On Interference Management for Wireless Networks

Huacheng Zeng

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Y. Thomas Hou, Chair
R. Michael Buehrer
Wenjing Lou
Jeffrey H. Reed
Hanif D. Sherali
Yi Shi

January 30, 2015
Blacksburg, Virginia

Keywords: Wireless networks, interference cancellation, interference alignment, interference neutralization, modeling and optimization, algorithm design

On Interference Management for Wireless Networks

Huacheng Zeng

ABSTRACT

Interference is a fundamental problem in wireless networks. An effective solution to this problem usually calls for a cross-layer approach. Although there exist a large volume of works on interference management techniques in the literature, most of them are limited to signal processing at the physical (PHY) layer or information-theoretic exploitation. Studies of advanced interference techniques from a cross-layer optimization perspective remain limited, especially involving multi-hop wireless networks. This dissertation aims at filling this gap by offering a comprehensive investigation of three interference techniques: interference cancellation (IC), interference alignment (IA), and interference neutralization (IN).

This dissertation consists of three parts: the first part studies IC in distributed multi-hop multiple-input multiple-output (MIMO) networks; the second part studies IA in multi-hop networks, cellular networks, and underwater acoustic (UWA) networks; and the third part focuses on IN in multi-hop single-antenna networks. While each part makes a step towards advancing an interference technique, they collectively constitute a body of work on interference management in the networking research community. Results in this dissertation not only advance network-level understanding of the three interference management techniques, but also offer insights and guidance on how these techniques may be incorporated in upper-layer protocol design.

In the first part, we study IC in multi-hop MIMO networks where resource allocation is achieved through neighboring node coordination and local information exchange. Based on a well-established degree-of-freedom (DoF) MIMO model, we develop a distributed DoF scheduling algorithm with the objective of maximizing network-level throughput while guaranteeing solution feasibility at the PHY layer. The proposed algorithm accomplishes a number of beneficial features, including polynomial-time complexity, amenability to local

implementation, a guarantee of feasibility at the PHY layer, and competitive throughput performance. Our results offer a definitive "yes" answer to the question — Can the node-ordering DoF model be deployed in a distributed multi-hop MIMO network? In particular, we show that the essence of the DoF model — a global node ordering, can be implicitly achieved via local operations, albeit it is invisible to individual node.

In the second part, we investigate IA in various complex wireless networks from a networking perspective. Specifically, we study IA in three different domains: spatial domain, spectral domain, and temporal domain.

- In the spatial domain, we study IA for multi-hop MIMO networks. We derive a set of simple constraints to characterize the IA capability at the PHY layer. We prove that as long as the set of simple constraints are satisfied, there exists a feasible IA scheme (i.e., precoding and decoding vectors) at the PHY layer so that the data streams on each link can be transported free of interference. Therefore, instead of dealing with the complex design of precoding and decoding vectors, our IA constraints only require simple algebraic addition/subtraction operations. Such simplicity allows us to study network-level IA problems without being distracted by the tedious details in signal design at the PHY layer. Based on these IA constraints, we develop an optimization framework for unicast and multicast communications.
- In the spectral domain, we study IA in OFDM-based cellular networks. Different from spatial IA, spectral IA is achieved by mapping data streams onto a set of frequency bands/subcarriers (rather than a set of antenna elements). For the uplink, we derive a set of simple IA constraints to characterize a feasible DoF region for a cellular network. We show how to construct precoding and decoding vectors at the PHY layer so that each data stream can be transported free of interference. Based on the set of IA constraints, we study a user throughput maximization problem and show the throughput improvement over two other schemes via numerical results. For the downlink, we find that we can exploit the uplink IA constraints to the downlink case simply by reversing the roles of user and base station. Further, the downlink user throughput maximiza-

tion problem has the exactly same formulation as the uplink problem and thus can be solved in the exactly same way.

- In the temporal domain, we study IA for UWA networks. A fundamental issue in UWA networks is large propagation delays due to slow signal speed in water medium. But temporal IA has the potential to turn the adverse effect of large propagation delays into something beneficial. We propose a temporal IA scheme based on propagation delays, nicknamed PD-IA, for multi-hop UWA networks. We first derive a set of PD-IA constraints to guarantee PD-IA feasibility at the PHY layer. Then we develop a distributed PD-IA scheduling algorithm, called Shark-IA, to maximally overlap interference in a multi-hop UWA network. We show that PD-IA can turn the adverse propagation delays to throughput improvement in multi-hop UWA networks.

In the third part, we study IN for multi-hop single-antenna networks with full cooperation among the nodes. The fundamental problem here is node selection for IN in a multi-hop network environment. We first establish an IN reference model to characterize the IN capability at the PHY layer. Based on this reference model, we develop a set of constraints that can be used to quickly determine whether a subset of links can be active simultaneously. By identifying each eligible neutralization node as a neut, we study IN in a multi-hop network with a set of sessions and derive the necessary constraints to characterize neut selection, IN, and scheduling. These constraints allow us to study IN problems from a networking perspective but without the need of getting into signal design issues at the PHY layer. By applying our IN model and constraints to study a throughput maximization problem, we show that the use of IN can generally increase network throughput. In particular, throughput gain is most significant when there is a sufficient number of neuts that can be used for IN.

In summary, this dissertation offers a comprehensive investigation of three interference management techniques (IC, IA, and IN) from a networking perspective. Theoretical and algorithmic contributions of this dissertation encompass characterization of interference exploitation capabilities at the PHY layer, derivation of tractable interference models, development of feasibility proof for each interference model, formulation of throughput maximization

problems, design of distributed IC and PD-IA scheduling algorithms, and development of near-optimal solutions with a performance guarantee. The results in this dissertation offer network-level understanding of the three interference management techniques and lay the groundwork for future research on interference management in wireless networks.

Acknowledgments

In the past five years, I have benefited a lot from my interactions with many special individuals. First and foremost, I would like to thank my advisor Professor Tom Hou for his guidance and support throughout the course of my graduate studies. Professor Hou had always been encouraging me to think more deeply and more logically when I was mumbling vague ideas with him. He spent countless days and nights on helping me revise every sentence in my papers to make it correct, coherent, and elegant. What I learned from him is not just a solution to a problem, but his inspirations, his way of thinking, and his way of conducting research. It is his pursuit of scholarship at the highest level that led me to develop my mindset and skills to a level that I would never be able to achieve. His unique blend of research vision, work passion, and professional dedication serves an inspiring role model for my career.

I would like to thank Dr. Yi Shi for serving on my advisory committee. I am indebted to him for many helpful and simulating discussions on mathematical modeling of interference cancellation.

I also want to thank Professor Hanif Sherali for serving as a member of my committee and as a reader of this dissertation. His valuable feedback helps me improve this dissertation in every respect.

I am grateful to Professor Wenjing Lou, Professor Michael Buehrer, and Professor Jeffrey Reed for serving on my advisory committee and making insightful comments on this dissertation.

My gratitude extends to my colleagues Qiben Yan, Xu Yuan, Xiaoqi Qin, and Changlai

Du in the Complex Network and Security Research (CNSR) Lab for their fruitful collaboration with me. I also thank former and present members of the CNSR Lab, including Sushant Sharma, Canming Jiang, Liguang Xie, Borhan Jalaeian, Amr Nabil, Xiangwei Zheng, Rongbo Zhu, Lili Zhang, An Li, Nan Jiang, and Feng Tian.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. This work was supported in part by the U.S. National Science Foundation (NSF) and the Office of Naval Research (ONR). I also acknowledge Advanced Research Computing (ARC) at Virginia Tech for providing me with the computing service on powerful supercomputer BlueRidge.

I would like to thank my friends who helped and assisted me in the past five years. I thank my roommate Xiaokui Shu for sharing his Internet security knowledge and photography skills with me. I also thank my friends Jian Zhang, Hao Zhang, Qing Li for spending time with me on food and games. My thanks are due to my friends Qiong Huo and Aolin Xu for sharing their life and work experience with me. I would like to give my special and sincere thanks to Ning Xie for her tremendous support and encouragement during this journey.

Last, but not the least, I would like to thank my family, my parents, my sisters, and my brother, for supporting me unconditionally. No words can adequately express my gratitude to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Interference Management Techniques

In wireless networks, interference refers to the presence of undesired radio signals at an active receiver. The existence of interference makes it difficult for the receiver to successfully decode its desired signal, thereby reducing its achievable data rate. Interference in wireless networks can be attributed to the broadcast nature of wireless channel — the radio signal from a transmitter can reach not only its intended receivers but also its neighboring unintended receivers. Interference is a fundamental problem in wireless networks. To address the adverse effects of interference, the research community has been exploring interference management techniques for many decades. These efforts have produced a large volume of results for a broad class of wireless networks. At a coarse level, the existing interference management techniques can be mainly divided into two groups: interference avoidance and interference exploitation.

Interference avoidance is a traditional approach to manage interference in wireless networks. As its name suggests, this approach avoids interference to occur by assigning different data transmissions to orthogonal channels. It lays the foundation for transmission scheduling in many wireless networks, such as cellular networks (e.g., TDMA, FDMA, CDMA) and WiFi networks (e.g., CSMA). Although this approach is straightforward and easy to

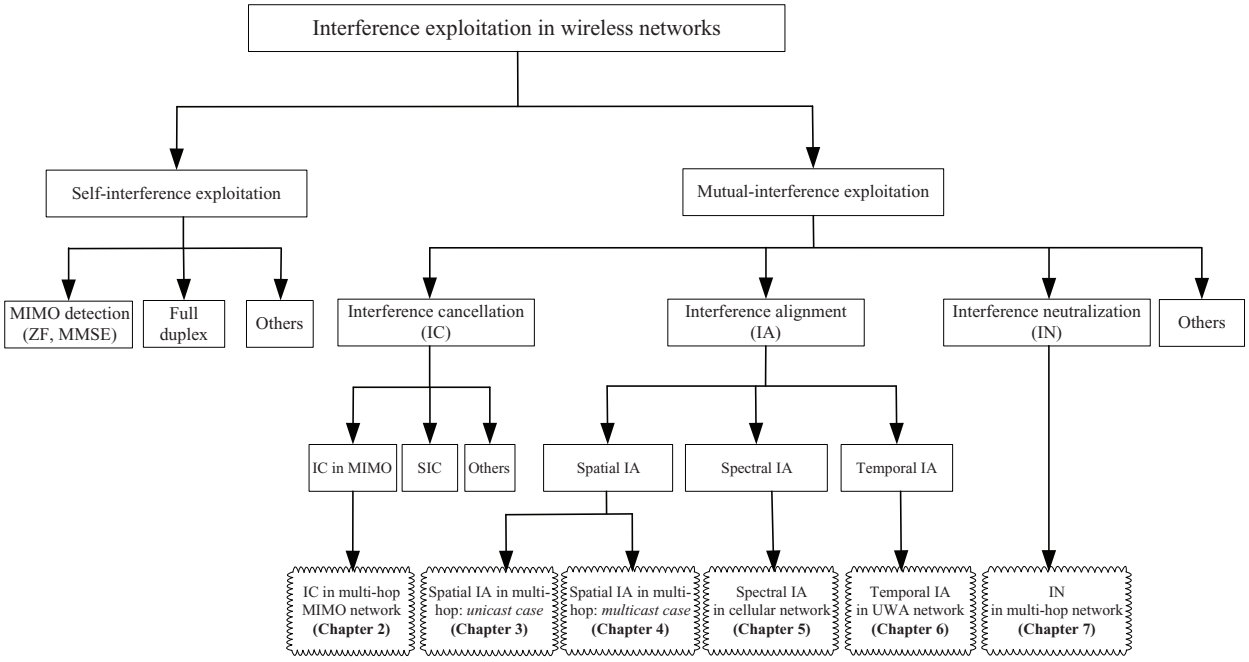


Figure 1.1: Interference management techniques and the structure of this dissertation.

implement, its performance is far from the optimum. Under this approach, the achievable throughput of each user decreases linearly with the total number of active users in the network. As such, interference avoidance is not regarded as an efficient approach to manage interference, albeit it is widely used in today's commercial wireless networks.

Interference exploitation is an advanced approach to manage interference in wireless networks. It allows simultaneous transmissions in the presence of interference. As shown in Fig. 1.1, the existing interference exploitation techniques can be roughly classified into two categories: self-interference exploitation and mutual-interference exploitation. Self-interference exploitation refers to intelligent suppression of intra-link or intra-node interference. Examples of self-interference exploitation include zero-forcing detection in a multiple-input multiple-output (MIMO) link and self-interference cancellation in a full-duplex radio. Zero-forcing detection allows an MIMO link to transport multiple data streams concurrently and self-interference cancellation allows a full-duplex node to transmit and receive radio signals simultaneously. Mutual-interference exploitation refers to intelligent mitigation of inter-link or inter-node interference. The basic idea of mutual-interference exploitation is to

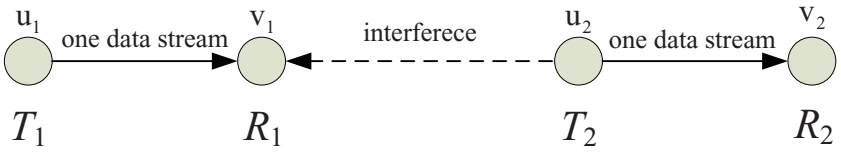


Figure 1.2: A two-link example that illustrates IC at either transmitter or receiver.

manage mutual interference via a sophisticated signal design at transmitters and/or receivers, so that each receiver can successfully decode its desired signals in the presence of mutual interference. Examples of mutual-interference exploitation include interference cancellation, interference alignment, and interference neutralization. These three mutual-interference exploitation techniques constitute the subject of this dissertation and we summarize them as follows.

Interference Cancellation (IC). IC is a classic interference management technique in wireless networks. It mitigates mutual interference through a sophisticated signal design at the physical (PHY) layer. For a mutual interference, IC can be done either at the transmitter side or at the receiver side. That is, an interference can be canceled either by a design of its precoding vectors at its transmitter or a design of its decoding vectors at its receiver. Consider the two links in Fig. 1.2 as an example. In this figure, a solid arrow line represents data transmission and a dashed arrow line represents an interference. Suppose that each node has two antennas and each transmission carries one data stream. Then, each transmitter has a 2×1 precoding vector for its data stream transmission and each receiver has a 2×1 decoding vector for its desired data stream reception. For the interference from T_2 to R_1, it can be canceled either at transmitter T_2 or at receiver R_1. Denote $\mathbf{u}_i$ as the precoding vector at transmitter T_i and $\mathbf{v}_j$ as the decoding vector at receiver R_j. Denote $\mathbf{H}_{ji}$ as the spatial channel matrix between receiver R_j and transmitter T_i. To cancel the interference at R_1 by transmitter T_2, we can construct the precoding vector at transmitter T_2 as follows:

$$\mathbf{u}_2 = \begin{bmatrix} (\mathbf{v}_1)^T\mathbf{H}_{12} \\ (\mathbf{v}_2)^T\mathbf{H}_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

To cancel the interference at receiver R_1 by receiver R_1 itself, we can construct the decoding

vector at receiver R_1 as follows:

$$\mathbf{v}_1 = \left([1\ 0][\mathbf{H}_{11}\mathbf{u}_1 \quad \mathbf{H}_{12}\mathbf{u}_2]^{-1}\right)^T .$$

Since an interference can be canceled either at its transmitter or at its receiver, a natural question to ask is, for a network with a lot of interferences, how to assign the IC responsibility for each interference? Actually, such a problem is not trivial as an arbitrary assignment of IC responsibility is likely to result in an infeasible solution at the PHY layer. Recently, Shi et al. in [78] addressed this problem by developing a node order-based degree-of-freedom (DoF) model. This novel node-ordering concept allows a systematic assignment of IC responsibilities for each interference. It was shown in [78] that, as long as the IC is done following a node-ordering, the resulting solution is guaranteed to be feasible at the PHY layer. Such a DoF model allows us to study IC for networking problems without the need of getting into the complex design of precoding and decoding vectors at the PHY layer.

Interference Alignment (IA). IA is widely regarded as a major advance in interference management in recent years. It offers a new direction to handle mutual interference in wireless networks. The basic idea of IA is to jointly construct signals at transmitters so that these signals overlap at their unintended receivers while remaining resolvable at their intended receivers. Hence, IA is a transmitter-side interference management technique and requires coordination among the transmitters. Since its inception, IA has gained tremendous momentum and has been applied to a variety of channels/networks. The most significant result of IA was developed by Camade and Jafar in [9]. They showed that the use of IA could achieve $K/2$ DoFs in the K-user interference channel. In addition to its theoretical advance, IA was also studied in practical networks. Gollakotta et al. [23] demonstrated experimentally that the use of IA could increase the average throughput by 1.5 times for the downlink and 2 times for the uplink in a 2×2 MIMO WLAN. El Ayach et al. [15] implemented IA in an MIMO-OFDM testbed and their experimental results showed a considerable throughput gain.

