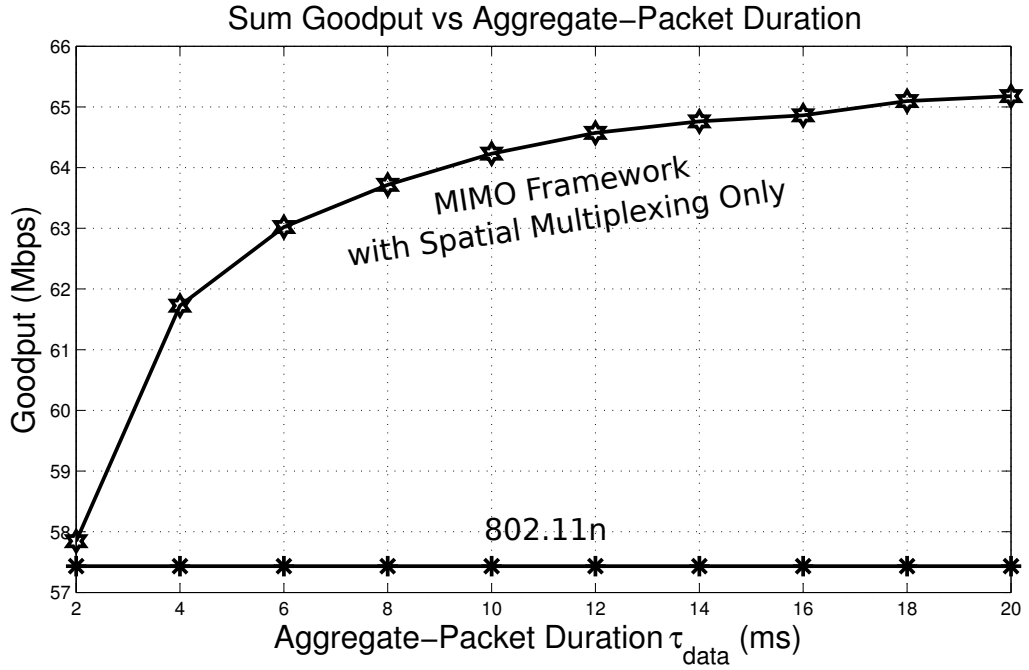

Figure 5.5: Sum goodput as a function of the aggregate-packet duration τ_{data}.

Next, in Chapter 6, we will describe several methods for computing sets of beamforming and combining weights that perform interference suppression and how they can be used to build a fair scheduler that maximizes the performance. We will show that the proposed framework with interference suppression far outperforms 802.11n.

5.6 Chapter Summary

We proposed a MIMO framework capable of implementing state-of-the-art MIMO interference suppression techniques. The proposed framework implements our ideas on how to overcome practical issues that arise when applying state-of-the-art MIMO interference suppression techniques to dense wireless networks. To reduce the overhead associated with computing the MIMO weights, we collect all CSI at a single node, perform the MIMO weight calculations within that node, and reuse the calculated MIMO weights whenever possible. To reduce the number of channel measurements that must be collected, we proposed CAPOCS, which forces all APs to operate as either transmitters or receivers, at a given time. Simulation results demonstrate that CAPOCS has negligible effect on the performance of the MIMO weight algorithm while significantly reducing the time required to collect the necessary CSI. Simulation results show that our framework with no interference suppression can achieve comparable performance to that of 802.11n. We will show in the following chapter that when interference suppression is used in conjunction to spatial multiplexing, our framework achieves considerably higher performance than 802.11n.

CHAPTER 6

SCHEDULING TO ACHIEVE HIGH PERFORMANCE AND MAINTAIN FAIRNESS IN DENSE NETWORKS

Traditionally, scheduling is done in such a way as to avoid or reduce interference. With MIMO links, some interference can be suppressed by the corresponding beamforming and combining weights. However, due to the limited degrees-of-freedom associated with the multiple antennas, not all interference can be suppressed. Determining which interference can be suppressed so as to determine which links can be scheduled together is difficult due to the interdependency between the beamforming and combining weights. Testing whether a link is compatible with a given set of links would require that the beamforming and combining weights be recomputed for the new link set, which is an expensive operation. Additionally, trying to achieve a certain fairness criteria is made difficult by the fact that the performance of a given link set is unknown until the beamforming and combining weights are computed.

In this chapter, we take the following approach to achieving high performance and fairness through scheduling. We will use the MIMO weight algorithm from Chapter 4 to compute various candidate link sets and their corresponding MIMO weights. Once the link sets and corresponding MIMO weights are computed, we will identify the number of times that each candidate link set should be used (if any) so as to meet the desired fairness criteria while also achieving high performance.

Performance evaluations in ns-3 demonstrate that spatial multiplexing and interference suppression with MIMO links is practical among a small number of competing networks. Our results show that, with four APs, two clients per AP, and four antenna elements per node, the average aggregate goodput of the MIMO framework is

increased by 170% with fair scheduling and 200% without fair scheduling, as compared to the goodput of 802.11n (which performs spatial multiplexing only). Simulation results of up to six clients per AP show that our framework can achieve a sum goodput that is 2.8 and 2.6 times that of the 802.11n with spatial multiplexing only for the cases of unfair scheduling and fair scheduling, respectively. Our results also show that even if channel conditions change by about 20% during a 15 second interval, the MIMO framework still achieves a goodput that is 130% better with fair scheduling and 160% better without fair scheduling than 802.11n.

In the following, we will focus our discussion on a given set of M half-duplex links, composed of links formed by the APs and their associated clients. Note that because each AP can have multiple associated clients, it is possible that some of the M links share a node. To simplify the discussion, we assume that CAPOCS is used, so that all M links are either downlinks or uplinks. The following discussion applies to each case separately. As mentioned in Chapter 5, our framework initializes the candidate link sets to the set of single-link link sets, where each link is scheduled separately from the other link and uses the optimal SVD weights (as described in Chapter 2.2.4). In Section 6.1, we provide details on how we use the algorithm from Chapter 4 to compute better candidate link sets than the single-link link sets. In Section 6.2 and Section 6.3, we present three scheduling algorithms that determine how to use the candidate link sets to achieve different fairness criteria. In Section 6.4, we provide ns-3 simulation results for the various scheduling algorithms. Finally, in Section 6.5, we provide a summary of this chapter.

6.1 Generating Better Candidate Link Sets

We will use the modified algorithm for computing the beamforming and combining weights as described in Chapter 5.4. We further modify the algorithm by replacing (4.15) with $\boldsymbol{W}_k = \omega_k \left(\boldsymbol{I} + \boldsymbol{S}_k^2\right)$, so that the resulting algorithm can maximize a

weighted sum rate, where ω_k is the *link weight* assigned to link k, as described in [20, 72]. We will adjust these link weights to generate various link sets.

We propose three methods for adjusting the link weights to produce different link sets. The simplest method, called the *Once Method* (OM), generates link sets sequentially such that only new links appear on each link set and stops when all links appear in the link sets. This method always sets $\omega_k = 1$ if link k is not present in the candidate link sets and $\omega_k = 0$ otherwise.

The second method, called the *Multiple Method* (MM), generates link sets sequentially such that each new link set contains at least one new link and stops when all links appear at least once. This method can have links appear in many link sets. To achieve this, MM uses the following link weights as input to the MIMO weight algorithm:

$$\omega_k = \frac{1}{1+\delta_k} \text{ for all } k \in \{1, \ldots, M\}, \tag{6.1}$$

where δ_k is the number of times link k has been selected as active in the link sets previously computed by MM. To ensure that at least one new link (with $\omega_k = 1$) appears in the final link set, we square the values of ω_k for all $k \in \{1, \ldots, M\}$ at each iteration of the MIMO weight algorithm if the current link set does not activate at least one link for which $\omega_k = 1$.

The third method, called the *compensating method* (CM), first executes either the OM or the MM methods to completion. Then, the CM method tries to aid the scheduler in achieving certain proportions of bandwidth by generating extra link sets that compensate for previously generated link sets by the OM or MM methods. Let $\boldsymbol{b}$ be a vector of M positive elements, where the k^{th} element b_k represents the desired bandwidth portion to allocate to link k, so that $\sum_{k=1}^{M} b_k = 1$. Assume that a total of N link sets have already been computed. Also, let $\boldsymbol{A}$ be a $M \times N$ matrix, where the element $a_{k,n} \geq 0$ at the k^{th} row and n^{th} column of $\boldsymbol{A}$ contains the data rate of link k in the n^{th} link set. In case link k is not active in link set n, we set $a_{k,n} = 0$. This

method sets the link weights as follows:

$$\omega_k = \max\left(1 - \frac{\sum_{n=1}^{N} a_{k,n}}{b_k \sum_{i=1}^{M}\sum_{n=1}^{N} a_{i,n}}, 0\right) \tag{6.2}$$

for all $k \in \{1, \ldots, M\}$. Therefore, links that exceed their bandwidth proportion according to previously computed link sets will not be considered for the current link set ($\omega_k = 0$). This second method creates link sets until the total number exceeds a threshold N_{linksets}.

The OM and MM methods will be used by the scheduling algorithm proposed in Section 6.3, while the CM method will be used by the scheduling algorithms proposed next, in Section 6.2.

6.2 Maximizing Sum Rate while Achieving Proportional Fairness

In this section, we propose a technique for maximizing the sum rate while achieving proportional fairness as given by the vector of desired bandwidth portions $\boldsymbol{b}$. Our approach is to find the relative number of times to schedule each link set, then use a scaling function to scale this relative number to occupy the schedule duration. We also provide a technique to compensate for the non-ideality of the scaling function.

We assume that CM is used to compute the link sets, and that a total of N link sets (out of the N_{linksets} total link sets) are available. We wish to find a column vector $\boldsymbol{x} \geq \boldsymbol{0}$ that satisfies

$$\tfrac{1}{\alpha}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{6.3}$$

where: the n^{th} element x_n of vector $\boldsymbol{x}$ denotes the relative number of times to schedule the n^{th} link set with respect to the other $m \neq n$ link sets; $\boldsymbol{x} \geq \boldsymbol{0}$ denotes that each element x_n of $\boldsymbol{x}$ satisfies the inequality $x_n \geq 0$; and $\alpha = \sum_{k=1}^{M}\sum_{n=1}^{N} a_{k,n}$. In (6.3), we scale $\boldsymbol{A}$ with $\frac{1}{\alpha}$ to maintain numerical stability since the elements of $\boldsymbol{A}$ are potentially large compared to the elements of $\boldsymbol{b}$.

Notice that (6.3) can have zero, one, or many solutions, depending on $\boldsymbol{A}$. However, because we initialize the available link sets with the single-link link sets, problem (6.3) is guaranteed to always have at least one solution. For the case that many solutions exist for (6.3), we choose the solution that maximizes the sum rate. As we will see in the following theorem, the problem of finding the solution that maximizes the sum rate is related to the problem of finding the solution that minimizes the relative schedule length as given by the following linear programming problem:

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} & \sum_{n=1}^{N} x_k \\ \text{subject to } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned} \tag{6.4}$$

Theorem 2. *If problem* (6.3) *has a solution, then the solution that minimizes the relative schedule length in* (6.4) *is also the solution that maximizes the sum rate.*

Proof. Let $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{x} = \alpha\boldsymbol{b}$ with elements c_k for $k \in \{1, \ldots, M\}$. The total rate of link set n is given by $r_n = \sum_{k=1}^{M} a_{k,n}$, and so the relative amount of data that can be sent throughout the schedule is $\tau_{\text{slot}} \sum_{n=1}^{N} r_n x_n$. Therefore, the sum rate is given by

$$\begin{aligned} \text{Sum Rate} &= \frac{\tau_{\text{slot}} \sum_{n=1}^{N} r_n x_n}{\tau_{\text{slot}} \sum_{n=1}^{N} x_n} = \frac{\sum_{k=1}^{M} c_k}{\sum_{n=1}^{N} x_n} \\ &= \frac{\alpha \sum_{k=1}^{M} b_k}{\sum_{n=1}^{N} x_n} = \frac{\alpha}{\sum_{n=1}^{N} x_n}, \end{aligned}$$

since $\sum_{k=1}^{M} b_k = 1$. Therefore, given the setup of (6.4), by minimizing the relative schedule length $\sum_{n=1}^{N} x_n$, we are also maximizing the sum rate. □

The vector $\boldsymbol{x}$ that solves (6.4) defines a relative number of times to schedule each link set, the values of which are potentially non-integers. Let $\boldsymbol{s}$ be a vector where the n^{th} element s_n contains the integer number of times that link set n is to be scheduled. To compute the schedule $\boldsymbol{s}$ using $\boldsymbol{x}$, we find a factor β such that

$$\boldsymbol{s} = \text{Round}(\beta\boldsymbol{x}), \tag{6.5}$$

$$\tau_{\text{slot}} \sum_{n=1}^{N} s_n \approx \tau_{\text{schedule}}, \tag{6.6}$$

where τ_{schedule} is the desired duration of the schedule.

Because the schedule duration is limited, the scaling function (6.5) can be non-ideal, causing the actual proportions to deviate from the desired proportions. To compensate for this non-ideality, we define a history-aware version of (6.4) that accounts for the history of schedules when computing the new schedule.

Let $\boldsymbol{h}$ be a column vector where the n^{th} element h_n contains the number of times that link set n has been scheduled. The *history-aware version of* (6.4) can be obtained by replacing $\boldsymbol{b}$ in (6.4) with $\hat{\boldsymbol{b}} = \boldsymbol{b} - \frac{1}{\alpha}\boldsymbol{A}\boldsymbol{h}$ and setting α so as to both provide numerical stability and to ensure that all elements of $\hat{\boldsymbol{b}}$ are non-negative. In our simulations, we choose $\alpha = 2(\sum_{k=1}^{M}\sum_{n=1}^{N} a_{k,n})(\sum_{n=1}^{N} h_n + 1)/\min(\boldsymbol{b})$. Intuitively, $\hat{\boldsymbol{b}}$ represents the desired proportions minus those proportions that have already been satisfied. Note that Theorem 2 also applies to the history-aware version of problem (6.4), and so the solution to this problem also maximizes the sum rate.

Since link sets generated by the MIMO weight algorithm become available at different times, it is possible that a previous schedule contained a link set that is no longer optimal based on the newly available link sets. The solution to the history-aware version of (6.4) corrects for any previously selected link set that is no longer optimal. However, once a fairness threshold is reached, we reset the history to prevent re-selecting sub-optimal link sets, and a new schedule is generated by solving the original problem (6.4).

To compute the schedule $\boldsymbol{s}$ for the history-aware objective function, we set

$$\boldsymbol{s} = \text{Round}\Big((\boldsymbol{x} + \boldsymbol{h})\,\beta - \boldsymbol{h}\Big)_+, \tag{6.7}$$

where $(\cdot)_+$ is vector $(\cdot)$ with the negative entries replaced with zeros, and where β is the smallest factor that either satisfies (6.6) or satisfies a fairness constraint within

some small value ϵ. In our simulations, we use the fairness index proposed in [8], as given by

$$f(\boldsymbol{u}, \boldsymbol{b}) = \frac{1}{\exp\left(\frac{1}{M}\sum_{k=1}^{M}\left|\ln\frac{u_k}{b_k\sum_{j=1}^{M}u_j}\right|\right)}, \tag{6.8}$$

where $\boldsymbol{u}$ is the actual bandwidth usage and $\boldsymbol{b}$ is the desired bandwidth allocation. The fairness index given in (6.8) has a range between $[0, 1]$, with $f = 1$ when the allocation is perfectly fair, and $f \to 0$ as the allocation becomes less and less fair.

Figure 6.1 summarizes our proportionally fair scheduling algorithm for MIMO links.

Input: $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{h}, \tau_{\text{slot}}, \tau_{\text{schedule}})$
Output: $(\boldsymbol{s}, \boldsymbol{h})$

$\boldsymbol{\phi} = \mathbf{0}$;
$\boldsymbol{\theta} = \mathbf{0}$;
if $\boldsymbol{h} \neq \mathbf{0}$ **then**
    Compute $\boldsymbol{x}$ by solving the history-aware version of (6.4) and compute $\boldsymbol{s}$ using (6.7);
    $\boldsymbol{\phi} = \boldsymbol{s}$;
    $\tau_{\text{schedule}} = \tau_{\text{schedule}} - \tau_{\text{slot}}\sum_{n=1}^{N} s_n$;
    if $1 - f(\boldsymbol{A}(\boldsymbol{s}+\boldsymbol{h}), \boldsymbol{b}) < \epsilon$ **then**
        $\boldsymbol{h} = \mathbf{0}$;
    end if
end if
if $\boldsymbol{h} == \mathbf{0}$ **then**
    Compute $\boldsymbol{x}$ by solving (6.4) and compute $\boldsymbol{s}$ using (6.5);
    $\boldsymbol{\theta} = \boldsymbol{s}$;
end if
$\boldsymbol{s} = \boldsymbol{\phi} + \boldsymbol{\theta}$;
$\boldsymbol{h} = \boldsymbol{h} + \boldsymbol{s}$;
if $1 - f(\boldsymbol{A}\boldsymbol{h}, \boldsymbol{b}) < \epsilon$ **then**
    $\boldsymbol{h} = \mathbf{0}$;
end if
return $(\boldsymbol{s}, \boldsymbol{h})$;

Figure 6.1: Pseudocode for proportionally fair scheduling with MIMO links.

We define $\boldsymbol{b}$ so as to consider the following two fairness goals:

- **Time-based proportional fairness** – Following the ideas of [8], we define the time-based proportions with MIMO links as the proportion of data rates when each link is allocated an equal number of interference-free time slots. This is the standard notion of time-based fairness in wireless networks, except that it eliminates interference-induced distortions on the data rates. The time-based proportional fairness criterion can be achieved by solving problem (6.4) with $b_k = \gamma_k / \sum_{j=1}^{M} \gamma_j$ for all $k \in \{1, \ldots, M\}$, where γ_k is the data rate for a link k in the absence of interference when using the optimal SVD weights. We will refer to the instance of our framework that uses time-based proportional fairness as *TimeFair*.

- **Rate-based proportional fairness** – With rate-based proportional fairness, the goal is to achieve equal average rate across all links. This fairness criterion can be achieved by solving problem (6.4) with $b_k = 1/M$ for all $k \in \{1, \ldots, M\}$. We will refer to the instance of our framework that uses rate-based proportional fairness as *RateFair*.

6.3 Greedily Maximizing the Sum Rate

We also define an algorithm that greedily chooses a set of link sets to maximize the sum rate subject to the constraint that each link must appear at least once in the schedule while trying to keep the schedule length to a minimum.

We assume that for this algorithm, the link sets are generated using either the OM or the MM method. The algorithm is described as follows: for each link set in descending order according to their total rate $r_n = \sum_{k=1}^{M} a_{k,n}$, schedule each link set that contains a link that has not been previously scheduled. Since we initialize the link sets to contain the single-link link sets, this algorithm is guaranteed to schedule each link at least once. The algorithm then scales the schedule such as to occupy the desired duration τ_{schedule}. We will refer to the instance of the framework that

implements this greedy algorithm as *GreedyMaxRate*.

6.4 Simulation Results

In this section, we present results obtained using the ns-3 network simulator (see Section 5.5.1 for setup). In the following, we show results for the performance of 802.11n and that of the MIMO framework using the various scheduling algorithms and methods for generating link sets discussed in this chapter. We denote the Time-Fair scheduling algorithm with link sets generated by the OM method followed by the CM method as TimeFair-OM/CM. Similarly, we will denote the TimeFair scheduling algorithm with link sets generated by the MM method followed by the CM method as TimeFair-MM/CM. Using the same strategy, we can denote the RateFair algorithms. We will denote the GreedyMaxRate algorithm that uses the OM and the MM method for generating its link sets as GreedyMaxRate-OM and GreedyMaxRate-MM, respectively. We also show results for the MIMO framework using a modified GreedyMaxRate scheduling algorithm, called *NoISuplink* in which link sets are generated by the OM or MM mechanism for the downlink only. We denote the two variants as NoISuplink-OM and NoISuplink-MM. The NoISuplink algorithm does not perform interference suppression in the uplink and so it considers only the single-link link sets for scheduling in the uplink.

We assume that $\tau_{\text{schedule}} = 500$ ms initially. Once the Worker AP finishes computing all link sets and their associated MIMO weights, we set $\tau_{\text{schedule}} = 3$ secs. Additionally, we set $N_{\text{linksets}} = 2.5M$ for both the uplink link sets and downlink link sets, where M is the number of links in the network. Unless otherwise stated, we will assume that the duration in which packets can be aggregated is set to $\tau_{\text{data}} = 10$ ms. Also, we allocate a time $\tau_{ack} = 210$ μs. We assume that all data packets are UDP packets of 1024 bytes. Unless otherwise stated, we set the packet generation rate high enough so that each AP has data to send to each of its clients and each client has

data to send to its AP.

In the following, we compute the weighted average fairness index, so as to account for uplink and downlink traffic ratios as follows: the average fairness index is $f_{\text{average}} = p_{\text{downlink}} f_{\text{downlink}} + (1 - p_{\text{downlink}})\, f_{\text{uplink}}$, where f_{uplink} and f_{downlink} are the fairness index given by (6.8) for the uplink and downlink, respectively. For all algorithms except those based on RateFair, we used the time-based fairness proportions to measure the fairness index. For the algorithms based on RateFair, we used the rate-based fairness proportions to measure the fairness index.

This section is organized as follows. In Section 6.4.1, we present results for a controlled topology. In Section 6.4.2, we randomize the location of the clients. In Section 6.4.3, we study the performance of our proposed framework and scheduling algorithms under slow-varying channels. Finally, in Section 6.4.2.1, we present results on our framework and scheduling algorithms servicing a simulated voice over IP (VoIP) application.

6.4.1 Under a Controlled Topology

In this section, we consider the topology of Figure 6.2. We study the effects of interference by fixing the distance between every AP and their associated clients (thereby fixing the signal-to-noise ratio), and varying the distance between interfering APs (thereby varying the interference). For this simulation, we set the simulation time to 30 seconds.

Figure 6.3 shows the sum goodput as a function of x for $y = 50$ m, averaged over 50 trials, for the variations of the TimeFair-, RateFair-, GreedyMaxRate-, and NoISuplink-based algorithms that produced the highest sum goodput. In addition to these algorithms, we have also plotted results for an algorithm called *BestLinkSetOnly*, which forms the schedule using only the best link set available. For algorithm BestLinkSetOnly, we generate only two link sets. Both link sets set $\omega_k = 1$ for all

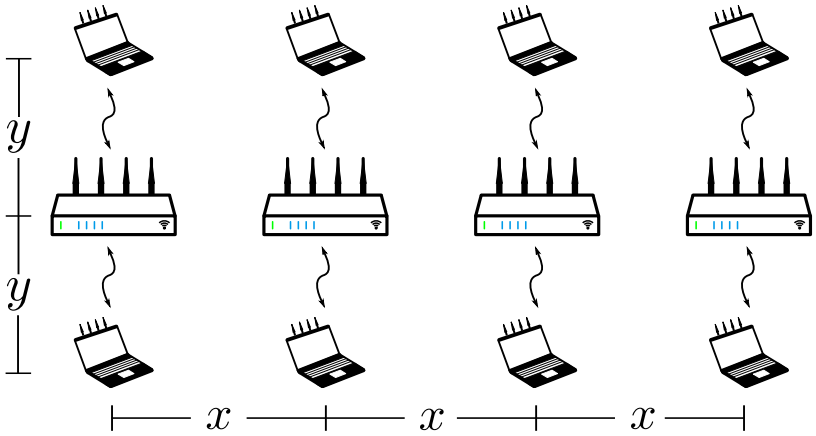

Figure 6.2: Topology of four APs with two associated clients each. The curved arrows between an AP and a client means that the client is associated with that AP.

links $k \in \{1, \ldots, M\}$. However, the first sets the maximum number of iterations to ten, whereas the second sets the maximum number of iterations to 100.

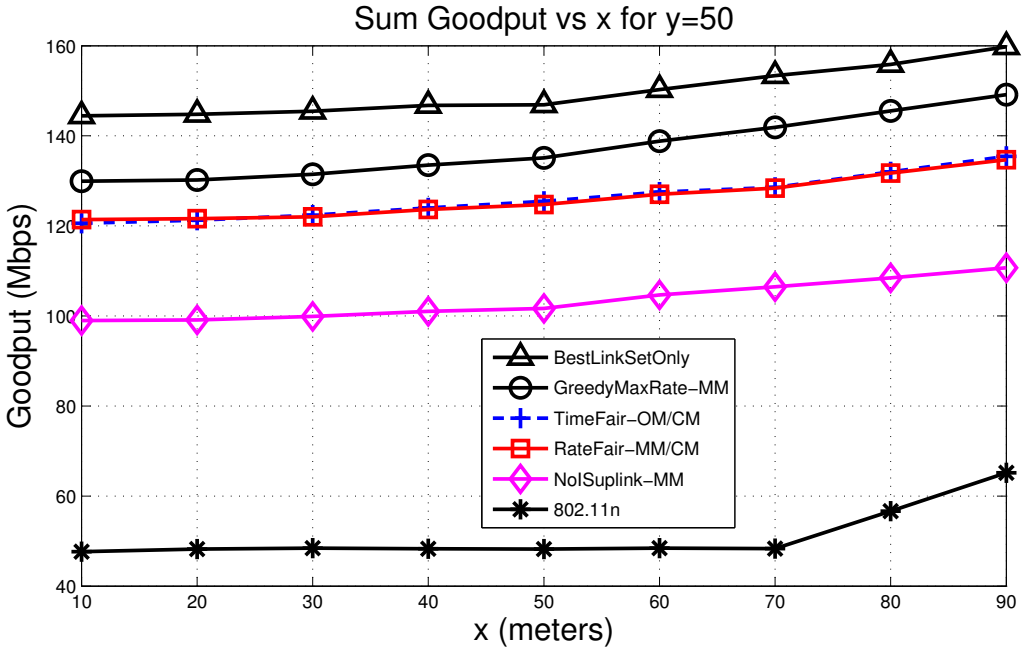

Figure 6.3: Sum goodput for topology of Figure 6.2.

The results show that the goodput is much greater for the algorithms that perform both interference suppression and spatial multiplexing, than for 802.11n, which performs spatial multiplexing only. The BestLinkSetOnly algorithm achieved a goodput

that is 217% and 19% better than that of 802.11n and that of both the TimeFair-OM/CM and RateFair-MM/CM algorithms, respectively. The GreedyMaxRate-MM algorithm achieved a goodput that is approximately 200% better than that of 802.11n at $x = 70$. Figure 6.3 also shows that GreedyMaxRate-MM achieved a goodput that is almost 35% better than NoISuplink-MM, highlighting the importance of performing interference suppression on both uplinks and downlinks. These results also show that, on average, GreedyMaxRate-MM achieves a goodput that is about 11% higher than that of both the TimeFair-OM/CM and RateFair-MM/CM algorithms. Both the TimeFair-OM/CM and RateFair-MM/CM algorithms achieved a goodput that is almost 165% better than that of 802.11n at $x = 70$. The TimeFair-OM/CM and RateFair-MM/CM algorithms achieved similar goodputs because the time-fair proportions and the rate-fair proportions are almost equal due to the structure of the simulated topology.