To see how IA works in MIMO networks, we consider the three links in Fig. 1.3. In the figure, a solid arrow line represents data transmission and a dashed arrow line represents

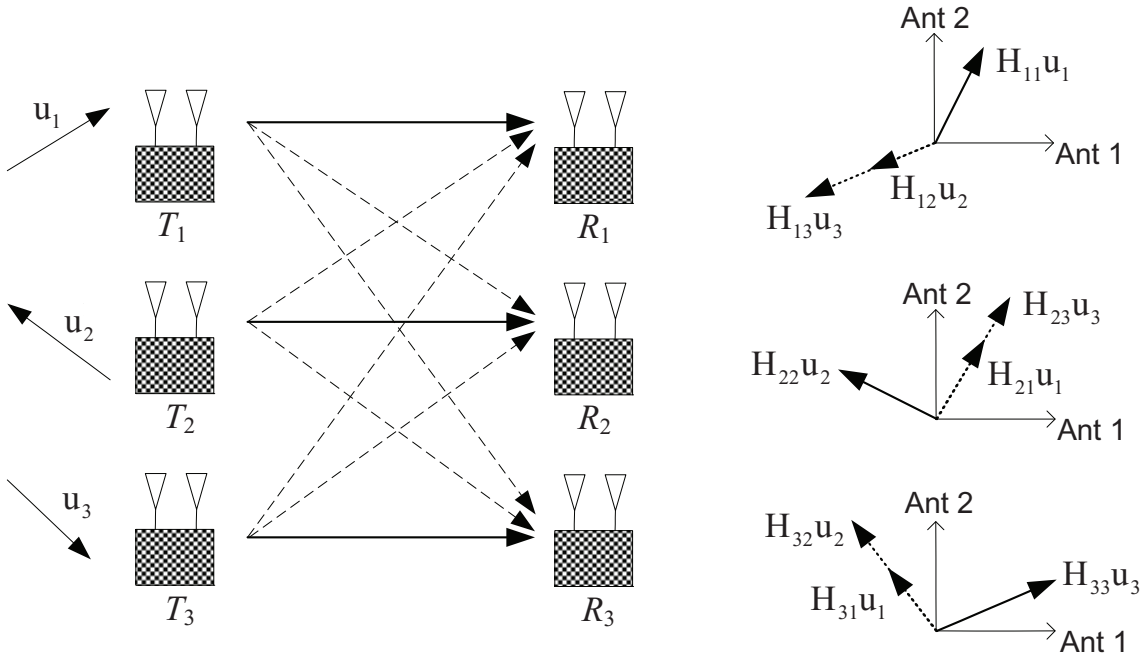


Figure 1.3: An example that illustrates IA in an MIMO network.

interference. Each node has two antennas and all nodes are in the same collision domain. When IA is not employed, at most two independent data streams can be transported on the three links. In contrast, when IA is employed, three independent data streams can be transported on the three links, with one data stream on each link. To see how this is possible, consider the received data stream and interfering streams at each receiver. Since each receiver has only two antennas, it can decode its desired data stream free of interference only if its two interfering streams are aligned in the same direction. Denote $\mathbf{u}_i$ as the precoding vector at transmitter T_i. Denote $\mathbf{H}_{ji}$ as the spatial channel matrix between receiver R_j and transmitter T_i. Then, one precoding scheme that can achieve the desired IA at all three receivers is as follows:

$$\mathbf{u}_1 = \text{eigvec}(\mathbf{H}_{21}^{-1}\mathbf{H}_{23}\mathbf{H}_{13}^{-1}\mathbf{H}_{12}\mathbf{H}_{32}^{-1}\mathbf{H}_{31}) \ ,$$
$$\mathbf{u}_2 = \mathbf{H}_{32}^{-1}\mathbf{H}_{31}\mathbf{u}_1 \ ,$$
$$\mathbf{u}_3 = \mathbf{H}_{23}^{-1}\mathbf{H}_{21}\mathbf{u}_1 \ ,$$

where eigvec($\mathbf{H}$) is an eigenvector of square matrix $\mathbf{H}$. It can be verified that by using the

above precoding vectors, the two interfering streams at each receiver will be aligned in the same direction, as shown in the figure. Denote $\mathbf{v}_j$ as the decoding vector of the desired data stream at receiver R_j. To decode the desired data stream at each receiver, one may use the following decoding vectors:

$$\mathbf{v}_1 = \left([1\ 0][\mathbf{H}_{11}\mathbf{u}_1\ \mathbf{H}_{12}\mathbf{u}_2]^{-1}\right)^T,$$
$$\mathbf{v}_2 = \left([0\ 1][\mathbf{H}_{21}\mathbf{u}_1\ \mathbf{H}_{22}\mathbf{u}_2]^{-1}\right)^T,$$
$$\mathbf{v}_3 = \left([0\ 1][\mathbf{H}_{31}\mathbf{u}_1\ \mathbf{H}_{33}\mathbf{u}_3]^{-1}\right)^T.$$

One can verify that the above precoding and decoding vectors satisfy the zero-forcing IC requirement, i.e.,

$$(\mathbf{v}_j)^T\mathbf{H}_{ji}\mathbf{u}_i = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, each link can transport one data stream free of interference.

While the example in Fig. 1.3 illustrated how IA can be achieved in the spatial domain, IA can also be achieved in the spectral (frequency) and temporal (time) domains. In the spectral domain, IA is achieved by mapping data streams onto multiple orthogonal frequency bands (a.k.a. subcarriers). Unlike spatial IA, spectral IA does not require multiple antennas at each node. Instead, it requires multiple orthogonal subcarriers available for data transmissions in the network. As such, spectral IA is mainly considered for OFDM networks, which has plenty of orthogonal frequency subcarriers.

In the temporal domain, there exist different approaches to achieve IA. One of them is to achieve IA based on signal propagation delays and is called PD-IA. PD-IA relies on a joint scheduling of transmit signals at different transmitters so that at each receiver, the interfering signals overlap in some time intervals while the desired signals are free from interference in the temporal domain. A summary of the three forms of IA is given in Table 1.1. Details of IA in the spatial, spectral, and temporal domains will be given in Chapters 3–6.

Interference Neutralization (IN). The terminology of IN was recently coined by Mohajer et al. in [58, 59, 60], although a similar idea has been around for many years under different

Table 1.1: A summary of spatial-, temporal-, and spectral-IA.

	Application domain	CSI at TX	Coordination among TX	SNR requirement	Channel requirement
Spatial IA	MIMO networks	Required	Required	High SNR regime	Full rank and independent channel
Temporal IA	Networks with large propagation delays (e.g., UWA networks)	Required	Required	Any SNR regime	No requirement
Spectral IA	OFDM networks	Required	Required	High SNR regime	Full rank and frequency selective channel

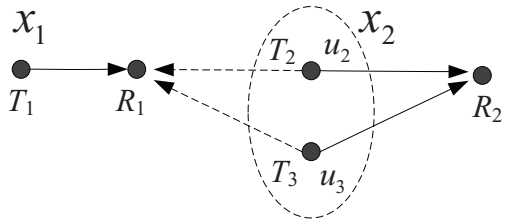


Figure 1.4: An example that illustrates the basic idea of IN.

names such as distributed MIMO and multiuser zero-forcing (see, e.g., [1, 12]). The basic idea of IN is to jointly design the signals at transmitters, so that these transmit signals nullify themselves in the air[1] at their unintended receivers while remaining resolvable at their intended receivers. Hence, IN is a transmitter-side interference management technique. To achieve interference nullification in the air, IN requires that multiple transmitters have the same data information under transmission. But this can be easily accomplished in wireless networks, due to the broadcast nature of a wireless channel.

To illustrate the basic idea of IN, let's consider the network in Fig. 1.4. In this network, each node has a single antenna and there are two data transmissions: $T_1 \to R_1$ and $T_2 \to R_2$. Suppose that node T_3 has the same information as T_2 and can cooperate with T_2. Denote u_i as the precoding coefficient at transmitter T_i. Denote h_{ji} as the channel coefficient between receiver R_j and transmitter T_i. Then, IN can be achieved in this network through

[1]By "in the air" we mean that the signals nullify themselves at the radio wave level.

constructing the precoding coefficients at transmitters T_2 and T_3 as follows:

$$u_2 = h_{13} \quad \text{and} \quad u_3 = -h_{12} \ .$$

It is easy to verify that by using the above precoding coefficients, the interferences at receiver R_1 can be neutralized, i.e., $h_{12}u_2x_2 + h_{13}u_3x_2 = 0$, where x_2 is the outgoing data stream at transmitters T_2 and T_3. Meanwhile, the desired signal at receiver R_2 remains resolvable, i.e., $|h_{22}u_2x_2 + h_{23}u_3x_2| > 0$. As a result, the two data transmissions are transparent to each other and can occur simultaneously, albeit within the same interference domain.

1.2 Motivation and Goals

Interference exploitation has become a new paradigm for wireless networks. Although there is an increasing volume of new results on interference exploitation, most of them are limited to single-hop networks (e.g., cellular and WiFi networks) and two-hop networks (e.g., relay-aided cellular network). There has been very limited progress so far in the exploration and understanding of interference exploitation techniques from a networking perspective, particularly in the context of multi-hop networks. This stagnation can be attributed to the underlying technical challenges in a multi-hop network. Some of them are listed as follows:

- Interference exploitation always requires complex signal design at the PHY layer, as we showed for IC, IA, and IN in the previous section. For a large-scale wireless network with many nodes, it rapidly becomes intractable to design the signals at each transmitter and receiver for interference exploitation. A promising approach is to develop a tractable model that can characterize the interference exploitation capability at the PHY layer. Such a model, if successful, allows us to study network-level problem with the new interference exploitation capability without getting into the details of signal design at the PHY layer. However, developing such a simple model for an interference exploitation technique (e.g., IA and IN) is challenging and is our major pursuit in this dissertation.
- In a large-scale multi-hop wireless network, interference exploitation at the PHY layer

is interwoven with link connectivity, transmission scheduling, and information flow routing at the link/network layers. Due to such coupling, an approach that designs algorithm or protocol for each layer in isolation is neither plausible nor effective. Instead, a cross-layer approach with joint consideration of multiple layers is needed. However, joint design of interference exploitation with upper-layer algorithms is very challenging in a multi-hop network.

- For a multi-hop wireless network, a central controller and global coordination usually are not available. Coordination and information exchange may be limited to neighboring nodes in the network. In such a distributed multi-hop network, an interference exploitation scheme that is amenable to local implementation is desirable. How to design an interference exploitation scheme for a distributed multi-hop network that can still yield competitive performance is a challenging problem.

The goal of this dissertation is to make a concrete step towards advancing IC, IA, and IN for various wireless networks through characterizing their capabilities at the PHY layer, deriving their cross-layer optimization models, and developing their efficient distributed algorithms. Based on the interference techniques at the PHY layer, we divide this dissertation to three parts. In the first part, we address the IC problem in distributed multi-hop MIMO networks by developing an efficient DoF scheduling algorithm that is amenable to local implementation. In the second part, we offer a comprehensive investigation of IA by studying IA in the spatial, spectral, and temporal domains. In the third part, we study IN in multi-hop wireless networks with the objective of exploring how IN can increase network throughput.

1.3 Dissertation Outline and Contributions

This dissertation studies three interference management techniques (IC, IA, and IN) from a networking perspective. The organization of this dissertation is exhibited in Fig. 1.1. While each chapter individually makes a concrete step towards advancing an interference technique, they jointly constitute a body of work on interference management that contributes to the networking community. The main contributions of each chapter are summarized as follows:

- In Chapter 2, we study IC in multi-hop MIMO networks. In [78], a node-ordering DoF model was proposed to characterize MIMO's SM and IC capabilities for a multi-hop network. Although this DoF model has many merits, it hinges upon a global node ordering to keep track of IC responsibilities among the nodes. An open question about this model is whether its global ordering requirement can be achieved in a distributed network environment. In this chapter, we explore this question by studying IC in a multi-hop MIMO network. We propose an efficient scheduling algorithm to allocate DoF resources for SM and IC under the guidance of this node-ordering DoF model. We show that our algorithm is amenable to local implementation as it only requires neighboring node coordination and local information exchange. We prove that every solution found by our algorithm is guaranteed to be feasible at the PHY layer and show that there always exists a corresponding global node ordering for IC, albeit such a global node ordering is invisible at each node in the network. We also offer simulation results to show that the throughput performance of our distributed algorithm is close to that of a centralized perfect scheduling algorithm.
- In Chapter 3, we study spatial IA in multi-hop MIMO networks. Due to its intrinsic nature of being an interference management technique, most results of spatial IA are limited to point-to-point scenarios or single-hop networks. There is a lack of investigation of spatial IA from a networking perspective, especially in the context of multi-hop MIMO networks. In this chapter, we aim to fill in this gap by advancing spatial IA in multi-hop MIMO networks. We derive a set of simple IA constraints to characterize IA capabilities at the PHY layer. We prove that as long as these IA constraints are satisfied, there always exist precoding and decoding vectors at the PHY layer so that the data streams on each link can be transported free of interference. Based on these IA constraints, we develop an IA optimization framework for a multi-hop MIMO network. Such an IA optimization framework allows us to study a wide array of network-level throughput problems without being distracted by the complex design of precoding and decoding vectors at the PHY layer. As an application of this IA framework, we

study a specific throughput problem — maximizing the minimum achievable throughput among a set of unicast sessions. We demonstrate this throughput maximization problem can be optimally solved and offer simulation results to show the significant throughput gain.

- In Chapter 4, we extend the study of spatial IA from unicast communication to multicast communication. This study is motivated by two observations: (i) the power of IA is most profound when there is plenty of interference in the network; and (ii) multicast communication is a scenario that is associated with a lot of interference. We derive a set of multicast IA constraints to characterize the IA capability in multicast communication. These multicast IA constraints are very simple as they only require algebraic addition and subtraction operations. We show that as long as these simple multicast IA constraints are satisfied, there always exist precoding and decoding vectors at the PHY layer so that the data streams on each multicast link can be transported free of interference. As such, these multicast IA constraints allow us to study network-level throughput problems without being distracted by the complex design of precoding and decoding at the PHY layer. Based on the multicast IA constraints, we formulate a multicast throughput maximization problem. To solve this problem, we develop mathematical linearization techniques to eliminate the nonlinear constraints while maintaining a $(1-\varepsilon)$-optimality of the original problem. We offer simulation results to show that the use of IA can significantly increase multicast throughput and the throughput gain of IA increases with the volume of multicast traffic.
- In Chapter 5, we study spectral IA in OFDM-based cellular networks. Recent results in information theory have shown a huge potential of spectral IA in cellular networks. However, these results rely on some strong assumptions such as one data stream at each user, infinitely many frequency bands, infinitely many users in a cell, and a single interference domain, which prevent these IA results from being used in practical cellular networks. This chapter aims to relax those strong assumptions by studying spectral IA for OFDM-based cellular networks with more practical settings. For the uplink,

we propose a set of simple IA constraints to characterize a feasible DoF region for a cellular network. We prove that, as long as the set of simple IA constraints are satisfied, one can always construct precoding and decoding vectors at the PHY layer so that the data streams from each user can be transported free of interference to its base station. Based on the set of simple IA constraints, we study a user throughput maximization problem with the consideration of base station selection for each user. We offer simulation results to show the significant throughput gain of IA when compared to other cases. For the downlink, we show that the set of simple IA constraints derived for the uplink can be applied to the downlink by reversing the roles of user and base station. Moreover, the downlink user throughput maximization problem has the same formulation as the uplink problem and thus can be solved in the exactly same way.

- In Chapter 6, we study temporal IA in underwater acoustic (UWA) networks. In UWA networks, a fundamental issue is large signal propagation delays caused by slow signal travel speed in water medium. A new direction to address this issue is to take advantage of large propagation delays rather than considering them as solely a disadvantage. Recent advances in temporal IA promise a great potential to turn the adverse effect of large propagation delays into something that is beneficial to throughput improvement. In this chapter, we propose a temporal IA scheme based on propagation delays, nicknamed PD-IA, for multi-hop UWA networks. In this work, we first derive a set of PD-IA constraints to ensure PD-IA feasibility at the PHY layer. Based on these PD-IA constraints, we propose a distributed PD-IA scheduling algorithm, called Shark-IA, to maximally overlap interference in a multi-hop UWA network. Our proposed algorithm has a number of merits, including polynomial-time computational complexity, amenability to local implementation, and a guarantee of feasibility at the PHY layer. We offer simulation results to show that the use of PD-IA can turn the adverse propagation delays to throughput improvement in multi-hop UWA networks. Further, the throughput gain increases with the traffic volume in the network.
- In Chapter 7, we study IN for multi-hop single-antenna networks under the assumption

of full cooperation among all nodes in the network. We first establish an IN reference model to characterize the IN capability at the PHY layer. Based on this reference model, we develop a set of constraints that can quickly determine whether a subset of links can be active simultaneously. By introducing a new concept called "neut", which is used to identify an idle node that can be used for neutralization, we study IN in a multi-hop network with a set of independent sessions. We derive a set of necessary constraints to characterize neut selection, IN, and scheduling. These constraints allow us to study IN problems from a networking perspective but without being distracted by the complex signal design at the PHY layer. Finally, we apply these IN constraints to study a throughput maximization problem. We show that the use of IN can help increase network throughput. Further, throughput gain of IN is most significant when there is a sufficient number of neuts in the network.

In summary, this dissertation offers a comprehensive investigation of three interference management techniques (IC, IA, and IN) from a networking perspective. Theoretical and algorithmic contributions of this dissertation encompass characterization of interference exploitation capabilities at the PHY layer, derivation of tractable interference models, development of feasibility proof for each interference model, formulation of throughput maximization problems, design of distributed IC and PD-IA scheduling algorithms, and development of near-optimal solutions with a performance guarantee. The results in this dissertation offer network-level understanding of the three interference management techniques and lay the groundwork for future research on interference management in wireless networks.

Chapter 2

A Scheduling Algorithm for MIMO DoF Allocation

2.1 Introduction

In recent years, MIMO has attracted a growing interest in the wireless networking research community due to its ability to offer significant increases in data throughput without additional bandwidth or transmit power [90]. Among the research efforts of MIMO in multi-hop networks, there is an active research line that builds upon the so-called degree-of-freedom (DoF) model [5, 7, 17, 29, 44, 63, 64, 68, 86, 87]. The concept of DoF was originally defined to represent the maximum multiplexing gain of MIMO channel in the information theory (IT) community [38, 103]. It was then extended by the networking research community to characterize a node's spatial freedom provided by its multiple antennas. Typically, the number of available DoFs at a node is assumed to be equal to the number of antennas at the node and represents the total available resources at the node for spatial multiplexing (SM) and interference cancellation (IC) [14, 22, 46, 82]. SM refers to the use of one or multiple DoFs for data stream transmission/reception (at both transmit and receive nodes), with each data stream corresponding to one DoF. IC refers to the use of one or multiple DoFs to cancel interference, which can be done either at the transmit node or the receive

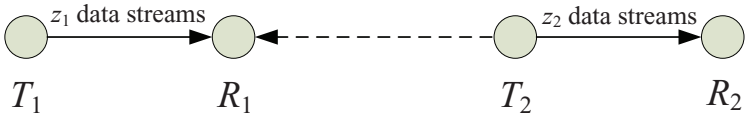


Figure 2.1: An example that illustrates SM and IC.

node. For example, consider the two links in Fig. 2.1. To transmit z_1 data streams on link (T_1, R_1), both nodes T_1 and R_1 need to consume z_1 DoFs for SM. Similarly, to transmit z_2 data streams on link (T_2, R_2), both nodes T_2 and R_2 need to consume z_2 DoFs for SM. The interference from T_2 to R_1 can be canceled by either R_1 or T_2. If R_1 cancels this interference, it needs to consume z_2 DoFs. If T_2 cancels this interference, it needs to consume z_1 DoFs.

A significant advantage of DoF model is that it only requires simple numeric computation (addition and subtraction) to keep track of SM and IC at a node. Although a DoF model is not able to completely capture all the PHY-layer capabilities of MIMO, it offers a simple yet effective tool to study MIMO in a multi-hop network. As such, various DoF models have been proposed and applied to solve a variety of network problems (see, e.g., [5, 17, 29, 44] for throughput maximization, [7, 63, 68, 86, 87] for MAC protocols).

Since interference can be canceled by either its transmit node or its receive node (as shown in Fig. 2.1), a question to ask is which node should take the responsibility for IC? The lack of a systematic rule in assigning IC responsibility is likely to lead to either sub-optimal or infeasible solution and is the main limitation in the prior efforts. In [63], Mumey et al. proposed an approximation algorithm for joint stream control and scheduling, where IC is done only at the receiver. In [86], Sundaresan et al. studied MAC design in which IC can be done only at the receiver. Without exploiting IC at the transmit nodes, the DoF model in [63, 86] tends to shrink the feasible solution space unnecessarily. In [87], Sundaresan et al. proposed to allocate only one DoF at each node for SM while reserving the remaining DoFs for IC. Such a static approach cannot be optimal for maximizing throughput in MIMO networks. Park et al. [68] studied a MAC problem and proposed to impose the IC responsibility to a newly active link (without changing the current DoF behavior at other links). Again, such non-collaborative IC strategy is overly restrictive and is unlikely to lead

to an optimal solution. In [5, 29], the authors studied cross-layer design for throughput maximization problems. Although both efforts allowed either transmit or receive node to perform IC, there was no clear guideline on how this should be done in a systematic manner. As a result, it was shown in [78] that such an approach results in a small DoF region that is far from optimal. In [7], Blough et al. proposed a scheduling algorithm for throughput maximization based on a DoF model, which allowed IC to be done at either transmitter or receiver without any other restriction. However, due to the lack of mathematical proof, their DoF allocation cannot guarantee a feasible solution at the physical layer.

In [78], we explored the important problem of how to use DoF correctly (to ensure feasibility) and efficiently (to avoid duplication in IC). The main result in [78] is a new DoF model that performs IC among the nodes based on a *global node ordering*. Specifically, each transmit/receive node only consumes DoFs for canceling interference to/from those nodes before itself in the global node ordering, it does not need to consume DoFs for canceling interference to/from those nodes after itself in the global node ordering. It was shown in [78] that once DoF allocation is performed at each node following a global node ordering, potential duplication in IC can be completely eliminated. Furthermore, such a DoF allocation is guaranteed to be feasible at the physical layer. We will review more details of this global node ordering concept in Section 2.2.

Despite its performance superiority over previous MIMO DoF models, the new DoF model in [78] relies on a *global* node ordering to keep track of IC responsibilities among all the nodes. A natural question that one may raise about this model is whether such characteristics would have difficulty in a network where all operations are performed distributedly. In this chapter, we explore this question by applying this model to a multi-hop MIMO network. We are interested in whether carefully designed local node operations (and local node ordering) can be translated into the desired ordering and feasibility on the global level. Specifically, for a set of sessions in the network, we study how to schedule DoF resources among the nodes so

that the minimum data throughput among the sessions can be maximized.[1] We formulate this throughput maximization problem as a cross-layer optimization problem and develop an efficient and fast DoF scheduling algorithm to solve it. Our DoF scheduling algorithm is an iterative algorithm and includes three modules in each iteration: *link selection module* (LSM), *resource allocation module* (RAM), and *local re-adjustment module* (LRM). Some of the highlights of our algorithm include:

- It is amenable to local implementation. We show that each module in our algorithm can be implemented in a distributed manner.
- Upon algorithm termination, the final DoF allocation solution is feasible at global level. That is, there exists a global node ordering for the final solution corresponding to the DoF allocation at each node. This is not trivial, given that each module performs operations locally without global knowledge.
- Its performance is highly competitive. Simulation results show that the objective values obtained by our algorithm are close to upper bounds of the same problem (obtained by CPLEX solver). We therefore conclude that the objective by our algorithm is very close to the optimum.
- It has a polynomial-time complexity and thus offers a solution rather quickly (in contrast to exponential complexity of solving the MILP problem).

The remainder of this chapter is organized as follows. In Section 2.2, we give a review of the new DoF model developed in [78]. In Section 2.3, we formulate a throughput maximization problem based on the new DoF model. Section 2.5 presents our DoF scheduling algorithm in detail and Section 2.7 analyzes the algorithm. In Section 2.8, we demonstrate the performance of our DoF scheduling algorithm through a simulation study. Section 2.9 concludes this chapter.

[1]Note that our DoF scheduling problem differs from those efforts on distributed MIMO scheduling (e.g., [52, 71]) as the latter was not based on a DoF link model.

2.2 Background: A Node Order-based MIMO DoF Model

We consider a multi-hop network consisting of a set of nodes, each of which is equipped with multiple antennas. Assume that the channel matrix between any two nodes has full rank. Then the number of DoFs available to a node is equal to the number of its antennas. A node can use some or all of its DoFs for either SM or IC, as long as the number of consumed DoFs does not exceed the total available DoFs at the node. For the MIMO DoF model in [78], DoFs (for SM and IC) at a node is allocated based on the following guideline.

For SM, both the transmit and receive nodes consume DoFs. The number of DoFs consumed at both the transmit and receive nodes is equal to the number of data streams that is transported between the two nodes. For IC, unlike SM, only a transmit node or a receive node needs to consume DoFs, not both. The questions of which node should be responsible for IC and how many DoFs are needed are effectively addressed by the node ordering concept described in [78]. Specifically, all nodes in the network are put into an ordered list. The position of a node in the list represents its order in the node list. A node consumes DoFs for IC as follows:

- *Transmit node.* If the node is a transmit node, then it only needs to cancel its interference to those receive nodes (within its interference range) that are before itself in the ordered node list. It does not need to consume DoFs to cancel its interference to those receive nodes that are after itself in the ordered node list. Interference from this transmit node to those receive nodes after itself will be canceled by those receive nodes latter. For IC, the number of DoFs consumed at this transmit node is equal to the total number of data streams received by those receive nodes from their intended transmitters.
- *Receive node.* If the node is a receive node, then it only needs to cancel interference from those transmit nodes (whose interference ranges cover this receive node) that are before itself in the ordered node list. It does not need to cancel interference from those

transmit nodes that are after itself in the ordered node list. Interference from those transmit nodes after this node will be canceled by those transmit nodes latter. For IC, the number of DoFs consumed at this receive node is equal to the total number of data streams transmitted by those transmit nodes.

As shown in [78], by referencing an ordered node list, one can avoid duplication in IC between transmit and receive nodes while ensuring the feasibility of the final DoF scheduling solution. Furthermore, an optimal ordering of a node list can be obtained by putting the ordering constraint into a problem formulation. In the rest of this section, we give a model for the ordering-based DoF allocation in a time-slotted system. Table 2.1 lists the notation in this chapter.

Assume that a time frame in data plane (time resource for data transmission and reception) consists of T equal-length time slots. Suppose that there are N nodes in the network and node i has A_i antennas. Denote a binary variable $x_i(t)$ as an indicator of whether node i is a transmitter in time slot t. Similarly, denote $y_i(t)$ as an indicator of whether node i is a receiver in time slot t. Let $\mathcal{L}_i^{\text{in}}$ and $\mathcal{L}_i^{\text{out}}$ be the set of possible incoming and outgoing links at node i (determined by the transmission range of a node), respectively. Denote $z_l(t)$ as the number of data streams on link l in time slot t. Then we have

$$x_i(t) \leq \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) \leq A_i \cdot x_i(t), \qquad (1 \leq i \leq N, 1 \leq t \leq T), \tag{2.1}$$

$$y_i(t) \leq \sum_{l \in \mathcal{L}_i^{\text{in}}} z_l(t) \leq A_i \cdot y_i(t), \qquad (1 \leq i \leq N, 1 \leq t \leq T). \tag{2.2}$$

Denote $\pi(t)$ as the order of nodes in the network in time slot t and denote $\pi_i(t)$ as the position of node i in order $\pi(t)$. Then we have

$$1 \leq \pi_i(t) \leq N, \qquad (1 \leq i \leq N, 1 \leq t \leq T). \tag{2.3}$$

Denote binary variable $\theta_{ji}(t)$ as the relative position of nodes j and i in order $\pi(t)$ as follows: $\theta_{ji}(t) = 1$ if node j is before node i in order $\pi(t)$ and 0 otherwise. Then we have

$$\pi_i(t) - N \cdot \theta_{ji}(t) + 1 \leq \pi_j(t) \leq \pi_i(t) - N \cdot \theta_{ji}(t) + N - 1, \qquad (1 \leq i \leq N, j \in \mathcal{I}_i, 1 \leq t \leq T), \tag{2.4}$$

Table 2.1: Notation for the design of DoF scheduling algorithm.

Network setting	
N	The number of nodes in the network
$\mathcal{L}$	The set of links in the network
L	The number of links in the network
$\mathcal{F}$	The set of sessions in the network
F	The number of sessions in the network
A_i	The number of antennas at node i
A	The maximum antenna number among all nodes
T	The number of time slots in a frame in data plane
Problem formulation	
$x_i(t)$	A binary variable to indicate whether node i is a transmitter for some link in time slot t
$y_i(t)$	A binary variable to indicate whether node i is a receiver for some link in time slot t
$z_l(t)$	The number of data streams on link l in time slot t
$\mathcal{L}_i^{in}$	The set of incoming links at node i
$\mathcal{L}_i^{out}$	The set of outgoing links at node i
$\pi(t)$	An ordering of all nodes in time slot t
$\pi_i(t)$	The position of node i in ordered list $\pi(t)$
$\theta_{ji}(t)$	A binary variable to indicate whether node i is placed after node j in $\pi(t)$
$r_l(f)$	The amount of rate on link l attributed to session f
$r(f)$	The data rate of session f
$r_{\min}$	The minimum data rate among all sessions

Table 2.1: Continued.

Algorithm design	
$\mathcal{B}$	A sorted link list for link selection
$\hat{\pi}(t)$	A node ordering in time slot t
$q_{(i,j)}$	The interference burden of link (i,j)
$\lambda_i(t)$	The number of DoFs that node i consumes for SM
	and IC in time slot t
$\overline{\lambda}_i(t)$	The number of remaining DoFs at node i in time slot t, $\overline{\lambda}_i(t) = A_i - \lambda_i(t)$
$\varphi(t)$	A DoF allocation for SM and IC in time slot t

where $\mathcal{I}_i$ is the set of nodes within node i's interference range.

For each node i in the ordered node list, we can mathematically model its DoF consumption for SM and IC as follows:

$$\text{If } x_i(t)=1, \text{ then } \sum_{l\in\mathcal{L}_i^{\text{out}}} z_l(t) + \sum_{j\in\mathcal{I}_i} \theta_{ji}(t) \sum_{k\in\mathcal{L}_j^{\text{in}}}^{\text{Tx}(k)\neq i} z_k(t) \leq A_i, \qquad (1\leq i\leq N, 1\leq t\leq T), \tag{2.5}$$

where on the left-hand side of the inequality, the first and second terms represent the number of DoFs consumed by node i for SM and IC, respectively. Similarly,

$$\text{If } y_i(t)=1, \text{ then } \sum_{l\in\mathcal{L}_i^{\text{in}}} z_l(t) + \sum_{j\in\mathcal{I}_i} \theta_{ji}(t) \sum_{k\in\mathcal{L}_j^{\text{out}}}^{\text{Rx}(k)\neq i} z_k(t) \leq A_i, \qquad (1\leq i\leq N, 1\leq t\leq T). \tag{2.6}$$

For constraint (2.5), if $x_i(t)=1$, then we have $\sum_{l\in\mathcal{L}_i^{\text{out}}} z_l(t) + \sum_{j\in\mathcal{I}_i} \theta_{ji}(t) \sum_{k\in\mathcal{L}_j^{\text{in}}}^{\text{Tx}(k)\neq i} z_k(t) \leq A_i$. On the other hand, if $x_i(t)=0$, then no DoF is consumed. Constraint (2.5) can be reformulated by incorporating binary variable $x_i(t)$ into the expression as follows:

$$\sum_{l\in\mathcal{L}_i^{\text{out}}} z_l(t) + \sum_{j\in\mathcal{I}_i} \theta_{ji}(t) \sum_{k\in\mathcal{L}_j^{\text{in}}}^{\text{Tx}(k)\neq i} z_k(t) \leq A_i x_i(t) + (1-x_i(t))B, \qquad (1\leq i\leq N, 1\leq t\leq T), \tag{2.7}$$

where $B=\sum_{i=1}^{N} A_i$ is an upper bound of the second term on the left-hand side of (2.7).

Similarly, constraint (2.6) can be reformulated as follows:

$$\sum_{l\in\mathcal{L}_i^{\text{in}}} z_l(t) + \sum_{j\in\mathcal{I}_i} \theta_{ji}(t) \sum_{k\in\mathcal{L}_j^{\text{out}}}^{\text{Rx}(k)\neq i} z_k(t) \leq A_i y_i(t) + (1-y_i(t))B, \qquad (1\leq i\leq N, 1\leq t\leq T). \tag{2.8}$$

2.3 From DoF Link Model to Network-level Throughput Maximization

The above DoF link model offers an excellent tool to study networking problems for MIMO networks. Consider a MIMO network consisting of a set of N nodes, where node i, $i = 1, 2, \cdots, N$, is equipped with A_i antennas. Suppose that there is a set of F unicast sessions in the network, with their source and destination nodes being randomly selected among all the nodes. The route of each session can be computed by some routing protocol (e.g., AODV and OLSR). We assume that scheduling is done in a time frame consisting of T time slots. In such a network, our goal is to find an optimal scheduling solution in each time slot so that the minimum achievable (end-to-end) throughput among all the sessions can be maximized. Denote $r(f)$ as the achievable end-to-end throughput of session f. Then our objective can be mathematically written as follows: maximize $\min_{1 \le f \le F}\{r(f)\}$.

2.3.1 Formulation

Half Duplex. We assume that a node's transceiver is half-duplex. Then we have

$$x_i(t) + y_i(t) \le 1, \qquad (1 \le i \le N; 1 \le t \le T). \tag{2.9}$$

Link Capacity Constraint. Denote src(f) and dst(f) as the source and destination nodes of session f, respectively. Denote $r_l(f)$ as the amount of data rate on link l that is attributed to session f. For simplicity, we assume that fixed modulation and coding scheme (MCS) is used for each data stream and that each data stream corresponds to one unit data rate. Then the average rate of link l over T time slots is $\frac{1}{T}\sum_t z_l(t)$. Thus, we have

$$\sum_{f=1}^{F} r_l(f) \le \frac{1}{T}\sum_{t=1}^{T} z_l(t), \qquad (1 \le l \le L), \tag{2.10}$$

where L is the number of links in the network.

Flow Balance at Each Node. At each node, flow conservation must be observed. Then

OPT-DoF-Raw:

max $r_{\min}$

s.t. MIMO DoF constraints: (2.1)–(2.4) and (2.7)–(2.8);

half-duplex constraints: (2.9);

link capacity constraints: (2.10);

flow balance constraints: (2.11) and (2.12);

throughput objective: (2.14).

Figure 2.2: A formulation for the DoF scheduling problem.

at a source node, we have

$$\sum_{l \in \mathcal{L}_i^{\text{out}}} r_l(f) = r(f), \qquad (i = \text{src}(f), 1 \le f \le F). \tag{2.11}$$

At an intermediate node, we have

$$\sum_{l \in \mathcal{L}_i^{\text{out}}} r_l(f) = \sum_{l \in \mathcal{L}_i^{\text{in}}} r_l(f), \qquad (1 \le i \le N, 1 \le f \le F, i \ne \text{src}(f), i \ne \text{dst}(f)). \tag{2.12}$$

At a destination node, we have

$$\sum_{l \in \mathcal{L}_i^{\text{in}}} r_l(f) = r(f), \qquad (i = \text{dst}(f), 1 \le f \le F). \tag{2.13}$$

It can be easily verified that if (2.11) and (2.12) are satisfied, then (2.13) is also satisfied. Therefore, it is sufficient to include only (2.11) and (2.12) in the problem formulation.

Throughput Objective. Denote $r_{\min}$ as the throughput rate of the bottleneck session. Then we have

$$r(f) \ge r_{\min}, \qquad (1 \le f \le F). \tag{2.14}$$

Based on the above constraints and the MIMO DoF model described in Section 2.2, our throughput optimization problem can be formulated in Fig. 2.2.