Figure 6.4 shows the average fairness index for the results shown in Figure 6.3. The results show that the TimeFair-OM/CM and RateFair-MM/CM algorithms achieved a fairness index that is very close to their goals. As expected, the GreedyMaxRate-MM is the most unfair algorithm. The NoISuplink-MM algorithm achieved a higher fairness index than the GreedyMaxRate-MM because the uplink, having only single-link link sets, achieved almost perfect time-based fairness.

6.4.2 With Varying Number of Clients

We now focus on the effect of varying the number of clients per AP and randomizing their placement. We simulate four APs in a line at 50 m intervals. Each AP has C associated clients that are uniformly distributed within a radius of 80 meters from the AP. For this simulation, we set the simulation time to 60 seconds.

Figure 6.5 shows the sum goodput as a function of the number of clients (C) per AP , averaged over 50 trials, for the variations of the TimeFair-, RateFair-,

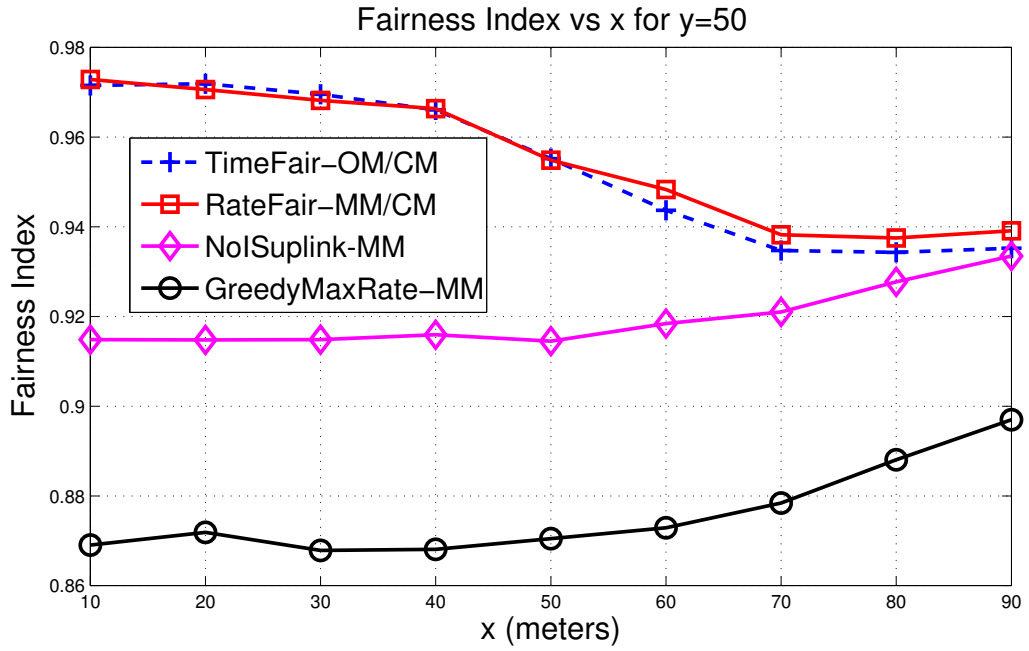

Figure 6.4: Average fairness index for the results of Figure 6.3 as a function of x for $y = 50$ m.

GreedyMaxRate-, and NoISuplink-based algorithms that produced the highest sum goodput. The results show that, again, the GreedyMaxRate-MM achieved the highest goodput of all algorithms, achieving a goodput that is almost 200% better than that of 802.11n. Additionally, the TimeFair-OM/CM algorithm achieved a goodput that is almost 160% better than that of 802.11n. Compared to the NoISuplink-MM algorithm, the GreedyMaxRate-MM algorithm achieved a goodput that is approximately 35% higher. Also, the GreedyMaxRate-MM algorithm achieved a goodput that is about 23% and 45% greater, on average, than the TimeFair-OM/CM and RateFair-MM/CM algorithm, respectively. Comparing the TimeFair-OM/CM and RateFair-MM/CM algorithms, it is clear that the TimeFair-OM/CM algorithm significantly outperforms the RateFair-MM/CM algorithm in these simulations. At $C = 6$, the TimeFair-OM/CM algorithm achieved a goodput that is approximately 23% better than that of the RateFair-MM/CM algorithm. These results are consistent with the well known fact that with rate-based fairness, the low rate links dominate air time,

thereby substantially reducing overall network performance [49].

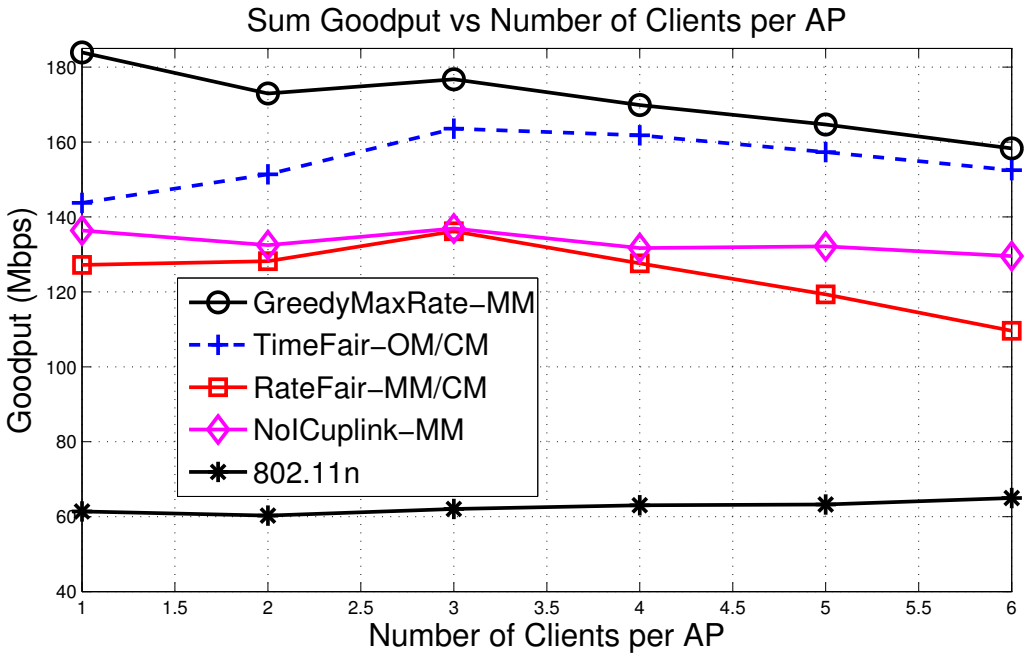

Figure 6.5: Sum goodput for four APs in a line, placed every $x = 50$ m, as a function of the number of clients per AP.

Notice in Figure 6.5 that algorithms that use the MIMO framework have decreasing goodput as the number of clients per AP is increased. The reason for this decrease in goodput is that as the number of clients per AP increases, the algorithms that use the MIMO framework spends a longer portion of the simulation time using the single-link link sets since the number of link sets to compute gets larger as the number of clients per AP increases.

Figure 6.6 shows the average fairness index as a function of C for the results shown in Figure 6.5. The results show that the TimeFair-OM/CM and RateFair-MM/CM algorithms maintained the highest fairness index of all algorithms. The GreedyMaxRate-MM algorithm, on the other hand, achieved the lowest fairness. The NoISuplink-MM algorithm, again, achieved higher fairness than the GreedyMaxRate-MM because the uplink portion of the average fairness index has almost perfect fairness.

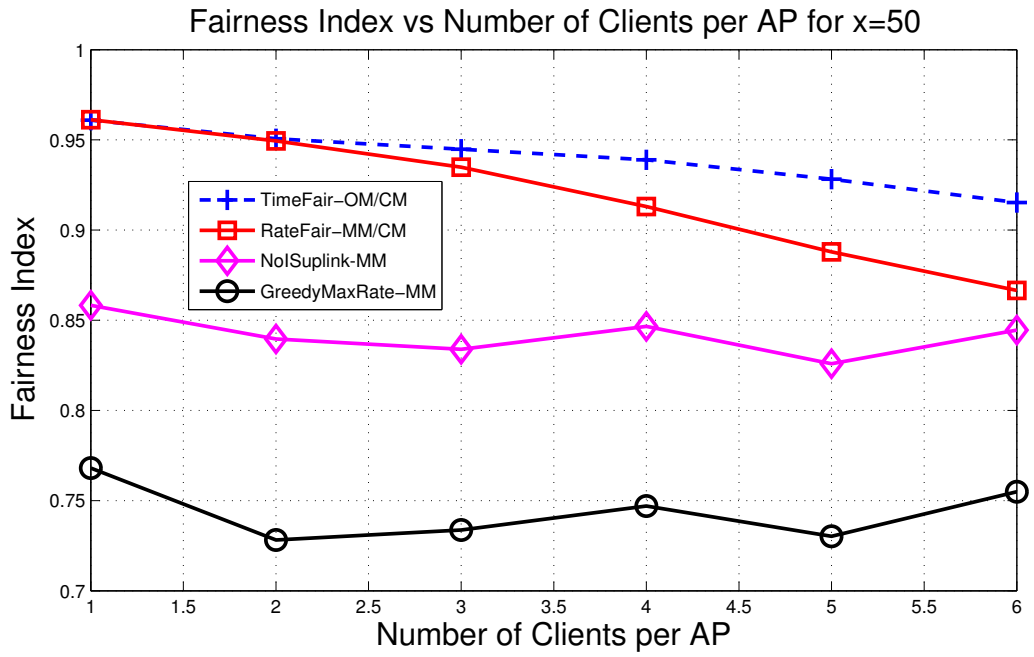

Figure 6.6: Average fairness index for the results of Figure 6.5.

6.4.2.1 Servicing a VoIP Application

In this section, we evaluate the performance of a VoIP application running within TimeFair-OM/CM or 802.11n. To simulate VoIP in ns-3, we implement the mathematical semi-call VoIP model proposed by [107] into the ns-3 simulator.

The simulation setup is as follows. We simulate four APs in a line, spaced every 90 meters. Each AP has four associated clients that are uniformly distributed within a radius of 80 meters from the AP. We randomly pick one of the links, and install the VoIP application in both the uplink and the downlink direction. This randomly chosen link only communicates VoIP packets and no other packets. We assume that a 64 kbps encoder is used and that packets are sent every 20 ms during the on-intervals of transmission. For the off-intervals of silence, we assume that a 200 byte packet is sent every 2 secs. For all other links, we assume that the downlink and uplink buffers always contain UDP data packets of 1024 bytes. For real-time applications, such as VoIP applications, the delay and jitter become an important measure for quality. Therefore, we are particularly interested in the performance of the TimeFair-OM/CM

algorithm in terms of delay and jitter.

The ITU G.114 [39], recommends that the delay for voice applications be less than 150 ms. In Figure 6.7, we show the ratio of VoIP packets that had less than 150 ms of delay for 50 random trials as a function of the aggregate-packet duration τ_{data} for the TimeFair-OM/CM algorithm. For reference, we also show results for 802.11n, which does not depend on the aggregate-packet duration τ_{data}.

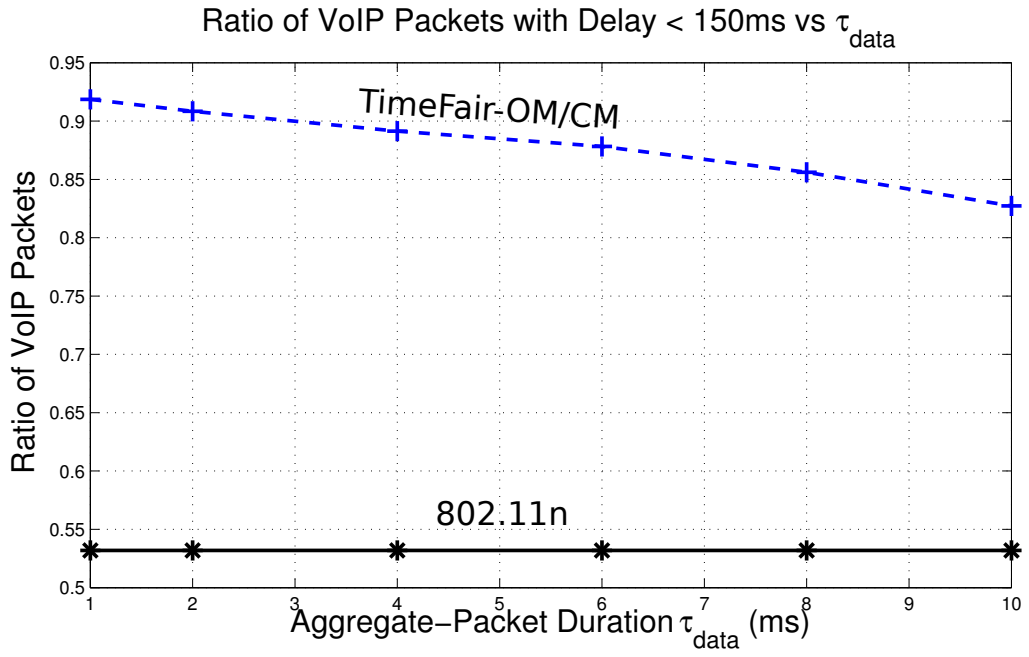

Figure 6.7: Ratio of VoIP packets with less than 150 ms of delay as a function of the aggregate-packet duration τ_{data}.

The results in Figure 6.7 show that as the τ_{data} increases, the number of VoIP packets that arrive with a delay less than 150 ms decreases. This is expected because a high value of τ_{data} means that links will have to wait longer for their turn to transmit. The results also show that for $\tau_{\text{data}} = 10$ ms, about 83% of packets have a delay less than 150 ms. However, if we decrease τ_{data} to 4 ms, almost 90% of packets have a delay less than 150 ms. This is a huge improvement over 802.11n, for which only 53% of packets have a delay less than 150 ms.