2.3.2 Reformulation

The formulation in Fig. 2.2 is in the form of mixed integer nonlinear program (MINLP). It is possible to reformulate the nonlinear constraints into linear ones. The nonlinear constraints in Fig. 2.2 are (2.7) and (2.8). To linearize them, we employ the Reformulation-Linearization Technique (RLT) [76], which replaces nonlinear terms by introducing new variables and new linear constraints. We define $\lambda_{ji}(t) = \theta_{ji}(t) \sum_{k \in \mathcal{L}_j^{\text{in}}}^{\text{Tx}(k) \neq i} z_k(t)$. Then we can replace the nonlinear constraint (2.7) by the following linear constraints:

$$\sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) + \sum_{j \in \mathcal{I}_i} \lambda_{ji}(t) \leq A_i x_i(t) + (1 - x_i(t))B, \qquad (1 \leq i \leq N, 1 \leq t \leq T), \tag{2.15}$$

$$0 \leq \lambda_{ji}(t) \leq A_j \cdot \theta_{ji}(t), \qquad (1 \leq i \leq N, j \in \mathcal{I}_i, 1 \leq t \leq T), \tag{2.16}$$

$$A_j \cdot \theta_{ji}(t) - A_j + \sum_{k \in \mathcal{L}_j^{\text{in}}}^{\text{Tx}(k) \neq i} z_k(t) \leq \lambda_{ji}(t) \leq \sum_{k \in \mathcal{L}_j^{\text{in}}}^{\text{Tx}(k) \neq i} z_k(t), \qquad (1 \leq i \leq N, j \in \mathcal{I}_i, 1 \leq t \leq T). \tag{2.17}$$

Similarly, we define $\mu_{ji}(t) = \theta_{ji}(t) \sum_{k \in \mathcal{L}_j^{\text{out}}}^{\text{Rx}(k) \neq i} z_k(t)$. Then we can replace the nonlinear constraint (2.8) by the following linear constraints:

$$\sum_{l \in \mathcal{L}_i^{\text{in}}} z_l(t) + \sum_{j \in \mathcal{I}_i} \mu_{ji}(t) \leq A_i y_i(t) + (1 - y_i(t))B, \qquad (1 \leq i \leq N, 1 \leq t \leq T), \tag{2.18}$$

$$0 \leq \mu_{ji}(t) \leq A_j \cdot \theta_{ji}(t), \qquad (1 \leq i \leq N, j \in \mathcal{I}_i, 1 \leq t \leq T), \tag{2.19}$$

$$A_j \cdot \theta_{ji}(t) - A_j + \sum_{k \in \mathcal{L}_j^{\text{out}}}^{\text{Rx}(k) \neq i} z_k(t) \leq \mu_{ji}(t) \leq \sum_{k \in \mathcal{L}_j^{\text{out}}}^{\text{Rx}(k) \neq i} z_k(t), \qquad (1 \leq i \leq N, j \in \mathcal{I}_i, 1 \leq t \leq T). \tag{2.20}$$

Based on the linearized constraints, the formulation in Fig. 2.2 can be reformulated in Fig. 2.3, which is in the form of MILP. It is not difficult to see that the problem formulation inevitably requires integer and binary variables. This indicates that the problem we are trying to solve is NP-hard in general [19, 74], although a formal proof is not given in this chapter.

OPT-DoF:

max $r_{\min}$

s.t. MIMO DoF constraints: (2.1)–(2.4) and (2.15)–(2.20);

half-duplex constraints: (2.9);

link capacity constraints: (2.10);

flow balance constraints: (2.11) and (2.12);

throughput objective: (2.14).

Figure 2.3: A reformulation for the DoF scheduling problem.

2.4 Problem Statement and Challenges

Goals. Although there already exist algorithms (e.g., Branch-and-Bound and sequential fixing [35]) and commercial optimization solvers (e.g., IBM CPLEX [104]) that can be used to solve the problem in Fig. 2.3, these algorithms/solvers are limited to the centralized environment. The goal of this chapter is to develop a distributed and efficient algorithm to solve the problem in Fig. 2.3. Meanwhile, we hope that the resulting DoF scheduling solution (from our developed algorithm) will be globally feasible in each time slot. By "globally feasible" we mean that there exist a set of precoding vectors at each transmitter and a set of decoding vectors at each receiver, so that the data streams on each link (in the DoF scheduling solution) can be transported free of interference using zero-forcing technique at the physical layer (see Section II in [78]). Instead of dealing with complex design of precoding and decoding vectors at the physical layer, the global node ordering concept in Section 2.2 allows us to ensure the global feasibility of the DoF scheduling solution through simply maintaining a global ordered node list. That is, in each time slot, if there exists a global ordered node list following which each node has enough DoFs for SM and IC, then one can always construct precoding and decoding vectors at the physical layer so that the data streams on each link can be transported free of interference.

Challenges. To make sure that the resulting final DoF scheduling solution is global feasible

at the physical layer, we must make sure there exists a global node ordering in each time slot in the network. As explained in Section 2.2, the relative ordering between two nodes directly determines DoF consumption responsibility at each node for IC. In a centralized environment, an optimal global node ordering can be found by putting the ordering constraints (2.3) and (2.4) into the problem formulation (see Fig. 2.3). However, in a distributed multi-hop network environment, each node can only exchange scheduling information with its neighboring nodes to establish and maintain some local relative ordering among neighboring nodes. It is not clear how such a distributed local ordering in each individual node can lead to a feasible global ordering among all the nodes in the network.

In our distributed algorithm, through proper design, we show that it is possible to have a per-node based local node ordering match to a global ordering of all nodes in the network, thereby achieving the *same* effect as that in a centralized environment. Specifically, we will show that the establishment of initial per-node based local node ordering and re-adjustment of neighboring node ordering during each iteration lead to a feasible global node ordering.

2.5 A Distributed Algorithm

In this section, we develop a distributed DoF allocation algorithm to solve the problem in Fig. 2.3. We first state our assumptions and then give an overview of the algorithm. Finally, we explain the key modules of the algorithm in detail.

2.5.1 Assumptions

We have the following assumptions in the design of our distributed algorithm.

- Network and Traffic: We assume that the network is static (with rare node mobility) over a long enough time. We also assume that the nodes in the network are synchronized at a resolution to the time slot level, which is not stringent [80]. We also assume that each session has a persistent and latency-tolerant traffic at its source. When the network topology or traffic pattern is changed, the scheduling algorithm should run

again so that the network resource can be reallocated accordingly.

- Channel State Information (CSI): A node is assumed to have CSI between itself and its neighboring nodes. The CSI can be obtained as follows. A node periodically broadcasts a public pilot sequence such that the CSI between them can be estimated. During the data communication, the estimated CSI can be used as CSIR (CSI at receiver-side) for SM and IC. Based on the reciprocity property of wireless channel, the estimated CSI can also be used as CSIT (CSI at transmitter-side) for SM and IC. More analysis on CSI acquisition in practical stationary networks can be found in [102, 94]. In [102], Zhang et al. proposed a channel estimation scheme in practical MIMO network to obtain CSI at the receiver. In [94], Xie et al. developed an adaptive feedback compression mechanism to send estimated CSI from the receiver to the transmitter with acceptable overhead.
- Control Channel: We assume there is a control channel for scheduling (e.g., IEEE 802.16j mobile multi-hop relay (MMR) networks [21]) and the control channel consists of a set of time slots.[2] In each time slot, a node may need to exchange scheduling information with the nodes within its interference range. A question to ask is if a node can communicate with those nodes within its interference range (beyond its transmission range) since the data communication is limited in the transmission range. In wireless network (e.g., cellular network), the transmit power of control plane is usually larger than that of data plane, allowing control plane to have a larger coverage than data plane. Given that the scheduling information exchange is within the control plane, it is practical for a node to exchange scheduling information with the nodes within its interference range.
- Chapter Scope: The goal of this chapter is to outline an efficient algorithm that can solve the problem in Fig. 2.3 in a distributed environment. The main contribution is to show the proposed algorithm can preserve global feasibility in DoF allocation even

[2]Note that the time slots in control channel/plane are different from the time slots in data channel/plane. The time slots in control channel are dedicated to the development of scheduling solution (e.g., scheduling information exchange), while the T time slots in data channel are devoted to data transmission and reception.

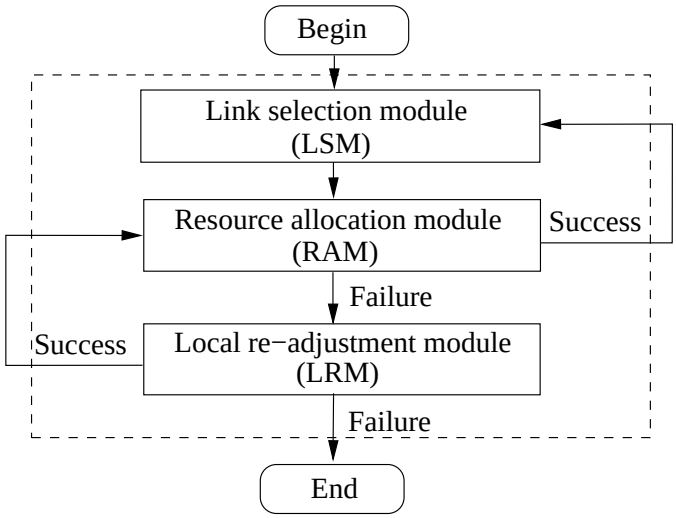


Figure 2.4: A flow chart of our proposed DoF scheduling algorithm.

through local operations. From protocol and implementation perspective, we recognize that there remain many details that need to be spelled out. Due to space limitation, we defer such details for future work and instead focus our efforts on algorithm design in this chapter.

2.5.2 Algorithm Overview

We offer an overview of the proposed DoF allocation algorithm to solve the optimization problem in Fig. 2.3. A flow chart of the algorithm is illustrated in Fig. 2.4, which includes three key modules: *link selection module* (LSM), *resource allocation module* (RAM), and *local re-adjustment module* (LRM). In essence, it is an iterative greedy algorithm that attempts to increase the minimum rate among all active links in each iteration. After a bottleneck link is identified, the RAM is invoked to see how the DoFs for SM can be increased in one of the T time slots while satisfying all local and neighboring interference constraints. If RAM is not able to yield a feasible increment, then we explore whether altering the local ordering of some nodes may yield a feasible increment, despite that a global node ordering information is not available to each individual node. This is done by LRM.

Here we give an overview of each module.

- **LSM.** The goal of this module is to identify a link for rate increment in an iteration. We propose a *session-independent* link selection approach by establishing a list of all links in the network based on their potential "interference burden". We show that this link selection approach is equivalent to the session-dependent link selection approach in terms of increasing the minimum rate among all sessions. We also show that this link selection approach can be implemented in a distributed environment.
- **RAM.** The goal of this module is to allocate DoF resource in one of the T time slots to increase the rate of the selected link. We first introduce local node ordering and global node ordering as well as the data structure that should be maintained at each node. Then, we explore the conditions under which the rate of the selected link can be increased in a given time slot and how DoFs should be allocated for the rate increment if the conditions are satisfied. We further show that if the DoF allocation *before* the rate increment is feasible based on a global node ordering, then the DoF allocation *after* the rate increment is also feasible and corresponds to a new global node ordering. Based on the outcome for the rate increment in a given time slot, we explain how to allocate DoF resource for the rate increment in a time frame.
- **LRM.** When RAM fails to increase the rate of the selected link, it is likely that the DoF allocation algorithm is stuck in a local optimal point. To allow the algorithm to jump out the local optimal point, we use LRM to alter some local node ordering so that some DoFs can be relieved from some nodes to accommodate one more data stream on the selected link. In a given time slot, we first identify the set $\mathcal{D}$ of nodes that are in shortage of DoF resource for the rate increment, and then explain how to adjust the local ordering for a node in $\mathcal{D}$ so that its remaining DoFs can be increased. We show that such local node ordering adjustment can preserve global feasibility of a solution and the existence of a global node ordering, albeit implicit. Based on the outcome of the local ordering adjustment in a given time slot, we explain how to perform the local ordering adjustment in a time frame.

In the rest of this section, we explain the three modules in more detail.

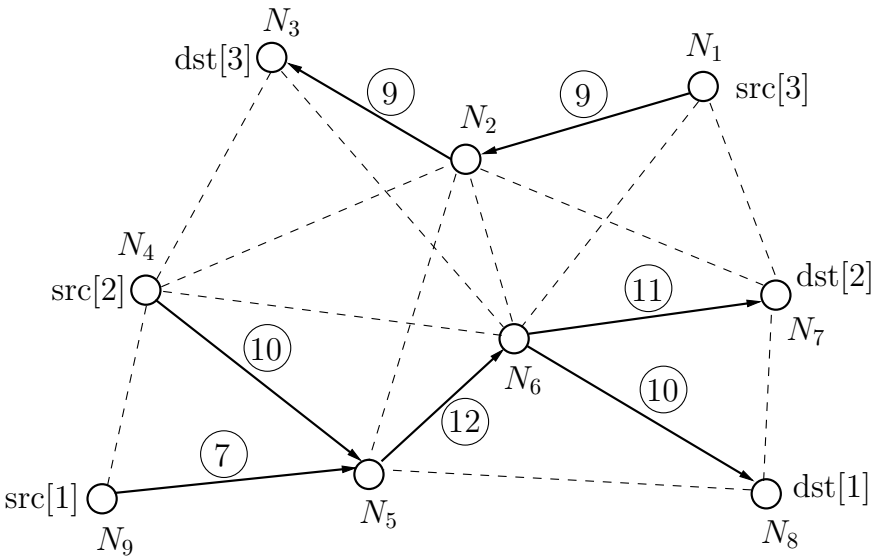


Figure 2.5: An example that illustrates link priorities.

Table 2.2: Each node's interference burden.

Node i	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
q_i	3	6	3	5	5	7	4	3	2

2.5.3 Link Selection Module

Several approaches may be considered to increase the minimum rate among all sessions iteratively. A straightforward approach is to identify a session with the minimum rate in the network and then try to increase the rate of each link by one unit along the session's path. Unfortunately, our simulation results reveal that an algorithm based on this approach does not perform well. The failure of such a *session-dependent* link selection approach may be attributed to the fact that it ignores the significance of potential "interference burden" of each link in the network. By "interference burden" of a link, we mean the number of DoFs required at both the link's transmitter and receiver for IC. This consideration motivates us to pursue a *session-independent* link selection approach based on a link's interference burden in the network.

More formally, for node i, we define its interference burden as q_i, which is the number

of nodes within node i's interference range. Then for link (i, j), we define is its interference burden $q_{(i,j)}$, or priority, as $q_{(i,j)} = q_i + q_j$. In our approach, we sort all active links in the network based on *non-increasing* order of their interference burden into a list, which we denote as $\mathcal{B}$. A small but important detail in $\mathcal{B}$ is the representation of a link that is traversed by multiple sessions. In our design, we would like to have session-independent link based approach to achieve the same effect as the session-dependent link based approach in term of increasing the minimum rate among all sessions iteratively. To do this, it is necessary to represent a link multiple times in list $\mathcal{B}$ if it is traversed by multiple sessions. The following example illustrates how to establish list $\mathcal{B}$.

Consider the network in Fig. 2.5. There are 3 active sessions in the network. Note that link (N_5, N_6) is traversed by sessions 1 and 2 and thus will be listed twice in $\mathcal{B}$. For each node i, Table 2.2 lists its interference burden q_i. Based on Table 2.2, it is easy to calculate the priority of each link $q_{(i,j)}$, which is shown next to each link in Fig. 2.5. For example, for link (N_1, N_2), $q_{(N_1,N_2)} = q_{N_1} + q_{N_2} = 9$. By sorting the links in non-increasing order by their priorities and representing link (N_5, N_6) twice, we have $\mathcal{B} = [(N_5, N_6), (N_5, N_6), (N_6, N_7), (N_4, N_5), (N_6, N_8), (N_1, N_2), (N_2, N_3), (N_9, N_5)]$. Note that the priority tie between (N_1, N_2) and (N_2, N_3) is broken by giving link (N_1, N_2) higher priority due to its smaller source node ID.

Based on this link list $\mathcal{B}$, we select link sequentially for rate increment (by one data stream). The reason why we consider links with higher priorities (i.e., larger interference burden) first is that resource allocation task for these links is likely to be more demanding than those links with lower priorities (i.e., smaller interference burden). Intuitively, once these most demanding links are taken care of first, it would be easier for us to perform resource allocation for those less demanding links with the remaining network resource.

Since list $\mathcal{B}$ is invisible to each link in a distributed network, the question to ask is how each link can obtain its rank in list $\mathcal{B}$. This problem can be solved by using the distributed ranking algorithm in [98]. To apply the distributed ranking algorithm to our problem, we can have the transmitter of each link maintain the priority of that link and then execute the

distributed ranking algorithm by treating the reciprocal priority of that link (i.e., $1/q_{(i,j)}$) as its initial value. At the end of the ranking algorithm, the transmitter of each link can obtain the rank of that link. Given that the ranking algorithm in [98] has two phases and the node operations in each phase do not require synchronization, it takes two time slots in control channel for the transmitter of each link to obtain its rank. In the worst case, the communication overhead of the ranking algorithm requires $N^2/2 + O(N)$ messages, which are acceptable in practical networks.

Once each active link obtains its rank in list $\mathcal{B}$, then one link in $\mathcal{B}$ will be selected in each time slot (in control channel) to schedule rate increment. Such link selection process is cyclic as time slots in control channel progress. As a result, for each link, it has precise knowledge of which time slots it will be chosen for rate increment operations.

2.5.4 Resource Allocation Module

The goal of the RAM is to allocate DoF resource for the selected link so that the rate of the selected link can be increased by one data stream in one of the T time slots on the data plane. To do this, we first discuss the relationship between local node ordering and global node ordering. Based on this understanding, we introduce the data structure that should be maintained at each node, and then explore the condition under which a link rate can be successfully increased by one data stream.

Local Node Ordering vs. Global Node Ordering. Recall that in Section 2.2, a global node ordering plays a key role in a feasible and efficient DoF scheduling [78]. However, in a distributed environment it is impractical to establish and maintain such a global node ordering in the network. Instead, we propose to have each node establish and maintain a relative ordering with its neighboring nodes in a distributed environment. In Section 2.6, we will show that the established and maintained local ordering at each individual node leads to a feasible global node ordering.

To establish a local ordering, we have each node i maintain two sets of its neighboring nodes: (i) $\mathcal{I}_i(t)$ the set of nodes for which node i has allocated DoFs for IC: these nodes are

Table 2.3: State information at each node i.