Another metric important to real-time applications is that of jitter or packet-delay

variation. In this document, we formally define jitter according to the definition in IETF RFC 3393 [31], which is given by

$$\text{Jitter}\,(P_n) = |\text{Delay}\,(P_n) - \text{Delay}\,(P_{n-1})|, \tag{6.9}$$

where Delay $(\cdot)$ is the delay of packet $(\cdot)$ and P_n is the packet at time instant n. To compensate for varying delays, real-time applications typically use a jitter buffer, which buffers the received packets and sends them in a steady stream to the playback interface. For the jitter buffer to be effective, however, jitter must be less than 100 ms [80].

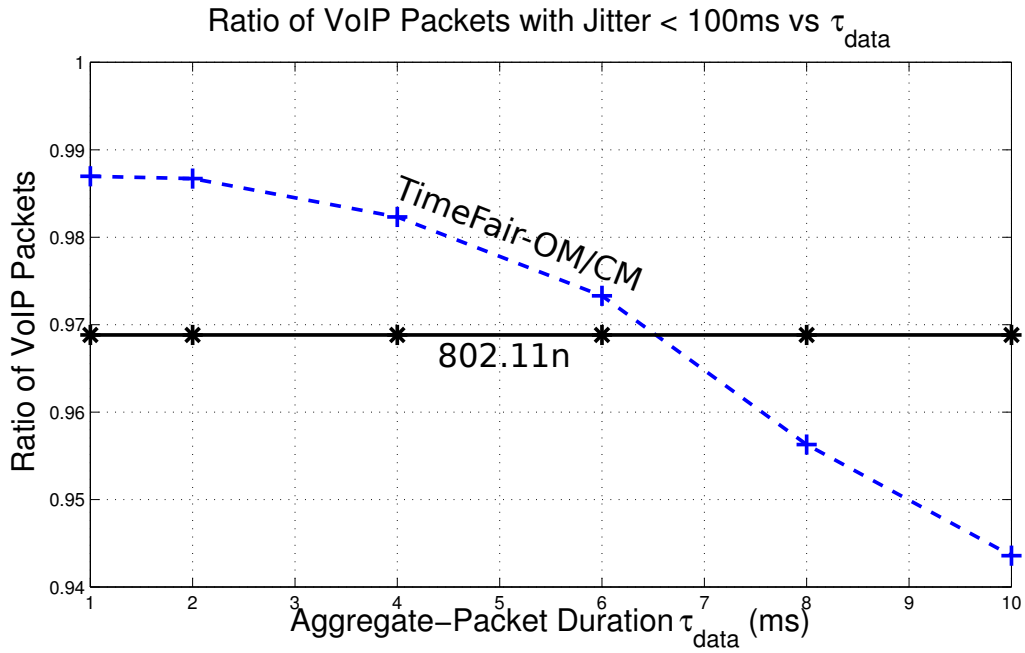

Figure 6.8: Ratio jitter measurements with less than 100 ms as a function of the aggregate-packet duration τ_{data}.

Figure 6.8 shows the ratio of jitter measurements that are less than 100 ms as a function of the aggregate-packet duration τ_{data} for the TimeFair-OM/CM algorithm for the experiment at hand. In Figure 6.8, we again show results for 802.11n, which does not depend on the aggregate-packet duration τ_{data}. The results show that overall, more than 90% of all jitter measurements are less than 100 ms at all values of τ_{data}

tested. As expected, the number of jitter measurements that are less than 100 ms decreased with increasing τ_{data}. Notice that for $\tau_{\text{data}} > 6.5$ ms, the TimeFair-OM/CM algorithm had a lower ratio of jitter measurements less than 100 ms as compared to the 802.11n. Although the 802.11n encountered less variation in packet delay, the actual delay of approximately 50% of all packets was higher than 150 ms (see Figure 6.7). On the other hand, when $\tau_{\text{data}} < 6.5$, the ratio of jitter measurements for the TimeFair-OM/CM algorithm was higher than that of 802.11n.

To see the effect of varying the aggregate-packet duration τ_{data} has on the goodput, we perform a similar experiment except that for this experiment no VoIP is installed and, instead, each AP sends as many UDP packets as possible to each of its clients and each client sends as many UDP packets its AP. Figure 6.9 shows the goodput as a function of the aggregate-packet duration τ_{data}, averaged over 50 trials. As expected, the results show that as the τ_{data} increase, the performance of the TimeFair-OM/CM algorithm increases. The results show that decreasing τ_{data} from 10 ms to 4 ms results in a decrease in throughput of about 4%.

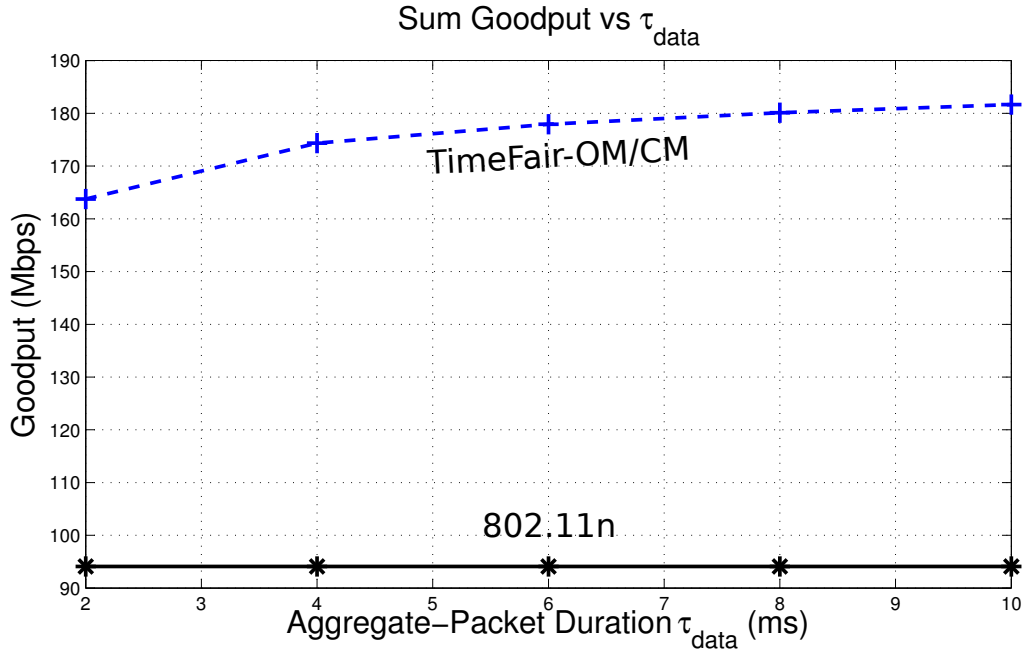

Figure 6.9: Average goodput as a function of the aggregate-packet duration τ_{data}.

These results suggest that the TimeFair-OM/CM algorithm can satisfy the packet delay and jitter requirements for VoIP applications in over 90% of packets in congested networks if $\tau_{\text{data}} \leq 4$ ms with minor goodput loss.

6.4.3 Under Time-Varying Channel Conditions

In this section, we consider the topology of Figure 6.2 with $y = 50$ meters for varying values of x under channel conditions that vary slowly with time, as would be expected in indoor environments. We model slow-varying channels using the random process described in [59, 105], as given by $\boldsymbol{H}_t = \omega \boldsymbol{H}_{t-1} + \sigma\sqrt{1-\omega^2}\boldsymbol{W}$, where: $\boldsymbol{H}_t$ is the new channel matrix at a discrete time t; $\boldsymbol{H}_{t-1}$ is the channel matrix at the previous discrete time $t-1$; and $\boldsymbol{W}$ is a random complex matrix representing the channel change. The entries of $\boldsymbol{H}_t$, $\boldsymbol{H}_{t-1}$, and $\boldsymbol{W}$ are i.i.d. zero-mean complex Gaussian variables with variance σ^2. We choose $\omega = 0.9996$ and we update the MIMO channels every 250 ms. Additionally, we set the simulation time to 15 seconds. This results in an average accumulated error of 20% where the accumulated error for each MIMO channel is calculated as $\|\boldsymbol{H}_{\text{end}} - \boldsymbol{H}_{\text{start}}\|/\|\boldsymbol{H}_{\text{end}}\|]$, where $\boldsymbol{H}_{\text{start}}$ is the MIMO channel at the start of the simulation, $\boldsymbol{H}_{\text{end}}$ is the MIMO channel at the end of the simulation, and $\|(\cdot)\|$ is the Frobenius norm of $(\cdot)$.

Figure 6.10 shows the sum goodput as a function of x for $y = 50$ m, averaged over 25 trials. As reference, we have also replotted the 802.11n results from Figure 6.2, where the channels are fixed throughout the simulation. The results show that although the MIMO channels are slowly varying, the performance of the TimeFair-OM/CM and GreedyMaxRate-MM is still up to 130% and 160% better better, respectively, than that of 802.11n. Note that the goodput increases more rapidly as x increases than in Figure 6.10. This rapid increase is due to the fact that as interference is decreased, the impact on the performance due to the imperfect CSI decreases.

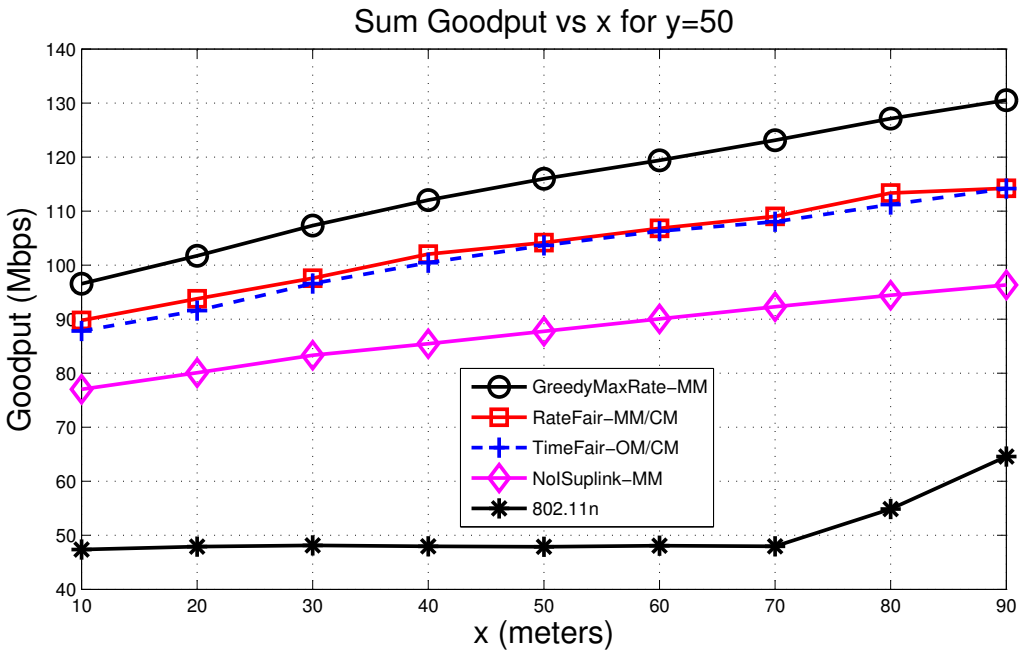

Figure 6.10: Sum goodput for topology of Figure 6.2 as a function of x for $y = 50$ m when channels are time varying.

6.5 Chapter Summary

We proposed the optimal scheduler that maximizes the sum rate subject to a proportional fairness constraint. To achieve this, we have proposed several methods for generating various link sets and their corresponding beamforming and combining weights. Then, the optimal scheduler picks which combination of link sets satisfy the fairness constraints and maximize the sum rate. Simulation results in ns-3 show that the scheduling algorithms in conjunction with our framework can increase the sum goodput by 165% with fair scheduling and by 200% without fair scheduling, as compared to the goodput of 802.11n. We have also evaluated our framework, with a time fair scheduling algorithm, using a VoIP application and found that our proposed time fair scheduling algorithm reduced the delay and jitter by 69% and 11%, respectively, as compared to that of 802.11n when the network is congested. Finally, we have evaluated the performance of our framework with the various scheduling algorithms with slow-varying MIMO channel conditions and found that the framework

and scheduling algorithms that we propose still increases the sum goodput by 130% with fair scheduling and 160% without fair scheduling, as compared to the goodput of 802.11n.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Concluding Remarks

In this dissertation, we described algorithms and techniques to take advantage of the spatial-multiplexing and interference-suppression capabilities of MIMO links to greatly improve the performance of single-hop wireless networks.

First, in Chapter 3, we showed that a simple bilateral strategy can greatly outperform all known unilateral strategies. In one example, we showed that a simple bilateral approach delivers an aggregate capacity that is 227% higher than that of the best unilateral approach.

Next, in Chapter 4, we designed a bilateral algorithm whose goal is to maximize the sum rate of a set of interfering MIMO links by jointly optimizing which subset of transmitters should transmit, the number of streams for each transmitter (if any), and the beamforming and combining weights that support those streams. Simulation results of this bilateral algorithm show that when interference is high, the algorithm can optimize which links to activate and deactivate, so as to achieve high sum rate. When a high number of links are present, the algorithm has diversity as to which links to activate and can achieve a higher sum rate than when only a few links are present.

Then, in Chapter 5, we designed a framework for optimizing dense single-hop wireless networks. The framework addresses several practical issues that arise when implementing interference suppression in dense wireless networks. Specifically, the framework implements techniques for: reducing the overhead of performing channel measurements and communicating channel state information; reducing the overhead

of computing the beamforming and combining weights; and overcoming the overhead of cooperation between the access points. To reduce the overhead associated with computing the beamforming and combining weights, the framework was designed to collect all CSI at a single node, calculate the beamforming and combining weights within that node, and reuse the calculated weights whenever possible. To reduce the number of channel measurements that must be collected, we proposed a technique called CAPOCS, which forces all APs to operate as either transmitters or receivers at any given time. We showed that CAPOCS has a negligible effect on the performance of the beamforming and combining weights algorithm while significantly reducing the time required to collect the necessary CSI. To overcome the overhead of cooperation between access points, the framework was design to aggregate packets and acknowledge these packets using BlockAcks. To avoid having to compute beamforming and combining weights for the reverse channel when transmitting the BlockAcks, the framework uses the concept of a virtual network in which transmitters become virtual receivers and receivers become virtual transmitters.