Symbol	Definition
$s_i(t)$	The status of node i (transmit, receive, or idle) in time slot t
$\mathcal{I}_i$	The set of nodes in node i's interference range
$\mathcal{I}_i^T(t)$	Transmitters in $\mathcal{I}_i$ in time slot t
$\mathcal{I}_i^R(t)$	Receivers in $\mathcal{I}_i$ in time slot t
$\mathcal{L}_i$	The set of incoming and outgoing links at node i
$\{z_l(t): l \in \mathcal{L}_i\}$	The number of data streams on the incoming or outgoing links of node i
$\lambda_i^{\text{SM}}(t)$	The number of DoFs at node i allocated for SM in time slot t
$\lambda_i^{\text{IC}}(t)$	The number of DoFs at node i allocated for IC in time slot t
$\mathcal{T}_i(t)$	The set of nodes to which node i has established links in time slot t
$\mathcal{I}_i(t)$	The set of nodes for which node i has allocated DoFs for IC in time slot t
$\mathcal{J}_i(t)$	The set of node i's neighboring nodes that have allocated their DoFs to cancel interference either to or from nodes i in time slot t

considered *before* node i in the local ordering; and (ii) $\mathcal{J}_i(t)$ the set of nodes that have allocated their DoFs to cancel interference either from or to node i: these nodes are considered *after* node i in the local ordering. We will show how these two sets can be established and maintained through a distributed mechanism. More importantly, we will show that by properly updating and maintaining these two sets at each node, one can determine which node is responsible for the cancellation of a particular interference. Further, we show in Theorem 1 that based on these two sets, one can identify a corresponding global node ordering in the network, although none of the nodes has such an explicit knowledge.

Data Structure at a Node. Table 2.3 lists state information that we maintain at each node in the network. In the table, $\mathcal{I}_i(t)$ and $\mathcal{J}_i(t)$ are the two sets representing the local ordering at node i. Since $\mathcal{I}_i(t)$ contains the set of nodes for which node i has allocated DoFs for IC in time slot t, we consider that these nodes are *before* node i in the local ordering. Similarly, since $\mathcal{J}_i(t)$ contains the set of node i's neighboring nodes that have allocated their

DoFs to cancel interference either to or from nodes i in time slot t, these nodes are *after* node i in the local ordering. $\lambda_i^{\text{SM}}(t)$ is the number of DoFs allocated for SM at node i in time slot t. $\lambda_i^{\text{IC}}(t)$ is the number of DoFs allocated for IC at node i in time slot t. For ease of explanation, denote $\overline{\lambda}_i(t)$ as the number of remaining DoFs at node i in time slot t. Thus, we have $\overline{\lambda}_i(t) = A_i - \lambda_i^{\text{SM}}(t) - \lambda_i^{\text{IC}}(t)$. $s_i(t)$ is the status of node i in time slot t, which is defined as follows: $s_i(t) =$ "T" if node i is a transmitter; $s_i(t) =$ "R" if node i is a receiver; and $s_i(t) =$ "I" if node i is idle.

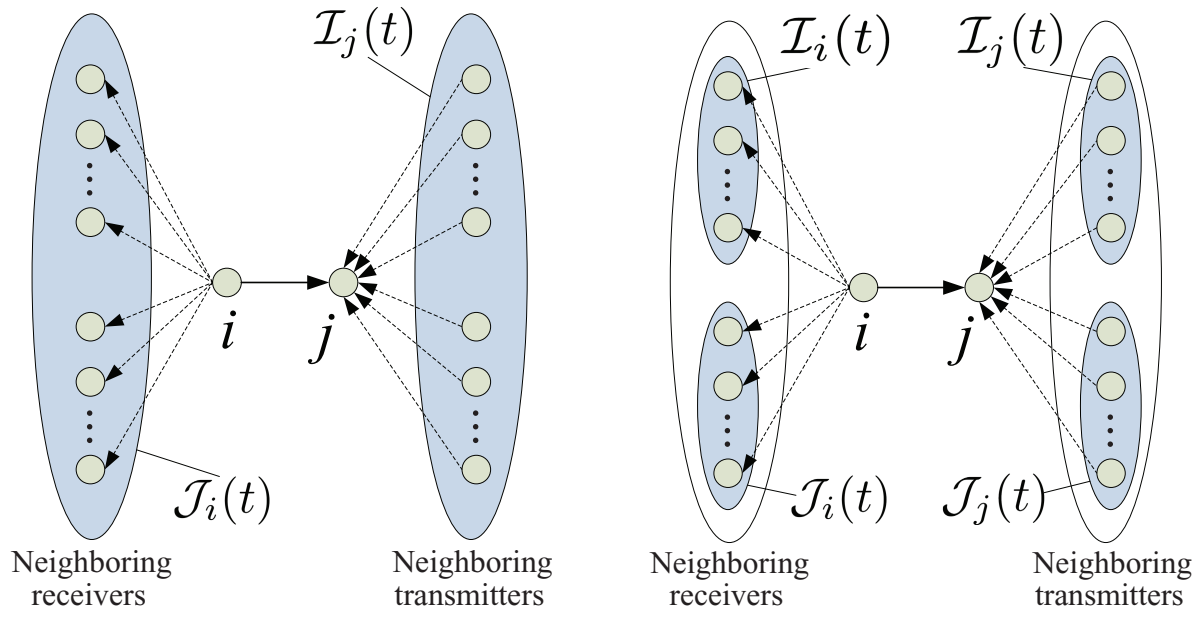
During the initialization stage, each node is set to idle status, i.e., $s_i(t) =$ "I" for $1 \leq i \leq N$, $1 \leq t \leq T$; each node allocates zero DoF for SM and IC, i.e., $\lambda_i^{\text{SM}}(t) = 0$ and $\lambda_i^{\text{IC}}(t) = 0$, $z_l(t) = 0$, $\mathcal{T}_i(t) = \emptyset$, $\mathcal{I}_i(t) = \emptyset$, $\mathcal{J}_i(t) = \emptyset$ for $1 \leq i \leq N$, $1 \leq l \leq L$, $1 \leq t \leq T$.

Rate Increment in Time Slot t. To increase the rate of link (i, j) by one data stream in time slot t, nodes i and j first check their current status in time slot t. If the status for both nodes meet the requirements (in Case I or Case II below), then nodes i and j as well as their neighboring nodes will check whether they have enough remaining DoFs for IC under their current local orderings. If yes, nodes i and j and relevant neighboring nodes update their state information to accommodate this increment on link (i, j).

Case I: Referring to Fig. 2.6(a), link (i, j) is not active and we wish to add one data stream on this link if the following conditions are satisfied: (i) node i is idle; (ii) all of the receivers within node i's interference range have at least one remaining DoF to cancel interference from node i; (iii) node j is idle and its total DoFs are more than the sum of DoFs for SM at those nodes that are interfering node j (assuming node j will become the last node in the local ordering).

If the above conditions are satisfied, then nodes i and j as well as their neighboring nodes do the following:

- At node i, its status is changed from idle to transmit. The number of DoFs consumed for SM at node i is updated to one. The rate of link (i, j) is increased to one. To update the local ordering at node i, we define node i to be the first node in its local ordering. To do this, we update $\mathcal{J}_i(t) = \mathcal{I}_i^R(t)$.



(a) Case I: link (i, j) is inactive

(b) Case II: link (i, j) is active

Figure 2.6: An example that illustrates RAM.

- At each of node i's neighboring nodes $a \in \mathcal{J}_i(t)$, node a adds node i into set $\mathcal{I}_a(t)$ and increases its DoF consumption (for IC) by one.
- At node j, its status is updated from idle to receive. The number of DoFs consumed for SM is updated to one. The rate of link (i, j) is updated to one. Correspondingly, to update the local ordering at node j, we define node j to be the last node in its local ordering. To do this, we update $\mathcal{I}_j(t) = \mathcal{I}_j^T(t)$. The number of DoFs consumed for IC is updated to be the sum of data streams of its neighboring transmitters except node i.
- At each of node j's neighboring nodes $b \in \mathcal{I}_j(t)$, node b adds node j into its set $\mathcal{J}_b(t)$.

Case II: Referring to Fig. 2.6(b), link (i, j) is already active and we wish to add one more data stream on this link if the following conditions are satisfied: (i) node i is a transmitter and has at least one DoF remaining for SM; (ii) each node in $\mathcal{J}_i(t)$ has at least one DoF remaining to cancel interference from node i; (iii) node j is a receiver and has at least one DoF remaining for SM; (iv) each node in $\mathcal{J}_j(t)$ has at least one DoF remaining to cancel its

interference to node j.

If the above conditions are satisfied, then nodes i and j as well as their neighboring nodes do the following:

- At node i, the number of DoFs consumed for SM is increased by one. The rate of link (i, j) is increased by one.
- Each node in $\mathcal{J}_i(t)$ increases its DoF consumption for IC by one.
- At node j, the number of DoFs consumed for SM is increased by one. The rate of link (i, j) is increased by one.
- Each node in $\mathcal{J}_j(t)$ increases its DoF consumption for IC by one.

It is easy to see that rate increment (as described in Cases I and II) is a local operation and also feasible (in terms of DoF allocation) for those nodes involved in this operation. A natural question to ask is how such local operation will affect global feasibility among all nodes. We now state an important property, which says that if the DoF allocation is feasible among all the nodes in the network, then this local operation will result in a new DoF allocation that is also globally feasible.

Formally, denote $\pi(t)$ as a global ordering for all nodes in the network. Based on $\pi(t)$, suppose that $\varphi(t)$ is a feasible DoF scheduling for SM and IC at all nodes in the network. Denote $\hat{\varphi}(t)$ as the new DoF scheduling after the rate increment operation on $\varphi(t)$. Then we have the following lemma:

Lemma 1. *$\hat{\varphi}(t)$ is a globally feasible DoF scheduling. Further, there exists a global ordering $\hat{\pi}(t)$ that corresponds to $\hat{\varphi}(t)$.*

Proof. We show that the lemma holds for two cases.

For Case I, we construct the global ordering $\hat{\pi}(t)$ by letting $\hat{\pi}(t) = [i\ \pi(t)\ j]$. Then we check each node's DoF consumption (for SM and IC) in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$ as follows: (i) Node i only needs to consume one DoF for SM and does not need to consume DoF for IC. Thus node i has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (ii) For each node in $\mathcal{I}_i^R(t)$, since it has at least one remaining DoF, it can perform SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (iii) Given $\sum_{h\in\mathcal{I}_j^T(t)} \lambda_h^{\mathrm{SM}}(t) < A_j$, node j has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based

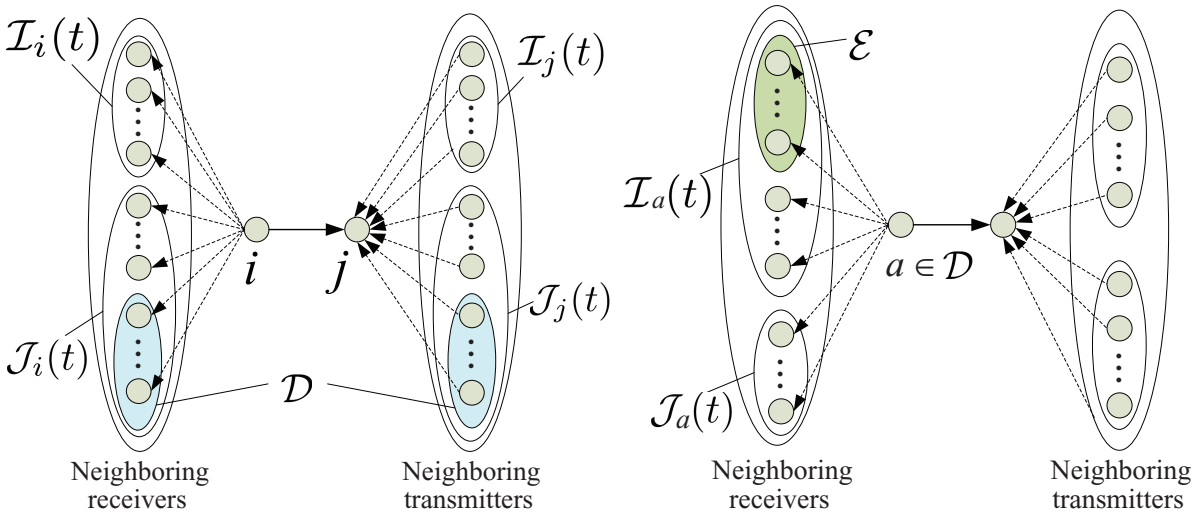
on $\hat{\pi}(t)$. (iv) For all other nodes in $\hat{\pi}(t)$, since they have enough DoFs for SM and IC in $\varphi(t)$ based on $\pi(t)$, they also have enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. Therefore, $\hat{\varphi}(t)$ is a globally feasible DoF scheduling and $\hat{\pi}(t)$ is a global ordering that corresponds to $\hat{\varphi}(t)$.

For Case II, we construct the global ordering $\hat{\pi}(t)$ by letting $\hat{\pi}(t) = \pi(t)$. Now we check each node's DoF consumption (for SM and IC) in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$: (i) Node i has at least one remaining DoF. Since node i only needs one more DoF for SM, it has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (ii) Every node in $\mathcal{J}_i(t)$ has at least one remaining DoF. Since each of these nodes only needs one more DoF for IC, all nodes in $\mathcal{J}_i(t)$ have enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (iii) Node j has at least one remaining DoF. Since node j only needs one more DoF for SM, it has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (iv) Every node in $\mathcal{J}_j(t)$ has at least one remaining DoF. Since each of these nodes only needs one more DoF for IC, all these nodes in $\mathcal{J}_j(t)$ have enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. (v) For all other nodes in $\hat{\pi}(t)$, since they have enough DoFs for SM and IC in $\varphi(t)$ based on $\pi(t)$, they have enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. Therefore, $\hat{\varphi}(t)$ is a globally feasible DoF scheduling and $\hat{\pi}(t)$ is a global ordering that corresponds to $\hat{\varphi}(t)$. □

Resource Allocation in a Time Frame. Recall that there are T time slots in a time frame. If the rate increment operation described above fails in the first time slot, we try it again in the second time slot and so forth, until it is successful in a time slot or fails after all T time slots. It is not difficult to see that this module is amenable to local implementation as all operations of this module are restricted on the selected link and its neighboring links.

2.5.5 Local Re-adjustment Module

When RAM fails to increase one data stream on the chosen link, it is likely that the algorithm is stuck in a local optimal point. Following the flow chart in Fig. 2.4, we invoke the *local re-adjustment module* (LRM) to allow the algorithm to jump out the local optimal point. The goal of this module is to adjust the local ordering for the nodes associated with the



(a) Set $\mathcal{D}$ for link (i, j) (b) Set $\mathcal{E}$ for node $a \in \mathcal{D}$

Figure 2.7: An example that illustrates LRM.

chosen link so that IC responsibilities can be transferred from one node to another, thereby relieving some DoF resources for some nodes so as to accommodate a new data stream on the chosen link.

Local Ordering Adjustment in Time Slot t. Given that RAM fails to increase a data stream on a given link (i, j), we conclude that there is a lack of DoF resources at a subset of nodes among i, j, or their neighboring nodes based on the current local ordering at these nodes. This subset of nodes can be easily identified in a hypothesized scenario by looking for those nodes that would use more DoFs than their total DoFs if one more data stream were added on link (i, j). Denote $\mathcal{D}$ as this subset of nodes in shortage of DoF resources, as shown in Fig. 2.7(a).

For each node $a \in \mathcal{D}$, we perform local ordering adjustment, with the goal of relieving one DoF (already used for IC) from a node so that a new DoF can become available. To avoid race condition in a distributed system, we use a token and let it pass from one node to the next in $\mathcal{D}$ so that at any time, only one node is allowed to perform local ordering adjustment. The initiation of this taken can be done by node i and then passed on to node j. The token is passed to the next node in $\mathcal{D}$ only if the local ordering adjustment in the

previous node in $\mathcal{D}$ is successful (resulting in one free DoF at that node). Otherwise, the token will not be passed to the next node and we move on to the next time slot in a frame.

Referring to Fig. 2.7(b), for a given node $a \in \mathcal{D}$ that currently holds the token, we first need to identify a set of a's neighboring nodes $\mathcal{E}$ that can relieve some of a's DoF consumption for IC. First, nodes in $\mathcal{E}$ should not include any node of i, j, and their neighboring nodes. Otherwise, we may run into a loop of changing local node ordering without yielding any net improvement. Second, nodes in $\mathcal{E}$ must be ahead of a in a's local ordering, i.e., $b \in \mathcal{I}_a(t)$, since a is using its DoFs to cancel interference from nodes in $\mathcal{E}$. Third, nodes in $\mathcal{E}$ should have enough remaining DoF resources to relieve a's DoF consumption for IC to b, i.e., $\overline{\lambda}_b(t) \geq \lambda_a^{\text{SM}}(t)$. Finally, we need to ensure that there does not exist another node, say c, that is in a higher local order than node $b \in \mathcal{E}$ but in a lower local order than a. This will ensure that a local node ordering swap between a and b will not violate the local ordering between b and c.

For the set of candidate nodes in $\mathcal{E}$ for node a, we only need one node to swap its local ordering with a. In our algorithm, we choose a node in $\mathcal{E}$ that has the most number of remaining DoFs. A tie can be broken by selecting the node with a smaller node ID. Denote this node in $\mathcal{E}$ as b^*. For nodes a and b^*, we perform the following operation: (i) node a moves b^* from its $\mathcal{I}_a(t)$ to $\mathcal{J}_a(t)$, indicating that node b^* is now behind node a in the new ordering; (ii) node a no longer needs to cancel interference from node b^* and its remaining DoFs are increased, i.e. $\lambda_a^{\text{IC}}(t) := \lambda_a^{\text{IC}}(t) - \lambda_{b^*}^{\text{SM}}(t)$, $\overline{\lambda}_a(t) := \overline{\lambda}_a(t) + \lambda_{b^*}^{\text{SM}}(t)$; (iii) node b^* moves a from its $\mathcal{J}_{b^*}(t)$ to $\mathcal{I}_{b^*}(t)$, indicating that node a is now before node b^* in the new ordering; (iv) node b^* now needs to cancel interference from node a and its remaining DoFs are decreased, i.e., $\lambda_{b^*}^{\text{IC}}(t) := \lambda_{b^*}^{\text{IC}}(t) + \lambda_a^{\text{SM}}(t)$, $\overline{\lambda}_{b^*}(t) := \overline{\lambda}_{b^*}(t) - \lambda_a^{\text{SM}}(t)$.

A question to ask is how such a local node reordering operation will affect global feasibility among all the nodes. We now state an important property, which says that if the DoF scheduling is feasible among all the nodes in the network, then the LRM operation will result in a new DoF scheduling that is also globally feasible. Formally, denote $\pi(t)$ as a global ordering for all the nodes in the network. Based on $\pi(t)$, suppose $\varphi(t)$ is a feasible

DoF scheduling for SM and IC for all nodes in the network. Denote $\hat{\varphi}(t)$ as the DoF scheduling for all the nodes after LRM is performed at some nodes a and b^* under $\varphi(t)$. Then we have the following lemma:

Lemma 2. *$\hat{\varphi}(t)$ is a globally feasible DoF scheduling. Further, there exists a global ordering $\hat{\pi}(t)$ that corresponds to $\hat{\varphi}(t)$.*

Proof. The proof is based on construction. Denote B as the set of nodes before b^* in $\pi(t)$, C as the set of nodes after a in $\pi(t)$, D as the set of nodes between a and b^* in $\pi(t)$. Then, we have $\pi(t) = [B\ b^*\ D\ a\ C]$. Denote Γ as the set of nodes that are in a lower ordering than node a. We construct the global ordering $\hat{\pi}(t)$ by letting $\hat{\pi}(t) = [B\ \ D \cap \Gamma\ \ a\ \ b^*\ \ D \cap \Gamma^c\ \ C]$, where Γ^c is the complement set of Γ. Next, we check the DoF consumption for SM and IC at each node in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$.

- For each node $h \in B \cup C$: From $\pi(t)$ to $\hat{\pi}(t)$, the local ordering of node h does not change. Since node h has enough DoFs for SM and IC in $\varphi(t)$ based on $\pi(t)$, it also has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$.
- For each node $h \in D \cap \Gamma$: From $\pi(t)$ to $\hat{\pi}(t)$, the nodes in $\{b^*\} \cup (D \cap \Gamma^c)$ are moved from the positions before node h to the positions after node h. Thus, the DoF consumption at node h in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$ is less than or equal to that in $\varphi(t)$ based on $\pi(t)$. Therefore, node h has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$.
- For node a: From $\pi(t)$ to $\hat{\pi}(t)$, the local ordering of node a does not change except that node b^* is moved from $\mathcal{I}_a(t)$ to $\mathcal{J}_a(t)$. Thus, node a does not need to cancel the interference from/to node b^* in $\hat{\pi}(t)$, indicating that the DoF consumption of node a in $\hat{\varphi}(t)$ is less than that in $\varphi(t)$. Therefore, node a has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. Further, the number of remaining DoFs at node a is increased by $\lambda_{b^*}^{\mathrm{SM}}(t)$.
- For node b^*: Since there does not exist any node in a higher local order than node b^* and in a lower local order than node a, we know that from $\pi(t)$ to $\hat{\pi}(t)$, the local ordering of node b^* does not change except that node a is moved from $\mathcal{J}_{b^*}(t)$ to $\mathcal{I}_{b^*}(t)$. Thus, node b^* needs to cancel the interference from/to node a in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$. Since

$\overline{\lambda}_{b^*}(t) \geq \lambda_a^{\text{SM}}(t)$, there are enough remaining DoFs at node b^* to cancel the interference from/to node a. Therefore, node b^* has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$.

- For each node $h \in D \cap \Gamma^c$: From $\pi(t)$ to $\hat{\pi}(t)$, the local ordering of node h does not change, indicating that the DoF consumption of node h in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$ is the same as that in $\varphi(t)$ based on $\pi(t)$. Therefore, node h also has enough DoFs for SM and IC in $\hat{\varphi}(t)$ based on $\hat{\pi}(t)$.

Since every node in $\hat{\varphi}(t)$ has enough DoFs for SM and IC based on $\hat{\pi}(t)$, we conclude that $\hat{\varphi}(t)$ is a globally feasible DoF scheduling and $\hat{\pi}(t)$ is a global ordering that corresponds to $\hat{\varphi}(t)$. □

Local Ordering Adjustment in a Time Frame. Recall that there are T time slots in a time frame. If the local ordering adjustment described above fails in the first time slot, we try again in the second time slot and so forth, until local ordering adjustment is successful or fails after all T time slots. It is easy to see that this module is amenable to local implementation as all operations of this module are restricted on the selected link and its neighboring links.

2.6 Proving Global Feasibility of Final Solution

Recall that both RAM and LRM in our algorithm perform local operations and are amenable to distributed implementation. A natural question to ask is whether the final DoF scheduling at all nodes in the network (upon the algorithm termination) is still feasible at a global level. The following theorem answers this question.

Theorem 1. *Suppose that $\varphi(t)$ is the final DoF scheduling for SM and IC at all nodes in the network. Then, $\varphi(t)$ is a globally feasible solution. Further, there exists a global ordering $\pi(t)$ that corresponds to $\varphi(t)$.*

Proof. We prove it by induction. Denote $\varphi_n(t)$ as the DoF scheduling at all nodes in the network at the end of the n-th iteration. It is easy to see that the iterative algorithm in

Fig. 2.4 will terminate in a finite number of iterations.

Base case: For $n = 1$, we show that $\varphi_n(t)$ is a global feasible solution with a global ordering $\pi_n(t)$. To see this, note that before the first iteration, none of the DoFs at any node in the network is allocated. So the LSM selects the link with the highest priority, say link (i, j). We perform RAM for the selected link (i, j) and thus obtain DoF scheduling $\varphi_1(t)$. Since there are no other active nodes in the network, $\varphi_1(t)$ is also a global feasible solution. Further, there exists a trivial global node ordering $\pi_1(t) = [i\ j]$ that corresponds to $\varphi_1(t)$.

Inductive step: Suppose that $\varphi_n(t)$ is a global feasible solution with a global ordering $\pi_n(t)$. We show that at the end of the $(n + 1)$-th iteration, $\varphi_{n+1}(t)$ is also a global feasible solution with a global ordering $\pi_{n+1}(t)$. From $\varphi_n(t)$ to $\varphi_{n+1}(t)$, the operations may include RAM only or both LRM and RAM (see Fig. 2.4). From Lemma 2, we know that LRM will preserve global feasibility of a solution as well as the existence of a global node ordering. From Lemma 1, we know that RAM will also preserve the global feasibility of a solution as well as the existence of a global node ordering. Therefore, if $\varphi_n(t)$ is a feasible solution with a global ordering $\pi_n(t)$, then $\varphi_{n+1}(t)$ is a feasible solution with a global ordering $\pi_{n+1}(t)$.

Combining the base case and the inductive step, we have that the final DoF scheduling $\varphi(t)$ is a globally feasible solution. Further, there exists a global ordering $\pi(t)$ that corresponds to $\varphi(t)$. □

2.7 Algorithm Analysis

2.7.1 Piecing up Global Node Ordering

In Section 2.6, we showed that the final DoF scheduling $\varphi(t)$ is a globally feasible solution and there exists a global ordering $\pi(t)$ that corresponds to $\varphi(t)$. Since the global ordering $\pi(t)$ is invisible to each individual node in the network, one may wonder how to piece up the global ordering $\pi(t)$ based on all the local node orderings in the network.

Denote $\mathcal{S}$ as the set of active nodes in time slot t in the final DoF scheduling solution. To obtain the global ordering $\pi(t)$, we first initialize $\pi(t)$ to an empty node list and then

Finding a global node ordering in time slot t

1. $\pi(t) = [\,]$; $\mathcal{S} = \{i : s_i(t) \neq$ "I", $1 \leq i \leq N\}$;
2. While $\mathcal{S} \neq \emptyset$ {
3. For each i in $\mathcal{S}$ {
4. If $\mathcal{I}_i(t) == \emptyset$ {
5. $\pi(t) := [\pi(t)\ \ i]$;
6. Remove node i from $\mathcal{S}$;
7. Remove node i from $\mathcal{I}_j(t)$ for each $j \in \mathcal{S}$; } } }

Figure 2.8: A pseudo-code of finding a global node ordering.

iteratively insert a node from $\mathcal{S}$ to $\pi(t)$ as follows:

- *Step 1:* From $\mathcal{S}$, select a node i with $\mathcal{I}_i(t) = \emptyset$.
- *Step 2:* Add node i at the end of node list $\pi(t)$, i.e., $\pi(t) := [\pi(t)\ \ i]$.
- *Step 3:* Remove node i from $\mathcal{S}$ and remove node i from $\mathcal{I}_j(t)$ for each $j \in \mathcal{S}$.
- *Step 4:* If $\mathcal{S}$ is not empty, then go to Step 1.

A pseudo-code for finding a global node ordering $\pi(t)$ is given in Fig. 2.8 (in supplemental material). Note that there may exist multiple global node orderings that correspond to the final DoF scheduling in a given time slot.

2.7.2 Computational Complexity

We now show that the DoF scheduling algorithm in Fig. 2.4 is of polynomial time complexity. For LSM, its computation mainly consists of sorting the links along the path of sessions in the network. As we explained in Section 2.5.3, the LSM can be done in a distributed fashion by employing the distributed ranking algorithm in [98], which has $O(N^4)$ complexity.

For RAM, nodes i and j (transmitter and receiver of the selected link) need to check the feasibility of increasing one data stream in T time slots. In each time slot, the computation mainly consists of two parts: (i) nodes i and j as well as their neighboring nodes check their remaining DoFs, which has $O(N)$ complexity; (ii) nodes i and j as well as their neighboring

nodes update their state information, which has $O(N)$ complexity. Since these operations can be done in T time slots, the complexity of RAM is $O(NT)$. For LRM, nodes i and j as well as their neighboring nodes perform local ordering adjustment in T time slots. In each time slot, the computation mainly consists of three parts: (i) identifying the subset of nodes $\mathcal{D}$, which has complexity $O(N)$; (ii) identifying a set of nodes $\mathcal{E}$ for each node $a \in \mathcal{D}$, which has complexity $O(N^2)$; (iii) adjusting the local ordering for each node $a \in \mathcal{D}$, which has complexity $O(N^2)$. Since these operations can be done in T time slots, the complexity of LRM is $O(N^2T)$. Following the flow chart in Fig. 2.4, the computation of each iteration mainly consists of RAM and LRM. Therefore, the complexity of each iteration of the DoF scheduling algorithm is $O(N^2T)$.

For this DoF scheduling algorithm, the number of iterations is bounded by $O(N^2TA)$. Since the complexity of each iteration is $O(N^2T)$, the total computational complexity is $O(N^4T^2A)$.

2.7.3 Overhead Analysis

In a distributed environment, the success of this DoF scheduling algorithm relies on the message exchange in control channel between neighboring nodes. Since it is hard to precisely quantify overhead induced by this algorithm, we develop an upper bound for the total volume of message exchanges between the neighboring nodes in this algorithm. In what follows, we analyze the volume of message exchanges in each module (LSM, RAM, and LRM) in the worst case.

For LSM, message exchanges are required to sort the links in the network. As we explained in Section 2.5.3, LSM sorts the links by directly employing the distributed ranking algorithm in [98]. Based on the results in [98], LSM requires $O(N^2)$ message exchanges in the worst case. For RAM and LRM, they are iterative modules and there are at most $O(N^2TA)$ iterations. To characterize their total volume of message exchanges, we first analyze their volume of message exchanges in each iteration. For RAM, nodes i and j need to check T time slots and the volume of message exchanges in each time slot is $O(1)$. So RAM requires

$O(T)$ message exchanges in an iteration. For LRM, nodes i and j need to check T time slots and the volume of message exchanges in each time slot is $O(N)$. So LRM requires $O(NT)$ message exchanges in an iteration. Since there are at most $O(N^2TA)$ iterations, the total volume of message exchanges induced by RAM and LRM is $O(N^3T^2A)$. By adding the message exchanges induced by LSM, RAM, and LRM together, this DoF scheduling algorithm requires $O(N^3T^2A)$ message exchanges to achieve convergence in the worst case. In practice, we are expecting the algorithm to use much less message exchanges, since the convergence speed should be much faster than $O(N^2TA)$ due to the existence of interference.

2.8 Performance Evaluation

In this section, we present simulation results to demonstrate the performance of the proposed DoF scheduling algorithm. Ideally, the best performance benchmark would be an optimal solution to the OPT-DoF problem in Fig. 2.3. However, OPT-DoF formulation is an MILP and its optimal solution cannot always be obtained (even with a large amount of time) by a solver such as CPLEX [104]. But one thing we can do is to compare the *performance* of the proposed algorithm against an upper bound of OPT-DoF, which can be obtained by solving the problem formulation in Fig. 2.3 by CPLEX with a fixed termination time (e.g., 3 hours).[3] Note that the optimal solution value lies between the upper bound and the feasible solution value found by our algorithm. Therefore, if simulation results show that the objective value found by our algorithm is close to the upper bound computed by CPLEX, then we can infer with confidence that the result by our algorithm is even closer to the optimal solution (thus highly competitive).

In addition to comparison between our algorithm and an upper bound from CPLEX, we also compare our algorithm with a simple DoF allocation algorithm where each node allocates DoF simply based on the current state of its neighbors without performing local

[3]When using CPLEX to solve MILP problems, it always yields a lower bound (a feasible solution) and an upper bound for the objective value. The gap between the lower and the upper bounds becomes smaller as time continues. If the lower bound coincides with the upper bound, then an optimal solution is found.

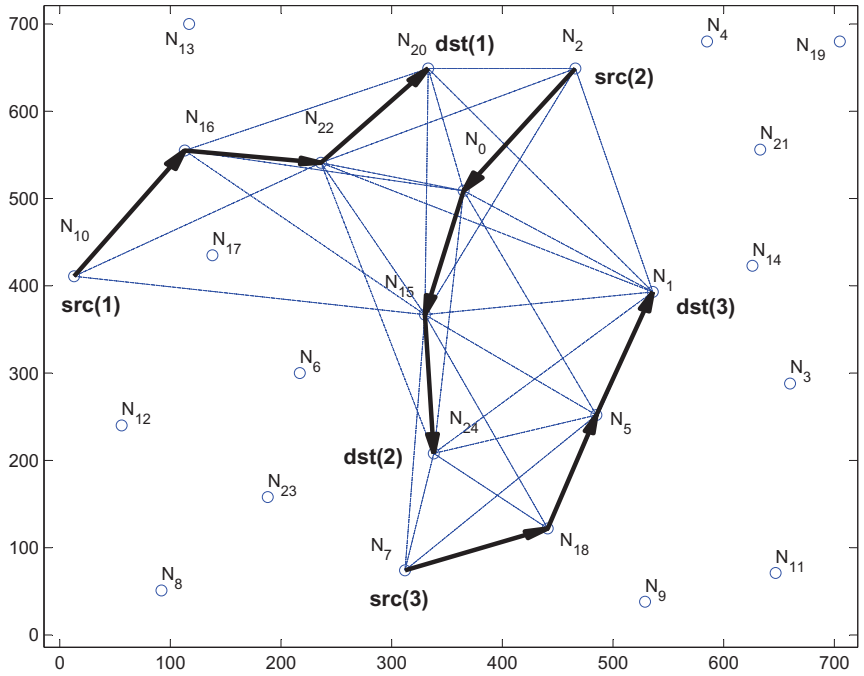
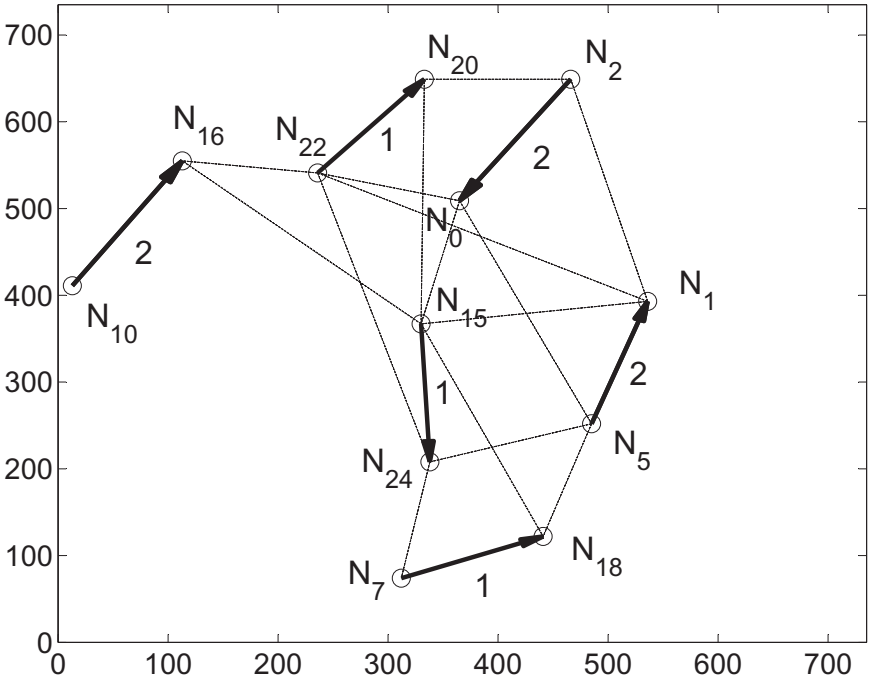


Figure 2.9: A 25-node network instance.