Finally, in Chapter 6, we have derived the optimal scheduler that, given a set of link sets for which beamforming and combining weights are available, computes how many times should each link set be used (if any) to achieve a given set of bandwidth proportions while also maximizing the sum rate.

We performed evaluations of our scheduling algorithm and framework using the ns-3 network simulator and showed that our scheduling algorithm achieves 160% higher proportionally fair goodput as compared to the goodput of 802.11n, which only supports spatial multiplexing.

7.2 Future Work

In future research, the bilateral algorithm in Chapter 4 can be studied further to create variants specific to centralized and distributed implementations. In Chapter 5,

we assumed that the beamforming and combining weight algorithm was implemented in a centralized fashion and we measured the CPU time the algorithm consumed and accounted for this overhead in the simulation. In practice, the algorithm can be specialized to support either multi-core computing processing units (CPUs) or graphical processing units (GPUs).

If a fully distributed implementation is desired, the bilateral algorithm in Chapter 4 should be studied further with emphasis on reducing the number of packet exchanges between the node and to account for unreliabilities and delays associated with the packet exchanges. Additionally, the framework that implements this specialized algorithm would need to be adapted to meet its requirements. In [28], we briefly discussed how a distributed framework can be designed. However, more work is needed to realize such an implementation.

The bilateral algorithm in Chapter 4 aims to maximize the sum rate based on Shannon's channel capacity equations. In practice, devices have a set of modulations and coding techniques that have different SINR requirements. Future research can tackle the problem of maximizing the sum rate for a discrete set of modulation and coding techniques given their SINR requirement.

In this dissertation, we assumed that CSI can be measured without errors. In future work, the bilateral algorithm in Chapter 4 can be re-designed to include the noise caused by imperfect CSI measurements and this imperfection can be accounted for within the simulator.

In Chapter 6, we considered the effects of slow-varying channels in our simulations. However, we used a simplistic method to model variations in the channel. In future work, a model based on real measurements can be used to better reflect the performance of the solutions presented in this dissertation.

Finally, since the solutions presented in this dissertation have been evaluated only through simulation, future work could involve implementing and field-testing these

solutions using hardware devices.

7.3 Publications

As part of the research conducted in this dissertation, we have written several documents that are either published, submitted, or in progress as follows:

- L. Cortés-Peña and D. Blough, "Achieving high performance and fairness in dense wireless networks of MIMO links," *IEEE/ACM Transactions on Networking*, 2014, In progress.

- L. Cortés-Peña and D. Blough, "Achieving high performance and fairness in dense wireless networks with MIMO interference cancellation," in *Proceedings of the ACM International Symposium Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Submitted, 2014.

- L. Cortés-Peña, J. Barry, and D. Blough, "Jointly optimizing stream allocation, beamforming and combining weights for the MIMO interference channel," *IEEE Transactions on Wireless Communications*, Dec. 2013, Submitted.

- L. Cortés-Peña, J. Barry, and D. Blough, "Joint optimization of stream allocation and beamforming and combining weights for the MIMO interference channel," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 4012–4018.

- L. Cortés-Peña and D. Blough, "Distributed MIMO interference cancellation for interfering wireless networks: protocol and initial simulations," in *Technical Report GIT-CERCS-13-02*, Available: `http://www.cercs.gatech.edu/tech-reports`, Feb. 2013.

- L. Cortés-Peña, J. Barry, and D. Blough, "The performance loss of unilateral interference cancellation," in *Proceedings of the IEEE International Conference*

on Communications (ICC), Jun. 2012, pp. 4181–4186.

- D. Blough, G. Resta, P. Santi, R. Shrinivasan, and L. Cortés-Peña, "Optimal one-shot scheduling for MIMO networks," in *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Jun. 2011, pp. 404–412.

- R. Srinivasan, D. Blough, L. Cortés-Peña, and P. Santi, "Maximizing throughput in MIMO networks with variable rate streams," in *Proceedings of European Wireless (EW) Conference*, Apr. 2010, pp. 551–559.

APPENDIX A

PROOF OF THEOREM 1

Theorem 1 (original on page 61): *The joint beamforming and combining weights that solve* (4.1) *are given by*

$$\boldsymbol{V}_k = \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Theta}_k, \qquad \text{(4.5 revisited)}$$

$$\boldsymbol{U}_k = \boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{F}_k\boldsymbol{\Phi}_k, \qquad \text{(4.6 revisited)}$$

where

$$\boldsymbol{\Theta}_k = \left(\boldsymbol{W}_k^{1/2}\boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2}\right)_+^{1/2}, \qquad \text{(4.7 revisited)}$$

$$\boldsymbol{\Phi}_k = \boldsymbol{W}_k^{-1/2}\boldsymbol{\Theta}_k, \qquad \text{(4.8 revisited)}$$

and $(\cdot)_+$ *is the matrix* $(\cdot)$ *with the negative entries replaced with zeros.*

Proof. The structure of the proof of Theorem 1 follows closely the proof of Lemma 1, Lemma 2, and Theorem 1 in [84], with several differences due to the presence of interfering links in our setting.

Before we begin, let us expand (4.3) to include the zero singular values as follows. Let

$$\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2} = \begin{bmatrix}\boldsymbol{F}_k & \widetilde{\boldsymbol{F}}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{D}_k & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{G}_k & \widetilde{\boldsymbol{G}}_k\end{bmatrix}^{\dagger} \qquad \text{(A.1)}$$

by singular-value decomposition (SVD), where $\widetilde{\boldsymbol{F}}_k \in \mathbb{C}^{n_{r_k}\times(n_{r_k}-d_k)}$ and $\widetilde{\boldsymbol{G}}_k \in \mathbb{C}^{n_{t_k}\times(n_{t_k}-d_k)}$ have orthonormal column vectors that correspond to the left and right eigenvectors of $\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2}$ with zero singular values, respectively.

The Lagrangian for (4.1) is given by

$$L(\boldsymbol{V}_k, \boldsymbol{U}_k, \mu_k) = \sum_{k=1}^{M} \mathrm{tr}\left(\boldsymbol{W}_k \boldsymbol{E}_k\right) + \mu_k \left(\mathrm{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}\right) - p_k\right). \tag{A.2}$$

where μ_k is the Lagrange multiplier for link k, and

$$\boldsymbol{E}_k = \boldsymbol{U}_k^{\dagger} \boldsymbol{H}_{kk} \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_{kk}^{\dagger} \boldsymbol{U}_k + \boldsymbol{U}_k^{\dagger} \boldsymbol{R}_{\bar{k}} \boldsymbol{U}_k - \boldsymbol{U}_k^{\dagger} \boldsymbol{H}_{kk} \boldsymbol{V}_k - \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_{kk}^{\dagger} \boldsymbol{U}_k + \boldsymbol{I}, \tag{A.3}$$

by expanding (4.2) using the definition of the received signal vector after combining (2.21). The Karush-Kuhn-Tucker (KKT) conditions to solve the optimization problem in (4.1) are

$$\nabla_{\boldsymbol{U}_k^{\dagger}} L = \boldsymbol{0}, \tag{A.4}$$

$$\nabla_{\boldsymbol{V}_k^{\dagger}} L = \boldsymbol{0}, \tag{A.5}$$

$$\mathrm{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}\right) - p_k \leq 0, \tag{A.6}$$

$$\mu_k \left(\mathrm{tr}\left(\boldsymbol{V}_k \boldsymbol{V}_k^{\dagger}\right) - p_k\right) = 0, \tag{A.7}$$

$$\mu_k \geq 0. \tag{A.8}$$

Setting the gradient of L with respect to $\boldsymbol{U}_k^{\dagger}$ to zero, we get

$$\boldsymbol{H}_{kk} \boldsymbol{V}_k = \boldsymbol{H}_{kk} \boldsymbol{V}_k \boldsymbol{V}_k^{\dagger} \boldsymbol{H}_{kk} \boldsymbol{U}_k + \boldsymbol{R}_{\bar{k}} \boldsymbol{U}_k, \tag{A.9}$$

and the gradient of L with respect to $\boldsymbol{V}_k^{\dagger}$ to zero, we get

$$\boldsymbol{H}_{kk}^{\dagger} \boldsymbol{U}_k \boldsymbol{W}_k = \boldsymbol{H}_{kk}^{\dagger} \boldsymbol{U}_k \boldsymbol{W}_k \boldsymbol{U}_k^{\dagger} \boldsymbol{H}_{kk} \boldsymbol{V}_k + \sum_{l=1, l \neq k}^{M} \boldsymbol{H}_{lk}^{\dagger} \boldsymbol{U}_l \boldsymbol{W}_l \boldsymbol{U}_l^{\dagger} \boldsymbol{H}_{lk} \boldsymbol{V}_k + \mu_k \boldsymbol{V}_k. \tag{A.10}$$

Using the structure of the proof of Lemma 1 in [84], we first prove that the joint beamforming and combining weights have the structure given in (4.5) and (4.6), where

$\boldsymbol{\Phi}_k$ and $\boldsymbol{\Theta}_k$ are arbitrary $d_k \times d_k$ matrices. We begin by assuming the most general expression for the beamforming and combining weights of link k as follows:

$$\boldsymbol{U}_k = \underbrace{\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{F}_k\boldsymbol{\Phi}_k}_{\boldsymbol{U}_{||}} + \underbrace{\boldsymbol{R}_{\bar{k}}^{-1/2}\widetilde{\boldsymbol{F}}_k\widetilde{\boldsymbol{\Phi}}_k}_{\boldsymbol{U}_{\perp}}, \tag{A.11}$$

$$\boldsymbol{V}_k = \underbrace{\boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Theta}_k}_{\boldsymbol{V}_{||}} + \underbrace{\boldsymbol{P}_{\bar{k}}^{-1/2}\widetilde{\boldsymbol{G}}_k\widetilde{\boldsymbol{\Theta}}_k}_{\boldsymbol{V}_{\perp}}, \tag{A.12}$$

where $\widetilde{\boldsymbol{\Phi}}_k$ is any $(n_{r_k} - d_k) \times d_k$ matrix and $\widetilde{\boldsymbol{\Theta}}_k$ is any $(n_{t_k} - d_k) \times d_k$ matrix. Note that since $\boldsymbol{R}_{\bar{k}}$ is square and full rank, it does not impose any constraint on (A.11). Similarly, since $\boldsymbol{P}_{\bar{k}}$ is square and assumed full rank, it does not impose any constraint on (A.12). In Section 4.1.2, we use Lemma 1 to prove that this assumption is correct.

To get (4.6) for arbitrary $\boldsymbol{\Phi}_k$, we premultiply (A.9) with $\boldsymbol{U}_{\perp}^{\dagger}$ to get

$$\boldsymbol{U}_{\perp}^{\dagger}\boldsymbol{R}_{\bar{k}}\boldsymbol{U}_k = \boldsymbol{0} \tag{A.13}$$

since

$$\begin{aligned}\boldsymbol{U}_{\perp}^{\dagger}\boldsymbol{H}_{kk}\boldsymbol{V}_k &= \widetilde{\boldsymbol{\Phi}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k^{\dagger}\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2}\left(\boldsymbol{G}_k\boldsymbol{\Theta}_k + \widetilde{\boldsymbol{G}}_k\widetilde{\boldsymbol{\Theta}}_k\right) \\ &= \widetilde{\boldsymbol{\Phi}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k^{\dagger}\boldsymbol{F}_k\boldsymbol{D}_k\boldsymbol{G}_k^{\dagger}\left(\boldsymbol{G}_k\boldsymbol{\Theta}_k + \widetilde{\boldsymbol{G}}_k\widetilde{\boldsymbol{\Theta}}_k\right) = \boldsymbol{0},\end{aligned}$$

and $\widetilde{\boldsymbol{F}}_k^{\dagger}\boldsymbol{F}_k = \boldsymbol{0}$. Expanding (A.13), we get

$$\begin{aligned}\widetilde{\boldsymbol{\Phi}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k^{\dagger}\boldsymbol{F}_k\boldsymbol{\Phi}_k + \widetilde{\boldsymbol{\Phi}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k\widetilde{\boldsymbol{\Phi}}_k &= \boldsymbol{0}, \\ \widetilde{\boldsymbol{\Phi}}_k^{\dagger}\widetilde{\boldsymbol{\Phi}}_k &= \boldsymbol{0},\end{aligned} \tag{A.14}$$

since $\widetilde{\boldsymbol{F}}_k^{\dagger}\widetilde{\boldsymbol{F}}_k = \boldsymbol{I}$. From (A.14), it is clear that $\widetilde{\boldsymbol{\Phi}}_k = \boldsymbol{0}$, and therefore $\boldsymbol{U}_{\perp} = \boldsymbol{0}$.

To get (4.5) for arbitrary $\boldsymbol{\Theta}_k$, we premultiply (A.10) by $\boldsymbol{V}_{\perp}^{\dagger}$ to get

$$\begin{aligned}\boldsymbol{V}_{\perp}^{\dagger}\left(\sum_{l=1,l\neq k}^{M}\boldsymbol{H}_{lk}^{\dagger}\boldsymbol{U}_l\boldsymbol{W}_l\boldsymbol{U}_l^{\dagger}\boldsymbol{H}_{lk} + \mu_k\boldsymbol{I}\right)\boldsymbol{V}_k &= \boldsymbol{0}, \\ \boldsymbol{V}_{\perp}^{\dagger}\boldsymbol{P}_{\bar{k}}\boldsymbol{V}_k &= \boldsymbol{0},\end{aligned} \tag{A.15}$$

because

$$\begin{aligned}\boldsymbol{V}_{\perp}^{\dagger}\boldsymbol{H}_{kk}^{\dagger}\boldsymbol{U}_k &= \widetilde{\boldsymbol{\Theta}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k^{\dagger}\boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}^{\dagger}\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{F}_k\boldsymbol{\Phi}_k \\ &= \widetilde{\boldsymbol{\Theta}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k^{\dagger}\boldsymbol{G}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{F}_k^{\dagger}\boldsymbol{F}_k\boldsymbol{\Phi}_k = \boldsymbol{0},\end{aligned}$$

and $\widetilde{\boldsymbol{G}}_k^{\dagger}\boldsymbol{G}_k^{\dagger} = \boldsymbol{0}$. Expanding (A.15), we get

$$\begin{aligned}\widetilde{\boldsymbol{\Theta}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k^{\dagger}\boldsymbol{G}_k\boldsymbol{\Theta}_k + \widetilde{\boldsymbol{\Theta}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k\widetilde{\boldsymbol{\Theta}}_k &= \boldsymbol{0}, \\ \widetilde{\boldsymbol{\Theta}}_k^{\dagger}\widetilde{\boldsymbol{\Theta}}_k &= \boldsymbol{0}, \qquad \text{(A.16)}\end{aligned}$$

since $\widetilde{\boldsymbol{G}}_k^{\dagger}\widetilde{\boldsymbol{G}}_k = \boldsymbol{I}$. It is clear from (A.16) that $\widetilde{\boldsymbol{\Theta}}_k = \boldsymbol{0}$, and therefore $\boldsymbol{V}_{\perp} = \boldsymbol{0}$.

Next, using the technique in the proof of Lemma 2 in [84], we prove that matrices $\boldsymbol{\Phi}_k$ in (4.6) and $\boldsymbol{\Theta}_k$ in (4.5) are diagonal matrices. Premultiplying (A.9) with $\boldsymbol{U}_k^{\dagger}$, premultiplying (A.10) with $\boldsymbol{V}_k^{\dagger}$, and simplifying using (A.1) we get

$$\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k = \boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k\boldsymbol{\Theta}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Phi}_k + \boldsymbol{\Phi}_k^{\dagger}\boldsymbol{\Phi}_k, \qquad \text{(A.17)}$$

$$\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k\boldsymbol{W}_k = \boldsymbol{\Theta}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Phi}_k\boldsymbol{W}_k\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k + \boldsymbol{\Theta}_k^{\dagger}\boldsymbol{\Theta}_k. \qquad \text{(A.18)}$$

From (A.17), we see that $\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k$ is Hermitian since the other terms are Hermitian. Similarly, $\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k\boldsymbol{W}_k$ in (A.18) is Hermitian since the other terms are Hermitian. Assuming that $\boldsymbol{W}_k$ has distinct diagonal entries, then $\boldsymbol{\Theta}_k$ and $\boldsymbol{\Phi}_k$ are diagonal matrices since $\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k$ is Hermitian, $\boldsymbol{W}_k$ is diagonal, and their multiplication $\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k\boldsymbol{W}_k$ is Hermitian. For the case where the diagonal elements of $\boldsymbol{W}_k$ have repeated entries, we follow [20, 84] and add a perturbation matrix $\triangle_{\boldsymbol{W}_k}$ that ensures that the elements of $\boldsymbol{W}_k$ are distinct. Since $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are continuous functions of $\boldsymbol{W}_k$ in (A.9) and (A.10), and $\lim_{\triangle_{\boldsymbol{W}_k}\to\boldsymbol{0}} \boldsymbol{V}_k(\boldsymbol{W}_k + \triangle_{\boldsymbol{W}_k}) = \boldsymbol{V}_k(\boldsymbol{W}_k)$, then we can treat $\boldsymbol{\Phi}_k^{\dagger}\boldsymbol{D}_k\boldsymbol{\Theta}_k$ to be diagonal for any $\boldsymbol{W}_k$.

Now, using the analysis in the proof of Lemma 2 in [84], we show that the diagonal entries of $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Theta}_k$ are nonnegative. Let $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2, \ldots$ denote diagonal matrices.

Then, let

$$\boldsymbol{\mathcal{D}}_1 = \boldsymbol{\Phi}_k^\dagger \boldsymbol{D}_k \boldsymbol{\Theta}_k, \tag{A.19}$$

$$\boldsymbol{\mathcal{D}}_2 = \boldsymbol{\Phi}_k^\dagger \boldsymbol{\Phi}_k \succeq 0, \tag{A.20}$$

$$\boldsymbol{\mathcal{D}}_3 = \boldsymbol{\Theta}_k^\dagger \boldsymbol{\Theta}_k \succeq 0, \tag{A.21}$$

where $(\cdot) \succeq 0$ denotes that $(\cdot)$ is a positive semidefinite matrix. Let $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ be unitary matrices, then (A.20) and (A.21) can be rewritten as

$$\boldsymbol{\Phi}_k = \boldsymbol{\mathcal{U}} \boldsymbol{\mathcal{D}}_2^{1/2}, \tag{A.22}$$

$$\boldsymbol{\Theta}_k = \boldsymbol{\mathcal{V}} \boldsymbol{\mathcal{D}}_3^{1/2}. \tag{A.23}$$

Plugging in (A.22) and (A.23) into (A.19), we get

$$\begin{aligned} \boldsymbol{\mathcal{D}}_1 &= \boldsymbol{\mathcal{D}}_2^{1/2} \boldsymbol{\mathcal{U}}^\dagger \boldsymbol{D}_k \boldsymbol{\mathcal{V}} \boldsymbol{\mathcal{D}}_3^{1/2}, \\ \boldsymbol{\mathcal{D}}_4 &= \boldsymbol{\mathcal{D}}_2^{-1/2} \boldsymbol{\mathcal{D}}_1 \boldsymbol{\mathcal{D}}_3^{-1/2} = \boldsymbol{\mathcal{U}}^\dagger \boldsymbol{D}_k \boldsymbol{\mathcal{V}}. \end{aligned} \tag{A.24}$$

By left and right multiplication of (A.24) with its conjugate transpose, we get

$$\boldsymbol{\mathcal{D}}_4^2 = \boldsymbol{\mathcal{V}}^\dagger \boldsymbol{D}_k^2 \boldsymbol{\mathcal{V}}, \tag{A.25}$$

$$\boldsymbol{\mathcal{D}}_4^2 = \boldsymbol{\mathcal{U}}^\dagger \boldsymbol{D}_k^2 \boldsymbol{\mathcal{U}}, \tag{A.26}$$

respectively. From (A.25) and (A.26) , it is clear that

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{D}}_5, \tag{A.27}$$

where $\boldsymbol{\mathcal{D}}_5$ has elements $e^{j\theta_1}, \ldots, e^{j\theta_{d_k}}$ in its diagonal, for arbitrary $\theta_i \in [0, 2\pi]$. Because the choice of θ_i does not impose any restrictions on the solution, we follow [84] and choose $\theta_i = 0$ for all $i \in \{1, \ldots, d_k\}$ so that $\boldsymbol{\mathcal{D}}_5 = \boldsymbol{I}$. Therefore, plugging in (A.27) into (A.22) and (A.23) we get

$$\boldsymbol{\Phi}_k = \boldsymbol{\mathcal{D}}_2^{1/2} \succeq 0, \tag{A.28}$$

$$\boldsymbol{\Theta}_k = \boldsymbol{\mathcal{D}}_3^{1/2} \succeq 0, \tag{A.29}$$

which proves that $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Theta}_k$ are diagonal matrices with real nonnegative entries.

Finally, using a similar analysis as the one presented in the proof of Theorem 1 in [84], we derive (4.7) and (4.8). Simplifying (A.17) and (A.18) using (A.19), and then plugging in (A.28) and (A.29) into the resulting expressions and into (A.19), we get

$$\boldsymbol{\mathcal{D}}_1 = \boldsymbol{\mathcal{D}}_1^2 + \boldsymbol{\mathcal{D}}_2, \tag{A.30}$$

$$\boldsymbol{\mathcal{D}}_1 \boldsymbol{W}_k = \boldsymbol{\mathcal{D}}_1 \boldsymbol{W}_k \boldsymbol{\mathcal{D}}_1 + \boldsymbol{\mathcal{D}}_3, \tag{A.31}$$

$$\boldsymbol{\mathcal{D}}_1 = \boldsymbol{\mathcal{D}}_2^{1/2} \boldsymbol{D}_k \boldsymbol{\mathcal{D}}_3^{1/2}. \tag{A.32}$$

Solving (A.30) for $\boldsymbol{\mathcal{D}}_2$, (A.31) for $\boldsymbol{\mathcal{D}}_3$, plugging the results into (A.32), and simplifying, we get

$$\boldsymbol{W}_k^{1/2} \boldsymbol{\mathcal{D}}_1 \left(\boldsymbol{I} - \boldsymbol{\mathcal{D}}_1 \right) \boldsymbol{D}_k = \boldsymbol{\mathcal{D}}_1. \tag{A.33}$$

Solving for $\boldsymbol{\mathcal{D}}_1$, we get

$$\boldsymbol{\mathcal{D}}_1 = \left(\boldsymbol{I} - \boldsymbol{W}_k^{-1/2} \boldsymbol{D}_k^{-1} \right)_+, \tag{A.34}$$

where $(\cdot)_+$ is necessary since $\boldsymbol{\mathcal{D}}_1 \succeq 0$. Plugging in (A.34) into (A.30) and (A.31) we get

$$\boldsymbol{\mathcal{D}}_2 = \left(\boldsymbol{W}_k^{-1/2} \boldsymbol{D}_k^{-1} - \boldsymbol{W}_k^{-1} \boldsymbol{D}_k^{-2} \right)_+, \tag{A.35}$$

$$\boldsymbol{\mathcal{D}}_3 = \left(\boldsymbol{W}_k^{1/2} \boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2} \right)_+. \tag{A.36}$$

Finally, plugging in (A.35) and (A.36) into (A.28) and (A.29), we get (4.8) and (4.7), respectively.

APPENDIX B

PROOF OF LEMMA 1

Lemma 1 (original on page 62): *If* $\mathbf{\Pi}_k$ *in* (4.4) *is singular, then the limit of* $\mathrm{tr}(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger)$ *using* $\boldsymbol{V}_k$ *in* (4.5) *as* μ_k *approaches zero from the right is*

$$\lim_{\mu_k \to 0^+} \mathrm{tr}\left(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\right) = \infty. \qquad \text{(4.10 revisited)}$$

Proof. Let $\mathbf{\Pi}_k = \boldsymbol{J}_k\mathbf{\Sigma}_k\boldsymbol{J}_k^\dagger$ by eigenvalue decomposition, where $\mathbf{\Sigma}_k \in \mathbb{R}^{n_{t_k}\times n_{t_k}}$ is diagonal and $\boldsymbol{J}_k \in \mathbb{C}^{n_{t_k}\times n_{t_k}}$ is unitary. Then, we can rewrite (4.4) as

$$\begin{aligned}\boldsymbol{P}_{\bar{k}} &= \boldsymbol{J}_k\mathbf{\Sigma}_k\boldsymbol{J}_k^\dagger + \mu_k\boldsymbol{I} \\ &= \boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)\boldsymbol{J}_k^\dagger. \end{aligned} \qquad \text{(B.1)}$$

Using (B.1), we rewrite (4.5) as

$$\begin{aligned}\boldsymbol{V}_k &= \left(\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)\boldsymbol{J}_k^\dagger\right)^{-1/2}\boldsymbol{G}_k\mathbf{\Theta}_k \\ &= \left(\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{1/2}\boldsymbol{J}_k^\dagger\right)^{-1}\boldsymbol{G}_k\mathbf{\Theta}_k \\ &= \boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1/2}\boldsymbol{J}_k^\dagger\boldsymbol{G}_k\mathbf{\Theta}_k. \end{aligned} \qquad \text{(B.2)}$$

Similarly, we rewrite the SVD on (4.3) as

$$\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{k\bar{k}}\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1/2}\boldsymbol{J}_k^\dagger = \boldsymbol{F}_k\boldsymbol{D}_k\boldsymbol{G}_k^\dagger, \qquad \text{(B.3)}$$

and the power used at the transmitter of link k as

$$\begin{aligned}\mathrm{tr}\left(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\right) &= \mathrm{tr}\left(\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1/2}\boldsymbol{J}_k^\dagger\boldsymbol{G}_k\mathbf{\Theta}_k^2\boldsymbol{G}_k^\dagger\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1/2}\boldsymbol{J}_k^\dagger\right) \\ &= \mathrm{tr}\left(\boldsymbol{J}_k\left(\mathbf{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1}\boldsymbol{J}_k^\dagger\boldsymbol{G}_k\mathbf{\Theta}_k^2\boldsymbol{G}_k^\dagger\right), \end{aligned} \qquad \text{(B.4)}$$

where we have used the identity $\mathrm{tr}\left(\boldsymbol{X}\boldsymbol{Y}\right) = \mathrm{tr}\left(\boldsymbol{Y}\boldsymbol{X}\right)$ to obtain (B.4). If $\boldsymbol{\Pi}_k$ is singular then at least one diagonal element in $\boldsymbol{\Sigma}_k$ is zero. Then, in the limit as $\mu_k \to 0^+$, at least the first element of $\boldsymbol{D}_k$ in (B.3) will approach infinity, and so the corresponding element in $\boldsymbol{\Theta}_k^2 = (\boldsymbol{W}_k^{1/2}\boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2})_+$ of (B.4) will approach zero. Notice that, as $\mu_k \to 0^+$, the $(\cdot)_+$ operator has no effect on the first element of $\boldsymbol{\Theta}_k^2$ since the first element of $\boldsymbol{D}_k^{-2}$ approaches zero faster than that of $\boldsymbol{D}_k^{-1}$. Also, in the limit as $\mu_k \to 0^+$, at least one element of $\left(\boldsymbol{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1}$ in (B.4) approaches infinity. Because this element of $\left(\boldsymbol{\Sigma}_k + \mu_k\boldsymbol{I}\right)^{-1}$ increases with μ_k^{-1} and the smallest nonzero value of $\boldsymbol{\Theta}_k^2$ decreases with $\mu_k^{-1/2}$ as $\mu_k \to 0^+$ in (B.4), then the power used by the transmitter of link k approaches infinity as $\mu_k \to 0^+$. Therefore,

$$\lim_{\mu_k \to 0^+} \mathrm{tr}\left(\boldsymbol{V}_k\boldsymbol{V}_k^\dagger\right) = \infty. \tag{B.5}$$