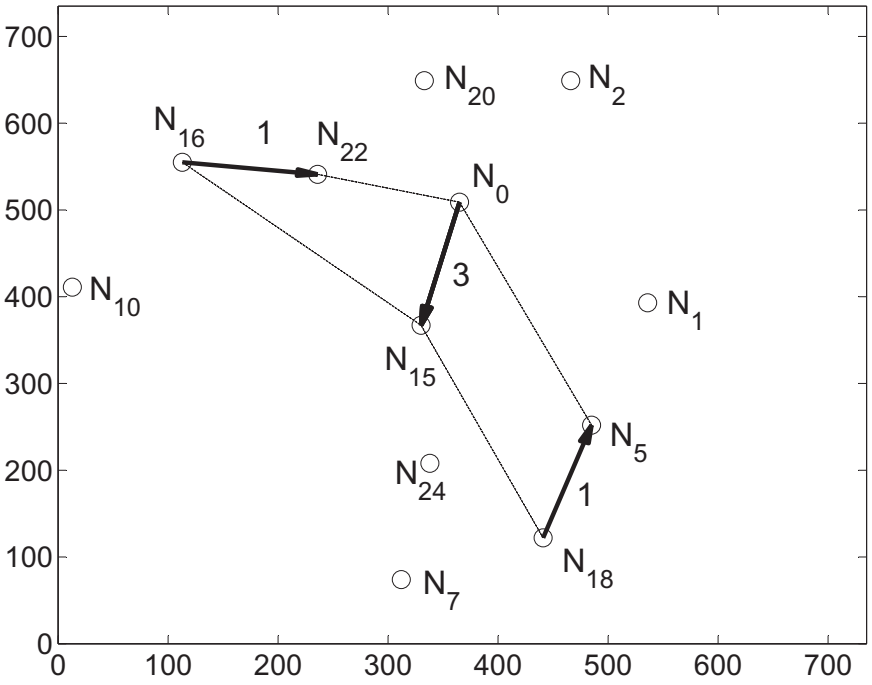
ordering adjustment. Such a simple DoF allocation algorithm is equivalent to our algorithm when its LRM is disabled. Therefore, the objective value from the simple DoF allocation algorithm can be obtained through running our algorithm with its LRM disabled.

2.8.1 Simulation Setting

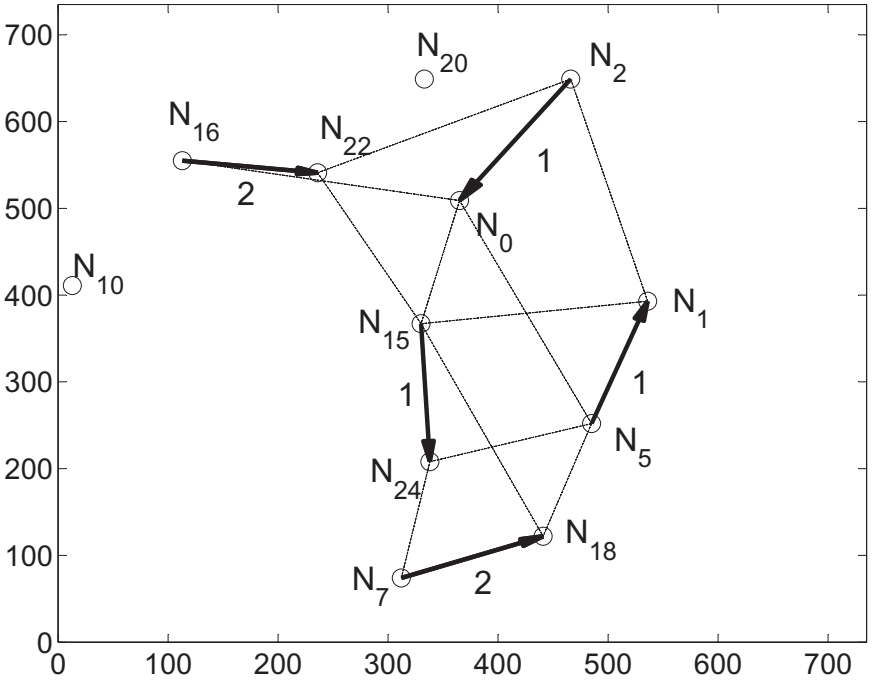
For ease of exposition, we normalize all units for distance, time, bandwidth, and data rate with appropriate dimensions. We consider networks of three sizes: (i) 25 nodes in a 750×750 area with 3 sessions; (ii) 50 nodes in a 1000×1000 area with 4 sessions; and (iii) 100 nodes in a 1500×1500 area with 5 sessions. We assume that all transmit nodes have the same transmission range 180 and the same interference range 360. For each network size, 100 randomly generated network instances are studied. For each network instance, the source and destination nodes of each session are randomly selected, with the route between them being shortest path route. We assume that each node is equipped with four antennas and there are four time slots in a time frame.



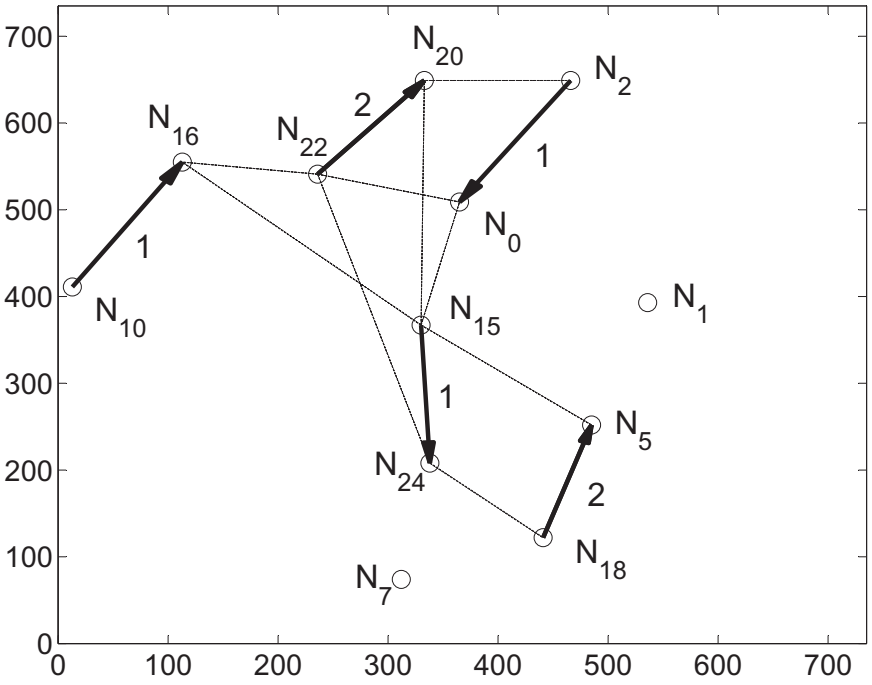
(a) Time slot 1.



(b) Time slot 2.



(c) Time slot 3.



(d) Time slot 4.

Figure 2.10: Active links in each time slot in the case study.

Table 2.4: Local ordering and DoF consumption at each node in the first time slot.

Node i	$\mathcal{I}_i(t)$	$\mathcal{J}_i(t)$	$\lambda_i^{\mathrm{SM}}(t)$	$\lambda_i^{\mathrm{IC}}(t)$
N_0	$\{N_{22}\}$	$\{N_{15}, N_5\}$	2	1
N_1	$\{N_{15}, N_{22}\}$	$\{N_2\}$	2	2
N_2	$\{N_1\}$	$\{N_{20}\}$	2	2
N_5	$\{N_0\}$	$\{N_{18}, N_{24}\}$	2	2
N_7	$\{N_{24}\}$	$\emptyset$	1	1
N_{10}	$\emptyset$	$\emptyset$	2	0
N_{15}	$\{N_0\}$	$\{N_{18}, N_1, N_{16}, N_{20}\}$	1	2
N_{16}	$\{N_{15}, N_{22}\}$	$\emptyset$	2	2
N_{18}	$\{N_{15}, N_5\}$	$\emptyset$	1	3
N_{20}	$\{N_2, N_{15}\}$	$\emptyset$	1	3
N_{22}	$\emptyset$	$\{N_0, N_1, N_{16}, N_{24}\}$	1	0
N_{24}	$\{N_5, N_{22}\}$	$\{N_7\}$	1	3

Table 2.5: A global ordering in each time slot.

t	A global node ordering $\pi(t)$
1	$[N_{10}, N_{22}, N_0, N_5, N_{15}, N_{16}, N_{18}, N_{24}, N_1, N_2, N_7, N_{20}]$
2	$[N_0, N_5, N_{16}, N_{22}, N_{15}, N_{18}]$
3	$[N_2, N_5, N_7, N_{16}, N_{24}, N_0, N_{15}, N_{18}, N_{22}, N_1]$
4	$[N_2, N_{10}, N_{15}, N_{20}, N_{22}, N_{24}, N_0, N_5, N_{16}, N_{18}]$

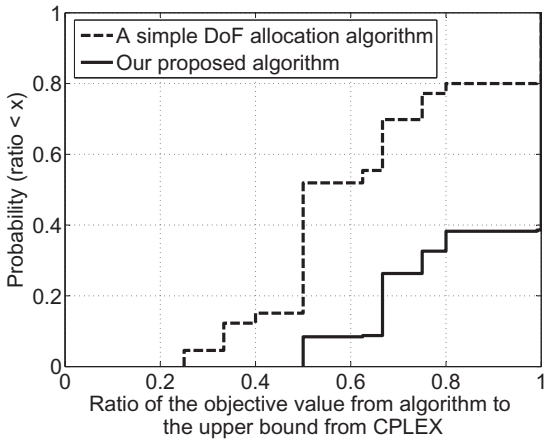
2.8.2 A Case Study

In this subsection, we study a 25-node network instance as shown in Fig. 2.9. In this figure, the solid arrow line represents link while the dashed line represents potential interference. There are three sessions in this network as shown in the figure. Figure 2.10 gives the details of the solution found by our algorithm in four time slots. Let's consider the first time slot as an example. The set of active links are (N_{10}, N_{16}), (N_{22}, N_{20}), (N_2, N_0), (N_{15}, N_{24}), (N_7, N_{18}), and (N_5, N_1). The two local node sets for each node are given in Table 2.4. For example, for node N_0, it considers N_{22} before itself (i.e., $\mathcal{I}_{N_0}(1) = \{N_{22}\}$) while N_{15} and N_5 after itself (i.e., $\mathcal{J}_{N_0}(1) = \{N_{15}, N_5\}$). Node N_0 uses two DoFs for SM to receive two data streams from N_2 (i.e., $\lambda_{N_0}^{\mathrm{SM}}(1) = 2$) and uses one DoF to cancel interference from N_{22} (i.e., $\lambda_{N_0}^{\mathrm{IC}}(1) = 1$). Also for each time slot, we can find a global node ordering for the DoF scheduling solution based on each node's local ordering. Such a global ordering for each time slot is shown in Table 2.5. It is easy to verify that for each time slot, each node has enough DoFs for SM and IC based on this global ordering, indicating that the final DoF scheduling solution is globally feasible.

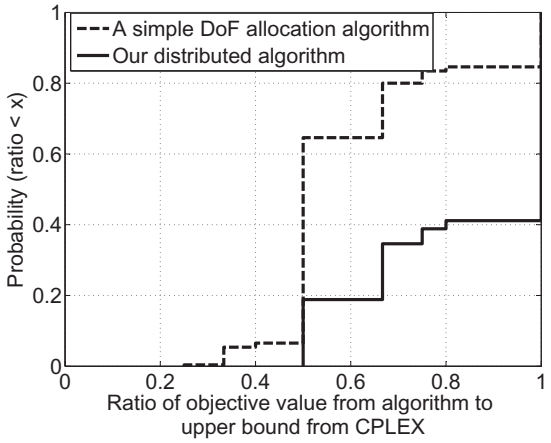
The objective value found by our algorithm is 0.75, corresponding to 3 data streams in 4 time slots for a bottleneck session. The upper bound by CPLEX is also 0.75, which shows that our solution is optimal in this case study. The objective value found by the simple DoF allocation algorithm is 0.5, which means our algorithm has 50% throughput improvement compared to this simple DoF allocation algorithm.

2.8.3 Complete Results

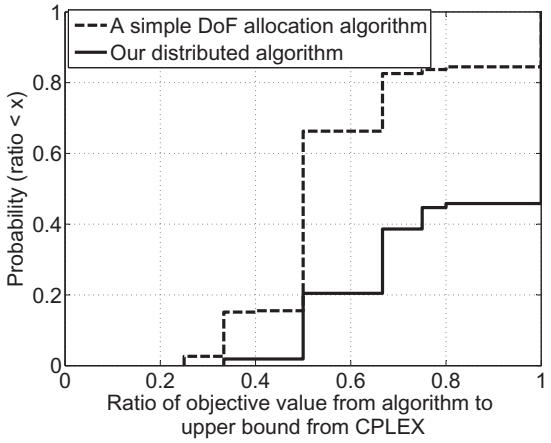
We now present extensive simulation results (200 network instances for each network size) to show the performance of our algorithm. Figure 2.11 present the CDF of the ratio of the objective value found by algorithm to the upper bound obtained by CPLEX. In each sub-figure, there are two curves: the solid one is the performance of our proposed algorithm and the dashed one is the performance of the simple DoF allocation algorithm. We can see that the CDF curve is not smooth but with stair-shape. This is because the feasible objective



(a) 25-node network



(b) 50-node network



(c) 100-node network

Figure 2.11: The CDF of the ratio of the objective value from two algorithms to the upper bound from CPLEX.

value of the problem in Fig. 2.3 is discrete by nature. We now analyze those results in the average sense. The objective value found by our proposed algorithm can achieve 84.1% of the upper bound obtained by CPLEX in 25-node network, 83.2% in 50-node network, and 81.5% in 100-node network. Since the optimal solution lies between the feasible solution by our algorithm and the upper bound by CPLEX solver, we conclude that our solution is very close to the optimum. Moreover, the average objective value found by our proposed algorithm has 35.2% improvement over the simple DoF scheduling algorithm in 25-node network, 38.3% improvement in 50-node network, and 40.3% improvement in 100-node network.

2.9 Chapter Summary

A recent advance in MIMO link model allows research of MIMO on a network scale (i.e., multi-hop MIMO networks). However, such new MIMO link model hinges upon a global node ordering to keep track of IC responsibilities among the nodes, which is centralized by default. The goal of this chapter is to show that it is possible to develop a distributed MIMO scheduling algorithm that achieves both global IC feasibility and an implicit global node ordering even if all operations are performed at a local level among neighboring nodes. The proposed MIMO DoF scheduling algorithm has both a greedy component as well as an aggressive component to counter potential trap of a local optimum. Simulation results show that it is able to achieve objective values very close to upper bound results by CPLEX, which is a very stringent benchmark to measure competitiveness.

Chapter 3

Spatial Interference Alignment in Multi-hop MIMO Networks

3.1 Introduction

Interference management is a fundamental problem in wireless networks. Interference alignment is a major advance in interference management in recent years that offers a new direction to handle mutual interference among different users. The basic idea of IA is to construct signals at transmitters so that these signals overlap (align in the same direction) at their unintended receivers while they are resolvable at their intended receivers. It was shown in [9] that IA allows $K/2$ degrees of freedom (DoF) to be achieved for the K-user interference channel. It was also shown in [23, 53] that IA can significantly increase the user throughput in practical MIMO WLAN. Given its huge potential in improving throughput in wireless networks, IA has brought tremendous attention in the wireless communications community and has become an important research topic.

Due to its intrinsic nature of being a physical layer technique, most of the results are limited in the point-to-point or single-hop scenario. There is a lack of advance of IA technique from networking perspective, in particular, in the context of *multi-hop* wireless networks. Such an extension does not appear to be straightforward, as interference pattern in a multi-

hop network is much more complex and can easily become intractable. In [51], Li et al. made the first attempt to explore IA in a multi-hop MIMO network. The idea of IA was discussed in several example scenarios in the paper to illustrate its benefits. However, the key concept of IA (i.e., constructing signals at transmitters so that these signals overlap at their unintended receivers while remaining resolvable at their intended receivers) was not incorporated into their problem formulation and the final solution. In another effort in [37], only transmitter-side zero-forcing technique was considered for video transmission in cognitive radio networks, rather than the key concept of IA.

The lack of results of IA in multi-hop networks underscores both the technical barrier in this area and the critical need to close this gap by the research community. The goal of this chapter is to make a concrete step toward advancing IA technique in multi-hop MIMO networks. We study IA in its most basic form [9], i.e., the construction of transmit data streams so that (i) they overlap at their unintended receivers and (ii) they remain resolvable at their intended receivers. The construction of transmit data streams requires the design of precoding vector for each data stream at its transmitter. Since the interfering streams overlap at the receivers, one can use fewer number of DoFs to cancel these interfering streams. As a result, the DoF resources consumed for IC can be reduced and more DoF resources can be made available to transport data streams. The main contributions of this chapter are summarized as follows.

- We develop an analytical IA model for a multi-hop MIMO network. Our model consists of a set of constraints at each transmitter to determine the subset of interfering streams used for IA and a set of constraints at each receiver to determine the alignment pattern of the interfering streams.
- We prove the feasibility of the proposed analytical IA model. Specifically, we show that a feasible DoF vector exists at the physical layer as long as the IA constraints are satisfied. This is done by constructing the precoding and decoding vectors for each data stream such that all data streams in this DoF vector can be transported free of interference.

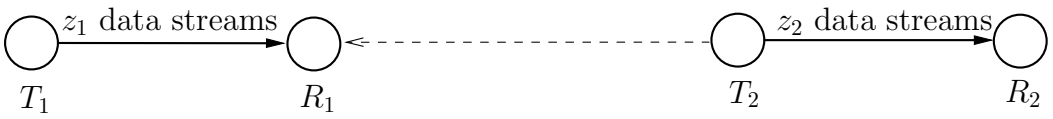


Figure 3.1: An example that illustrates SM, IC, and DoF in MIMO.

- Based on the analytical IA model, we develop a set of constraints across multiple layers of a multi-hop MIMO network. Collectively, these constraints form an IA optimization framework for a multi-hop MIMO network. Under this framework, IA can be exploited to the fullest extent for a target network performance objective.
- To evaluate the efficacy of the IA optimization framework, we study a network throughput optimization problem and compare the results to those for the same problem when IA is not employed. We show that the use of IA can conserve DoF resources in the network and achieve much higher throughput objective.