APPENDIX C

PROOF OF LEMMA 3

Lemma 3 (original on page 65): *If the combining weights for link k are given by* (4.6)*, then the MMSE combining weights for link k of the virtual network are $\overleftarrow{\boldsymbol{U}}_k^{MMSE} = \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k$, and so the beamforming weights of the real network are given by* (4.5)*.*

Proof. It is easy to see that the MMSE combining weights for the virtual receiver of link k are given by

$$\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} = \left(\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k\overleftarrow{\boldsymbol{V}}_k^{\dagger}\overleftarrow{\boldsymbol{H}}_{kk}^{\dagger} + \overleftarrow{\boldsymbol{R}}_{\bar{k}}\right)^{-1}\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k. \tag{C.1}$$

Equation (C.1) can be rewritten as

$$\begin{aligned}\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} &= \left(\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{1/2}\left(\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2}\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k\overleftarrow{\boldsymbol{V}}_k^{\dagger}\overleftarrow{\boldsymbol{H}}_{kk}^{\dagger}\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2} + \boldsymbol{I}\right)\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{1/2}\right)^{-1}\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k \\ &= \overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2}\left(\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2}\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k\overleftarrow{\boldsymbol{V}}_k^{\dagger}\overleftarrow{\boldsymbol{H}}_{kk}^{\dagger}\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2} + \boldsymbol{I}\right)^{-1}\overleftarrow{\boldsymbol{R}}_{\bar{k}}^{-1/2}\overleftarrow{\boldsymbol{H}}_{kk}\overleftarrow{\boldsymbol{V}}_k. \end{aligned}\tag{C.2}$$

Substituting $\overleftarrow{\boldsymbol{R}}_{\bar{k}} = \boldsymbol{P}_{\bar{k}}$, $\overleftarrow{\boldsymbol{H}}_{lk} = \boldsymbol{H}_{kl}^{\dagger}$, and $\overleftarrow{\boldsymbol{V}}_k = \boldsymbol{U}_k\boldsymbol{W}_k^{1/2}$ into (C.2) we get

$$\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} = \boldsymbol{P}_{\bar{k}}^{-1/2}\left(\boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}^{\dagger}\boldsymbol{U}_k\boldsymbol{W}_k\boldsymbol{U}_k^{\dagger}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2} + \boldsymbol{I}\right)^{-1}\boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}^{\dagger}\boldsymbol{U}_k\boldsymbol{W}_k^{1/2}. \tag{C.3}$$

Using (4.6) and (4.3) we see that

$$\begin{aligned}\boldsymbol{U}_k^{\dagger}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2} &= \boldsymbol{\Phi}_k\boldsymbol{F}_k^{\dagger}\boldsymbol{R}_{\bar{k}}^{-1/2}\boldsymbol{H}_{kk}\boldsymbol{P}_{\bar{k}}^{-1/2} \\ &= \boldsymbol{\Phi}_k\boldsymbol{D}_k\boldsymbol{G}_k^{\dagger} = \boldsymbol{D}_k\boldsymbol{\Phi}_k\boldsymbol{G}_k^{\dagger}\end{aligned}\tag{C.4}$$

and so (C.3) becomes

$$\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} = \boldsymbol{P}_{\bar{k}}^{-1/2}\left(\boldsymbol{G}_k\boldsymbol{\Phi}_k\boldsymbol{D}_k^2\boldsymbol{W}_k\boldsymbol{\Phi}_k\boldsymbol{G}_k^{\dagger} + \boldsymbol{I}\right)^{-1}\boldsymbol{G}_k\boldsymbol{\Phi}_k\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}. \tag{C.5}$$

Using the matrix inversion lemma we can rewrite (C.5) as

$$\begin{aligned}
\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} &= \boldsymbol{P}_{\bar{k}}^{-1/2}\left(\boldsymbol{I}-\boldsymbol{G}_k\boldsymbol{\Phi}_k\left(\boldsymbol{D}_k^{-2}\boldsymbol{W}_k^{-1}+\boldsymbol{\Phi}_k^2\right)^{-1}\boldsymbol{\Phi}_k\boldsymbol{G}_k^{\dagger}\right)\boldsymbol{G}_k\boldsymbol{\Phi}_k\boldsymbol{D}_k\boldsymbol{W}_k^{1/2} \\
&= \boldsymbol{P}_{\bar{k}}^{-1/2}\left(\boldsymbol{G}_k\boldsymbol{\Phi}_k\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}-\boldsymbol{G}_k\boldsymbol{\Phi}_k\left(\boldsymbol{D}_k^{-2}\boldsymbol{W}_k^{-1}+\boldsymbol{\Phi}_k^2\right)^{-1}\boldsymbol{\Phi}_k^2\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}\right) \\
&= \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k\Bigg(\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}-\boldsymbol{D}_k\boldsymbol{W}_k^{1/2} \\
&\quad\left(\boldsymbol{D}_k^{-1}\boldsymbol{W}_k^{-1/2}+\boldsymbol{\Phi}_k^2\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}\right)^{-1}\boldsymbol{\Phi}_k^2\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}\Bigg). \qquad \text{(C.6)}
\end{aligned}$$

Applying the matrix inversion lemma to (C.6) and substituting (4.8) into the resulting expression we get

$$\begin{aligned}
\overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}} &= \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k\left(\boldsymbol{D}_k^{-1}\boldsymbol{W}_k^{-1/2}+\boldsymbol{D}_k\boldsymbol{W}_k^{1/2}\boldsymbol{\Phi}_k^2\right)^{-1} \\
&= \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k\left(\boldsymbol{D}_k^{-1}\boldsymbol{W}_k^{-1/2}+\boldsymbol{D}_k\boldsymbol{W}_k^{-1/2}\boldsymbol{\Theta}_k^2\right)^{-1} \\
&= \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k\left(\boldsymbol{D}_k^{-1}\boldsymbol{W}_k^{-1/2}+\boldsymbol{D}_k\boldsymbol{W}_k^{-1/2}\left(\boldsymbol{W}_k^{1/2}\boldsymbol{D}_k^{-1}-\boldsymbol{D}_k^{-2}\right)_+\right)^{-1} \\
&= \boldsymbol{P}_{\bar{k}}^{-1/2}\boldsymbol{G}_k\boldsymbol{\Phi}_k. \qquad \text{(C.7)}
\end{aligned}$$

Since $\overleftarrow{\boldsymbol{U}}_k = \boldsymbol{V}_k\boldsymbol{W}_k^{-1/2}$, then we have $\boldsymbol{V}_k = \overleftarrow{\boldsymbol{U}}_k^{\mathrm{MMSE}}\boldsymbol{W}_k^{1/2}$ and so the beamforming weights are given by (4.5).

APPENDIX D

PROOF OF PROPOSITION 1

Proposition 1 (original on page 66): *For the case when the transmitter of link k causes no interference to other receivers but the receiver of link k is subject to interference from other transmitters and $\boldsymbol{W}_k = \alpha_k \boldsymbol{\Lambda}_k$ for any scalar $\alpha_k > 0$, the beamforming and combining weights in* (4.5) *and* (4.6) *reduce to the optimal eigen-mode transmission with power allocated through waterfilling.*

Proof. For this case, $\boldsymbol{P}_{\bar{k}} = \mu_k \boldsymbol{I}$ and (4.3) simplifies to

$$\begin{aligned} \mu_k^{-1/2} \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} &= \boldsymbol{F}_k \boldsymbol{D}_k \boldsymbol{G}_k^\dagger, \\ \boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk} &= \boldsymbol{F}_k (\mu_k^{1/2} \boldsymbol{D}_k) \boldsymbol{G}_k^\dagger. \end{aligned} \tag{D.1}$$

From (D.1), it is clear that $\boldsymbol{\Lambda}_k^{1/2} = \mu_k^{1/2} \boldsymbol{D}_k$ contains the singular values of the whitened channel $\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk}$ in decreasing order from top left to bottom right, and that $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ correspond to the left and right eigenvectors of $\boldsymbol{R}_{\bar{k}}^{-1/2} \boldsymbol{H}_{kk}$ with positive singular values, respectively. Plugging in $\boldsymbol{P}_{\bar{k}} = \mu_k \boldsymbol{I}$ into (4.5), the beamforming weights simplify to

$$\boldsymbol{V}_k = \boldsymbol{G}_k \left(\mu_k^{-1/2} \boldsymbol{\Theta}_k \right), \tag{D.2}$$

and so the transmitter of link k is transmitting through the eigen-modes of the whitened channel.

Now we show that power is allocated through waterfilling. Using (D.2), the inequality constraint $\mathrm{tr}(\boldsymbol{V}_k \boldsymbol{V}_k^\dagger) \leq p_k$ becomes

$$\mathrm{tr} \left(\boldsymbol{G}_k \left(\mu_k^{-1} \boldsymbol{\Theta}_k^2 \right) \boldsymbol{G}_k^\dagger \right) \leq p_k$$

$$\operatorname{tr}\left(\mu_k^{-1}\boldsymbol{\Theta}_k^2\right) \leq p_k \tag{D.3}$$

$$\operatorname{tr}\left(\mu_k^{-1}\left(\boldsymbol{W}_k^{1/2}\boldsymbol{D}_k^{-1} - \boldsymbol{D}_k^{-2}\right)_+\right) \leq p_k$$

$$\operatorname{tr}\left(\mu_k^{-1/2}\boldsymbol{W}_k^{1/2}\left(\mu_k^{1/2}\boldsymbol{D}_k\right)^{-1} - \left(\mu_k\boldsymbol{D}_k^2\right)^{-1}\right)_+ \leq p_k$$

$$\operatorname{tr}\left(\mu_k^{-1/2}\boldsymbol{W}_k^{1/2}\boldsymbol{\Lambda}_k^{-1/2} - \boldsymbol{\Lambda}_k^{-1}\right)_+ \leq p_k, \tag{D.4}$$

where we have used the fact that $\operatorname{tr}(\boldsymbol{X}\boldsymbol{Y}) = \operatorname{tr}(\boldsymbol{Y}\boldsymbol{X})$ to obtain (D.3). To maximize the rate on its link, the transmitter of link k transmits with maximum power and the inequality in (D.4) is treated with equality. Finally, with the choice of $\boldsymbol{W}_k = \alpha_k\boldsymbol{\Lambda}_k$ for any scalar $\alpha_k > 0$, (D.4) becomes the waterfilling solution [33, 101, 108]

$$\operatorname{tr}\left(\gamma_k\boldsymbol{I} - \boldsymbol{\Lambda}_k^{-1}\right)_+ = p_k, \tag{D.5}$$

where $\gamma_k = \sqrt{\alpha_k/\mu_k}$ is the waterfilling level.

REFERENCES

[1] "Aerohive: hive OS," Product information available at `http://www.aerohive.com/products/software-management/hiveos`.

[2] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, 1995.

[3] V. Annapureddy and V. Veeravalli, "Gaussian interference networks: sum capacity in the low-interference regime and new outer bounds on the capacity region," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3032–3050, Jul. 2009.

[4] G. Arslan, M. Demirkol, and Y. Song, "Equilibrium efficiency improvement in MIMO interference systems: a decentralized stream control approach," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2984–2993, Aug. 2007.

[5] S. J. Bae, B.-G. Choi, H. S. Chae, and M. Y. Chung, "Self-configuration scheme to alleviate interference among aps in IEEE 802.11 WLAN," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2012, pp. 1025–1030.

[6] J. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, Third. Springer Netherlands, 2004.

[7] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, May 2007, pp. 2326–2330.

[8] D. Blough, G. Resta, and P. Santi, "Interference-aware proportional fairness for multi-rate wireless networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2014.

[9] D. Blough, G. Resta, P. Santi, R. Shrinivasan, and L. Cortés-Peña, "Optimal one-shot scheduling for MIMO networks," in *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Jun. 2011, pp. 404–412.

[10] D. M. Blough, P. Santi, and R. Srinivasan, "On the feasibility of unilateral interference cancellation in MIMO networks," *IEEE/ACM Transactions on Networking*, 2014, To appear.

[11] R. Blum, "MIMO capacity with interference," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 793–801, Jun. 2003.

[12] R. Blum, J. Winters, and N. Sollenberger, "On the capacity of cellular systems with MIMO," *IEEE Communications Letters*, vol. 6, no. 6, pp. 242–244, Jun. 2002.

[13] A. Boukerche and K. Abrougui, "An efficient leader election protocol for mobile networks," in *Proceedings of the IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2006, pp. 1129–1134.

[14] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the k user interference channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[15] V. Chandrasekhar, J. Andrews, T. Muharemovic, Z. Shen, and A. Gatherer, "Power control in two-tier femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4316–4328, Aug. 2009.

[16] V. Chandrasekhar and J. G. Andrews, "Uplink capacity and interference avoidance for two-tier femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3498–3509, Jul. 2009.

[17] N. Chayat, "Tentative criteria for comparison of modulation methods," *Doc. IEEE 802.11-97/96*, pp. 1–97, 1997.

[18] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons, 2010.

[19] J. Choi, *Optimal Combining and Detection: Statistical Signal Processing for Communications*. Cambridge University Press, 2010.

[20] S. Christensen, R. Agarwal, E. Carvalho, and J. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.

[21] S. Chu and X. Wang, "Opportunistic and cooperative spatial multiplexing in MIMO ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1610–1623, Oct. 2010.

[22] H. Claussen and F. Pivit, "Femtocell coverage optimization using switched multi-element antennas," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2009, pp. 1–6.

[23] L. Cortés-Peña, J. Barry, and D. Blough, "Joint optimization of stream allocation and beamforming and combining weights for the MIMO interference channel," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 4012–4018.

[24] L. Cortés-Peña, J. Barry, and D. Blough, "Jointly optimizing stream allocation, beamforming and combining weights for the MIMO interference channel," *IEEE Transactions on Wireless Communications*, Dec. 2013, Submitted.

[25] L. Cortés-Peña, J. Barry, and D. Blough, "The performance loss of unilateral interference cancellation," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 4181–4186.

[26] L. Cortés-Peña and D. Blough, "Achieving high performance and fairness in dense wireless networks of MIMO links," *IEEE/ACM Transactions on Networking*, 2014, In progress.

[27] L. Cortés-Peña and D. Blough, "Achieving high performance and fairness in dense wireless networks with MIMO interference cancellation," in *Proceedings of the ACM International Symposium Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Submitted, 2014.

[28] L. Cortés-Peña and D. Blough, "Distributed MIMO interference cancellation for interfering wireless networks: protocol and initial simulations," in *Technical Report GIT-CERCS-13-02*, Available: `http://www.cercs.gatech.edu/tech-reports`, Feb. 2013.

[29] COST 231 TD(973)119-REV 2 (WG2): "Urban transmission loss models for mobile radio in the 900- and 1,800- mhz nands,"

[30] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "The impact of channel bonding on 802.11n network management," in *Proceedings of the ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2011.

[31] C. Demichelis and P. Chimento, "IP packet delay variation metric for IP performance ¡etrics (IPPM)," 2002, Available: `http://tools.ietf.org/html/rfc3393`.

[32] M. Demirkol and M. Ingram, "Control using capacity constraints for interfering MIMO links," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, vol. 3, Sep. 2002, pp. 1032–1036.

[33] M. Demirkol and M. Ingram, "Power-controlled capacity for interfering MIMO links," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, May 2001, pp. 187–191.

[34] M. Demirkol and M. Ingram, "Stream control in networks with interfering MIMO links," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 1, Mar. 2003, pp. 343–348.

[35] R. Etkin, D. Tse, and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5534–5562, Dec. 2008.

[36] Y. Fang, D. Gu, A. McDonald, and J. Zhang, "A two-level carrier sensing mechanism for overlapping BSS problem in WLAN," in *Proceedings of the IEEE International Workshop on Local and Metropolitan Area Networks (LANMAN)*, Sep. 2005, pp. 1–6.

[37] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, Mar. 1998.

[38] H. T. Friis, "A note on a simple transmission formula," *Proc. IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[39] I.-T. R. G.114, "One-way transmission time," *International Telecommunications Union (ITU)*, vol. 18, Feb. 1996.

[40] V. Garg, *Wireless Communications and Networking*, First. Morgan Kaufmann, 2007.

[41] D. Gesbert, M. Shafi, D. Shiu, P. 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, pp. 281–302, Apr. 2003.

[42] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," vol. 39, no. 4, pp. 159–170, Oct. 2009.

[43] K. Gomadam, V. Cadambe, and S. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3309–3322, Jun. 2011.

[44] K. Gomadam, V. Cadambe, and S. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Dec. 2008, pp. 1–6.

[45] M. Guillaud, D. Slock, and R. Knopp, "A practical method for wireless channel reciprocity exploitation through relative calibration," *Proceedings of the IEEE*

International Symposium on Signal Processing and its Applications (CSPA), 2005.

[46] B. Hamdaoui and K. Shin, "Characterization and analysis of multi-hop wireless MIMO network throughput," in *Proceedings of the ACM International Symposium Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Sep. 2007, pp. 120–129.

[47] B. Han, L. Ji, S. Lee, R. Miller, and B. Bhattacharjee, "Channel access throttling for overlapping BSS management," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2009, pp. 1–6.

[48] C. Hellings, D. A. Schmidt, and W. Utschick, "Optimized beamforming for the two stream MIMO interference channel at high SNR," in *Proceedings of the International ITG Workshop on Smart Antennas (WSA)*, 2009.

[49] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, vol. 2, Apr. 2003, pp. 836–843.

[50] "IEEE std. 802.11-2012," Mar. 2012.

[51] "IEEE std. 802.11n-2009: enhancements for higher throughput," Oct. 2009, `http://www.ieee802.org/`.

[52] R. Iltis, S. Kim, and D. Hoang, "Noncooperative iterative MMSE beamforming algorithms for ad hoc networks," *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 748–759, Apr. 2006.

[53] W. C. Jakes and D. C. Cox, *Microwave Mobile Communications*. John Wiley & Sons, 1994.

[54] H.-S. Jo, C. Mun, J. Moon, and J.-G. Yook, "Interference mitigation using uplink power control for two-tier femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 4906–4910, Oct. 2009.

[55] M. Joham, K. Kusume, M. H. Gzara, W. Utschick, and J. A. Nossek, "Transmit wiener filter for the downlink of TDDDS-CDMA systems," in *International Symposium on Spread Spectrum Techniques and Applications (ISSSTA)*, IEEE, vol. 1, 2002, pp. 9–13.

[56] T. Kaiser, *Smart Antennas–State of the Art*. Hindawi Publishing Corporation, 2005.

[57] J. Ketchum, S Nanda, R Walton, *et al.*, "System description and operating principles for high throughput enhancements to 802.11," *IEEE 802.11-04/0870r0*, Aug. 2004.

[58] A. Khachan, A. Tenenbaum, and R. Adve, "Linear processing for the downlink in multiuser MIMO systems with multiple data streams," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 9, Jun. 2006, pp. 4113–4118.

[59] N. Khaled, G. Leus, C. Desset, and H. De Man, "A robust joint linear precoder and decoder MMSE design for slowly time-varying MIMO channels," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, May 2004, pp. IV–485IV–488–.

[60] S.-J. Kim and G. B. Giannakis, "Optimal resource allocation for MIMO ad hoc cognitive radio networks," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3117–3131, Apr. 2011.

[61] S. Kim, X. Wang, and M. Madihian, "Cross-layer design of wireless multihop backhaul networks with multiantenna beamforming," *IEEE Transactions on Mobile Computing*, pp. 1259–1269, Nov. 2007.

[62] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *Proceedings of the ACM SIGCOMM Conference*, Aug. 2013, pp. 387–398.

[63] W. C. Y. Lee, "Mobile communications design fundamentals," *John Wiley & Sons*, 1993.

[64] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," vol. 41, no. 4, pp. 146–157, Aug. 2011.

[65] J. Liu, Y. Shi, and Y. Hou, "A tractable and accurate cross-layer model for multi-hop MIMO networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, Mar. 2010, pp. 1–9.

[66] S. Lo, G. Lee, and W. Chen, "An efficient multipolling mechanism for IEEE 802.11 wireless LANs," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 764–778, Jun. 2003.

[67] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Third. John Wiley & Sons, 2007, Available: `http://www.janmagnus.nl/misc/mdc2007-3rdedition.pdf`.

[68] N. Malpani, J. Welch, and N. Vaidya, "Leader election algorithms for mobile ad hoc networks," in *Proceedings of the ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000, pp. 96–103.

[69] S. Mangold, S. Choi, P. May, and G. Hiertz, "IEEE 802.11e - fair resource sharing between overlapping basic service sets," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2002, pp. 166–171.

[70] A. Molisch, *Wireless Communications*. John Wiley & Sons, 2005.

[71] A. Molisch, "A generic model for MIMO wireless propagation channels in macro- and microcells," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 61–71, Jan. 2004.

[72] F. Negro, S. Shenoy, I. Ghauri, and D. Slock, "On the MIMO interference channel," in *Information Theory and Applications Workshop*, Feb. 2010, pp. 1–9.

[73] F. Negro, S. Shenoy, I. Ghauri, and D. Slock, "Weighted sum rate maximization in the MIMO interference channel," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2010, pp. 684–689.

[74] Y. Okumura, E. Ohmori, T. Kawano, and K. Fukuda, "Field strength and its variability in VHF and UHF land-mobile radio service," *Electrical Communication Laboratory*, vol. 16, no. 9, pp. 825–873, 1968.

[75] J. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: enabling spatial reuse using MIMO channel-aware MAC," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 5, May 2005, pp. 3642–3646.

[76] M. Park, S. Choi, and S. Nettles, "Cross-layer MAC design for wireless networks using MIMO," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, vol. 5, Dec. 2005, pp. 2870–2874.

[77] D. Peleg, "Time-optimal leader election in general networks," *Journal of Parallel and Distributed Computing*, vol. 8, no. 1, pp. 96–99, 1990.

[78] S. Peters and R. Heath, "Cooperative algorithms for MIMO interference channels," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 206–218, Jan. 2011.

[79] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*, Available: `http://orion.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`, Nov. 2012.

[80] "Quality of service for voice over IP," *Cisco*, Jun. 2001, Available: `http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf`.

[81] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: scaling wireless capacity with user demands," in *Proceedings of the ACM SIGCOMM Conference*, Aug. 2012, pp. 235–246.

[82] F. Rashid-Farrokhi, K. Liu, and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1437–1450, Oct. 1998.

[83] M. Razaviyayn, M. Sanjabi, and Z. Luo, "Linear transceiver design for interference alignment: complexity and computation," *IEEE Transactions on Information Theory*, vol. 58, no. 5, May 2012.

[84] H. Sampath, P. Stoica, and A. Paulraj, "Generalized linear precoder and decoder design for MIMO channels using the weighted MMSE criterion," *IEEE Transactions on Communications*, vol. 49, no. 12, pp. 2198–2206, Dec. 2001.

[85] D. Schmidt, W. Utschick, and M. Honig, "Beamforming techniques for single-beam MIMO interference networks," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Oct. 2010, pp. 1182–1187.

[86] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 3, no. 27, pp. 379–423, 1948.

[87] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the Institute of Radio Engineers*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[88] H. Shen, B. Li, M. Tao, and X. Wang, "MSE-based transceiver designs for the MIMO interference channel," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3480–3489, Nov. 2010.

[89] C. Shi, D. Schmidt, R. Berry, M. Honig, and W. Utschick, "Distributed interference pricing for the MIMO interference channel," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2009, pp. 1–5.

[90] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[91] S. Shi, M. Schubert, N. Vucic, and H. Boche, "MMSE optimization with per-base-station power constraints for network MIMO systems," in *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2008, pp. 4106–4110.

[92] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y. Hou, "An optimal link layer model for multi-hop MIMO networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2011, pp. 1916–1924.

[93] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y Hou, "A dof-based link layer model for multi-hop MIMO networks," *IEEE Transactions on Mobile Computing*, 2014, To appear.

[94] R. Srinivasan, D. Blough, and P. Santi, "Optimal one-shot stream scheduling for MIMO links in a single collision domain," in *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Jun. 2009, pp. 1–9.

[95] R. Srinivasan, D. Blough, L. Cortés-Peña, and P. Santi, "Maximizing throughput in MIMO networks with variable rate streams," in *Proceedings of European Wireless (EW) Conference*, Apr. 2010, pp. 551–559.

[96] G. Stüber, *Principles of Mobile Communication*, Second. Springer Netherlands, 2001.

[97] G. L. Stuber, J. R. Barry, S. W. Mclaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband mimo-ofdm wireless communications," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 271–294, Feb. 2004.

[98] K. Sundaresan, R. Sivakumar, M. Ingram, and T. Chang, "Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 350–365, Dec. 2004.

[99] K. Sundaresan, W. Wang, and S. Eidenbenz, "Algorithmic aspects of communication in ad-hoc networks with smart antennas," in *Proceedings of the ACM International Symposium Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006, pp. 298–309.

[100] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," *Proceedings of the ACM SIGCOMM Conference*, vol. 40, no. 4, pp. 147–158, Oct. 2010.

[101] E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, Dec. 1999.

[102] "The ns-3 network simulator," `http://www.nsnam.org`.

[103] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, and D. Towsley, "Leader election algorithms for wireless ad hoc networks," in *Proc. DARPA Info. Survivability Conf. and Exposition*, vol. 1, Apr. 2003, pp. 261–272.

[104] S. Vasudevan, J. Kurose, and D. Towsley, "Design and analysis of a leader election algorithm for mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, IEEE, 2004, pp. 350–360.

[105] E. Visotsky and U. Madhow, "Space-time transmit precoding with imperfect feedback," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2632–2639, Sep. 2001.

[106] J. H. Winters, "On the capacity of radio communication systems with diversity in a rayleigh fading environment," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 5, pp. 871–878, Jun. 1987.

[107] B. Xi, H. Chen, W. S. Cleveland, T. Telkamp, *et al.*, "Statistical analysis and modeling of internet VoIP traffic for network engineering," *Electronic Journal of Statistics*, vol. 4, pp. 58–116, Jan. 2010.

[108] S. Ye and R. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2839–2848, Nov. 2003.

[109] C. Yetis, T. Gou, S. Jafar, and A. Kayran, "Feasibility conditions for interference alignment," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Dec. 2009, pp. 1–6.

[110] C. Yetis, T. Gou, S. Jafar, and A. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4771–4782, Sep. 2010.

[111] J. Yoon, M. Y. Arslan, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee, "A distributed resource management framework for interference mitigation in OFDMA femtocell networks," in *Proceedings of the ACM International Symposium Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2012, pp. 233–242.

[112] L. Zheng and D. Hoang, "Applying graph coloring in resource coordination for a high-density wireless environment," in *Proceedings of the IEEE International Conference on Computer and Information Technology (CIT)*, Jul. 2008, pp. 664–669.