The remainder of this chapter is organized as follows. Section 3.2 offers some essential background on IA in MIMO networks. Section 3.3 discusses the challenges of applying IA in multi-hop networks. Section 3.4 presents a new analytical IA model for MIMO networks and Section 3.5 proves the feasibility of this model. In Section 3.6, we apply the IA model to a multi-hop MIMO network and develop a cross-layer optimization framework. In Section 3.7, we apply the IA optimization framework to a throughput maximization problem and demonstrate the benefits of IA in a multi-hop MIMO network. Section 3.8 presents related work and Section 3.9 concludes this chapter. Table 3.1 lists the notation used in this chapter.

3.2 Preliminaries: IA in MIMO

In this section, we review MIMO in terms of its DoF resources for spatial multiplexing (SM) and interference cancellation (IC). We also review how IA can help conserve DoF consumption required for IC.

MIMO's DoF Resources for SM and IC. The concept of DoF was originally defined

Table 3.1: Notation used for spatial IA in multi-hop MIMO networks.

Symbol	Definition
$\mathcal{A}_{ij}$	The set of interfering streams from transmitter T_i to receiver R_j
$\mathbf{A}_{ij}$	The set of precoding vectors that correspond to the interfering streams in $\mathcal{A}_{ij}$
$\mathcal{B}_{ij}$	The subset of interfering streams in $\mathcal{A}_{ij}$ that are aligned to other interfering streams at receiver R_j
$\mathbf{B}_{ij}$	The set of precoding vectors that correspond to the interfering streams in $\mathcal{B}_{ij}$
$\mathbf{B}$	The set of precoding vectors that correspond to all interfering streams used for alignment in the network
$\mathbf{D}_j^{\mathrm{S}}$	The set of data stream directions at receiver R_j
$\mathbf{D}_j^{\mathrm{I}}$	The set of interfering stream directions at receiver R_j
$\mathbf{D}_j^{\mathrm{I,eff}}$	The set of "effective" interfering stream directions at R_j
$\mathbf{D}_j^{\mathrm{I,algn}}$	The set of interfering stream directions used for alignment at receiver R_j
$\mathbf{e}_k$	A vector with 1 for the k-th element and 0 for the rest
$\mathcal{F}$	The set of sessions in the network
F	The number of sessions in the network
$\mathbf{H}_{ji}$	Channel matrix between receiver R_j and transmitter T_i
$\mathcal{I}_i$	The set of nodes within node i's interference range
K	The number of time slots in each time frame
L	The number of links in the network
$\mathcal{L}$	The set of links in the network
$\mathcal{L}_i^{\mathrm{in}}$	The set of incoming links at node i
$\mathcal{L}_i^{\mathrm{out}}$	The set of outgoing links at node i
$\mathcal{N}$	The set of nodes in the network
N	The number of nodes in the network
N_{A}	The number of antennas at each node

Table 3.1: Continued.

N_{R}	The number of receiving nodes in a time slot
N_{T}	The number of transmitting nodes in a time slot
$r_{\min}$	The minimum data rate among all sessions
$r(f)$	The data rate of session f
$r_l(f)$	The amount of rate on link l that is used to session f
$\mathrm{Rx}(l)$	The receiver of link l
s_{ij}^k	A stream (intended or unintended) from transmitter T_i to receiver R_j
$\mathcal{S}_{ij}$	The set of data streams from transmitter T_i to R_j
$\mathbf{S}_{ij}$	The set of precoding vectors that correspond to the interfering streams in $\mathcal{S}_{ij}$
$\mathbf{U}$	The set of precoding vectors at all transmitters
$\mathbf{u}_i^k$	The precoding vector of the k-th stream from T_i
$\mathbf{v}_j^l$	The decoding vector of the k-th stream from T_i
$x_i(t)$	A binary variable to indicate whether node i is a transmitter in time slot t
$y_i(t)$	A binary variable to indicate whether node i is a receiver in time slot t
z_l	The number of data streams on link l
α_{ij}	The cardinality of $\mathcal{A}_{ij}$
β_{ij}	The cardinality of $\mathcal{B}_{ij}$
σ_{ij}	The cardinality of $\mathcal{S}_{ij}$
λ_i	The number of outgoing data streams at transmitter T_i
μ_j	The number of incoming data streams at receiver R_j
φ	A DoF vector for the network, i.e., $\varphi = (z_1, z_2, \cdots, z_L)$

to represent the maximum multiplexing gain of an MIMO channel by the information theory community (see e.g., [103]). It was then extended by the networking research community to characterize a node's spatial freedom provided by its multiple antennas (see e.g., [29, 78, 86]). Typically, the total number of DoFs of a node is equal to the number of antennas at the node and represents the total available resources at the node that can be used for SM and IC. SM refers to the use of one or multiple DoFs (both at transmit and receive nodes) for data stream transmission/reception, with each DoF corresponding to one independent data stream. IC refers to the use of one or more DoFs to cancel interference, with each DoF being responsible for canceling one interfering stream. IC can be done either at a transmit node (to cancel interference to a receive node) or a receive node (to cancel interference from a transmit node). For example, consider the two links in Fig. 3.1. To transmit z_1 data streams on link (T_1, R_1), both nodes T_1 and R_1 need to consume z_1 DoFs for SM. Similarly, to transmit z_2 data streams on link (T_2, R_2), both nodes T_2 and R_2 need to consume z_2 DoFs for SM. The interference from T_2 to R_1 can be canceled by either R_1 or T_2. If R_1 cancels this interference, it needs to consume z_2 DoFs. If T_2 cancels this interference, it needs to consume z_1 DoFs.

IA in MIMO. In the context of MIMO, IA refers to the construction of data streams at the transmitters so that (i) they overlap (align) at their unintended receivers and (ii) they remain resolvable at their intended receivers [9]. The construction of transmit data streams is equivalent to the design of precoding vector for each data stream at each transmitter. Since the interfering streams overlap at a receiver, one can use fewer number of DoFs to cancel these interfering streams. As a result, the DoF resources consumed for IC will be reduced and more DoF resources become available for data transport.

We use the following example to illustrate the benefits of IA in MIMO networks. Consider the 4-link network shown in Fig. 3.2. Assume that each node is equipped with three antennas. Suppose that there are two data streams on link (T_1, R_1), two data streams on link (T_2, R_2), and one data stream on link (T_3, R_3). At transmitter T_i, denote $\mathbf{u}_i^k$ as the precoding vector for its outgoing data stream k. Denote $\mathbf{H}_{ji}$ as the channel matrix between receiver R_j and

transmitter T_i. We assume that $\mathbf{H}_{ji}$ is of full rank, which is always true in practical networks.

When IA is not employed, R_4 needs to consume five DoFs to cancel the interference from transmitters T_1, T_2, and T_3 [29, 86]. Since there are only three DoFs available at receiver R_4, it is not possible to cancel all five interfering streams, let alone to receive any data stream from T_4. But when IA is used (see Fig. 3.2), we can align the five interfering streams into two dimensions, which can be canceled by R_4 with only two DoFs. Hence, R_4 still has one DoF remaining, allowing it to receive one data stream from transmitter T_4.

We give one possible approach to construct the five precoding vectors at transmitter T_1, T_2, and T_3, respectively. To show that the five interfering streams can indeed be aligned into two dimensions at receiver R_4, we denote $\mathbf{a} := \mathbf{b}$ if there exists a nonzero complex number c for vectors $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{a} = c \cdot \mathbf{b}$. We begin by constructing the precoding vectors at transmitter T_1 by letting $\mathbf{u}_1^1 := \mathbf{e}_1$ and $\mathbf{u}_1^2 := \mathbf{e}_2$, where $\mathbf{e}_k$ is a vector with the k-th element being 1 and all the other elements being 0. For the two precoding vectors $[\mathbf{u}_2^1\ \mathbf{u}_2^2]$ at transmitter T_2, we align the interfering stream corresponding to $\mathbf{u}_2^1$ to the interfering stream corresponding to $\mathbf{u}_1^1$ at receiver R_4. This can be done by letting $\mathbf{H}_{42}\mathbf{u}_2^1 := \mathbf{H}_{41}\mathbf{u}_1^1$ and thus $\mathbf{u}_2^1 := \mathbf{H}_{42}^{-1}\mathbf{H}_{41}\mathbf{u}_1^1$. Similarly, we can align the interfering stream corresponding to $\mathbf{u}_2^2$ to the interfering stream corresponding to $\mathbf{u}_1^2$ at receiver R_4. This is done by letting $\mathbf{H}_{42}\mathbf{u}_2^2 := \mathbf{H}_{41}\mathbf{u}_1^2$ and thus $\mathbf{u}_2^2 := \mathbf{H}_{42}^{-1}\mathbf{H}_{41}\mathbf{u}_1^2$. Finally, for the precoding vector $\mathbf{u}_3^1$ at transmitter T_3, we can align its interfering stream to the interfering stream corresponding to $\mathbf{u}_1^1$ at receiver R_4. This is done by having $\mathbf{H}_{43}\mathbf{u}_3^1 := \mathbf{H}_{41}\mathbf{u}_1^1$ and thus $\mathbf{u}_3^1 := \mathbf{H}_{43}^{-1}\mathbf{H}_{41}\mathbf{u}_1^1$. As a result of IA, the five interfering streams are aligned into only 2 dimensions and can be canceled with two DoFs (instead of five DoFs) by receiver R_4.

Note that in this example, we only illustrate how to achieve IA at one receiver. In a multi-hop network, the goal is to accomplish IA at as many receivers as possible so as to maximally harvest the benefits of IA. This requires careful coordination at network level and is a much harder problem, as we elaborate in the next section.

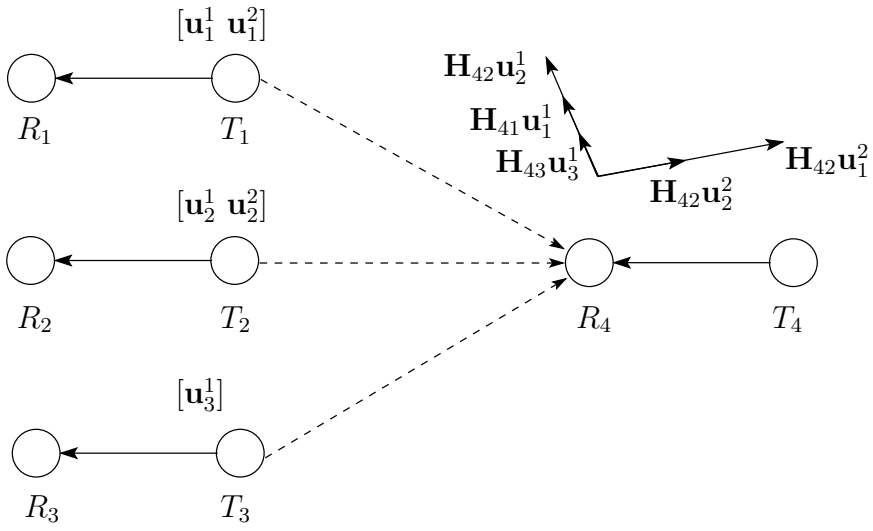


Figure 3.2: An illustration of IA at node R_4.

3.3 IA in Multi-hop Networks: Challenges

As discussed before, although there is a flourish of research on IA in the point-to-point or single-hop scenarios, results on extending IA to a multi-hop network remain very limited. This is because there are a number of new challenges, which we summarize as follows.

(i) How to coordinate IA among a large number of nodes in an MIMO network is a very hard problem. In particular, for each pair of nodes, one needs to decide which subset of interfering streams for IA and how to align them successfully at the receiver. While performing IA, one must also ensure that the desired data streams at each intended receiver remain resolvable. The answers to these questions require the development of a new IA model, as we shall present in this chapter.

(ii) Proving the feasibility of an IA model is not trivial task. One must show that any DoF vector can be supported in the network as long as it satisfies the constraints in the underlying IA model. Specifically, one needs to show that for each data stream characterized by the DoF vector, there exist a precoding vector at its transmitter and a decoding vector at its receiver so that this data stream can be transported free of interference. As we will see in Section 3.5, constructing such a precoding vector and decoding vector for each data stream is very challenging.

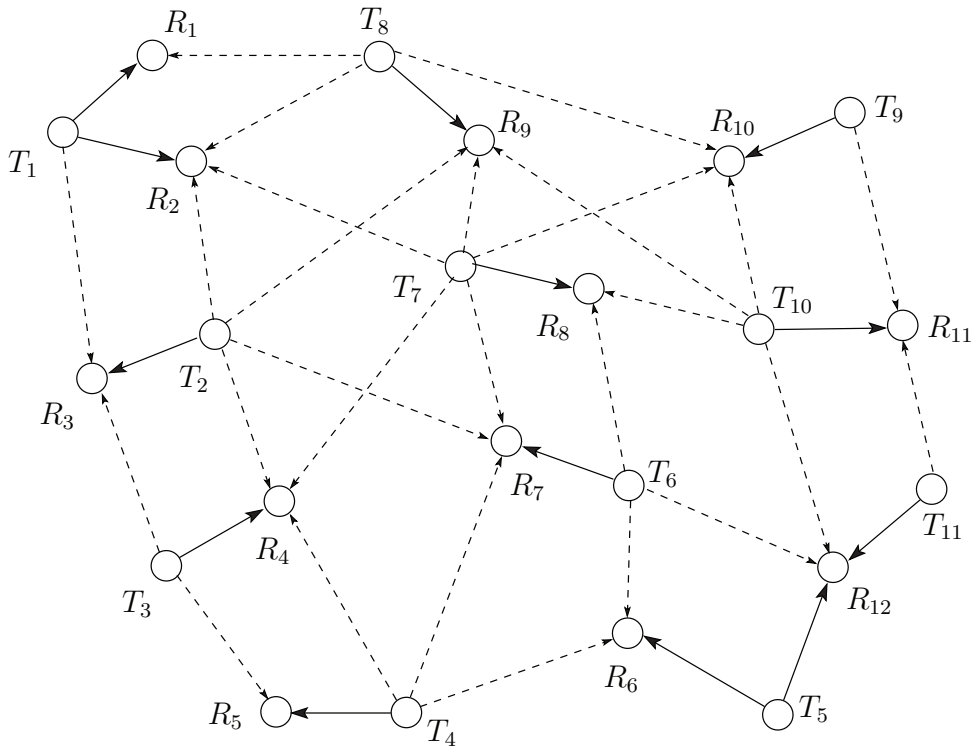


Figure 3.3: A MIMO network that illustrates IA.

(iii) In a multi-hop environment, an IA scheme is also coupled with the upper layer scheduling and routing algorithms. The upper layer algorithms determine the set of transmitters, the set of receivers, the set of links, and the number of data streams on each link, which vary from each time slot. Thus, an IA scheme must be jointly designed with upper layer scheduling and routing algorithms, which is again a challenging problem.

In this chapter, we address challenges (i) and (ii) in Section 3.4 and 3.5, respectively. Challenge (iii) is addressed in Section 3.6.

3.4 Modeling IA in MIMO Networks

Consider a multi-hop MIMO network in Fig. 3.3. Each node is equipped with N_{A} antennas. Assume that scheduling is done in time domain, with each time frame having K time slots. To develop an IA model, we focus on one time slot t ($1 \leq t \leq K$). Denote N_{T} as the number

of transmitters and N_{R} as the number of receivers in the time slot.[1] Denote $\mathcal{L}$ as the set of links in the network with $L = |\mathcal{L}|$. Denote $\varphi = (z_1, z_2, \cdots, z_L)$ as the DoF vector in the network, where z_l is the number of data streams on link $l \in \mathcal{L}$.[2] At each transmitter, a single data stream can only be sent to one receiver. But multiple different data streams from the same transmitter may go to different receivers (see T_1 in Fig. 3.3 for example).

For transmitter T_i, denote λ_i as the number of outgoing data streams and thus we have $\lambda_i = \sum_{l \in \mathcal{L}_i^{\mathrm{out}}} z_l$, where $\mathcal{L}_i^{\mathrm{out}}$ is the set of outgoing links from transmitter T_i. Similarly, for receiver R_j, denote μ_j as the number of its incoming data streams and thus we have $\mu_j = \sum_{l \in \mathcal{L}_j^{\mathrm{in}}} z_l$, where $\mathcal{L}_j^{\mathrm{in}}$ is the set of its incoming links to receiver R_j.

Consider a node pair (T_i, R_j). Denote s_{ij}^k as the transmission of stream k $(1 \leq k \leq \lambda_i)$ from transmitter T_i to receiver R_j. If this stream k is intended to receiver R_j, then s_{ij}^k is a *data stream* for receiver R_j. Otherwise, stream s_{ij}^k is an *interfering stream* for receiver R_j. Denote $\mathcal{S}_{ij}$ as the set of *data streams* from transmitter T_i to receiver R_j, with $\sigma_{ij} = |\mathcal{S}_{ij}|$. Denote $\mathcal{A}_{ij}$ as the set of *interfering streams* from transmitter T_i to receiver R_j, with $\alpha_{ij} = |\mathcal{A}_{ij}|$. Thus, we have $\sigma_{ij} + \alpha_{ij} = \lambda_i$ between transmitter T_i and receiver R_j. Note that without IA, receiver R_j needs to consume α_{ij} DoFs to cancel the interfering streams from transmitter T_i [29, 86].

For receiver R_j, to reduce its DoF consumption for IC, we can align a subset of its interfering streams to the other interfering streams by properly constructing their precoding vectors (as illustrated by Fig. 3.2). Among the interfering streams in $\mathcal{A}_{ij}$, denote $\mathcal{B}_{ij}$ as the subset of interfering streams that are aligned to the other interfering streams at receiver R_j, with $\beta_{ij} = |\mathcal{B}_{ij}|$. Then the number of "effective" interfering streams from transmitter T_i to receiver R_j is decreased from α_{ij} to $\alpha_{ij} - \beta_{ij}$.

The question to ask is how to perform IA among the nodes in the network so that

- **(C-1):** each interfering stream in $\mathcal{B}_{ij}$'s can be successfully aligned at the receivers;
- **(C-2):** each data stream remains resolvable at its intended receiver.

Sections 3.4.1 and 3.4.2 address this question by exploring constraints at a transmitter

[1] When there is no ambiguity, we omit the time slot index t in this section and the next section.

[2] The activity of link l is determined by the value of z_l. When $z_l = 0$, link l is inactive.

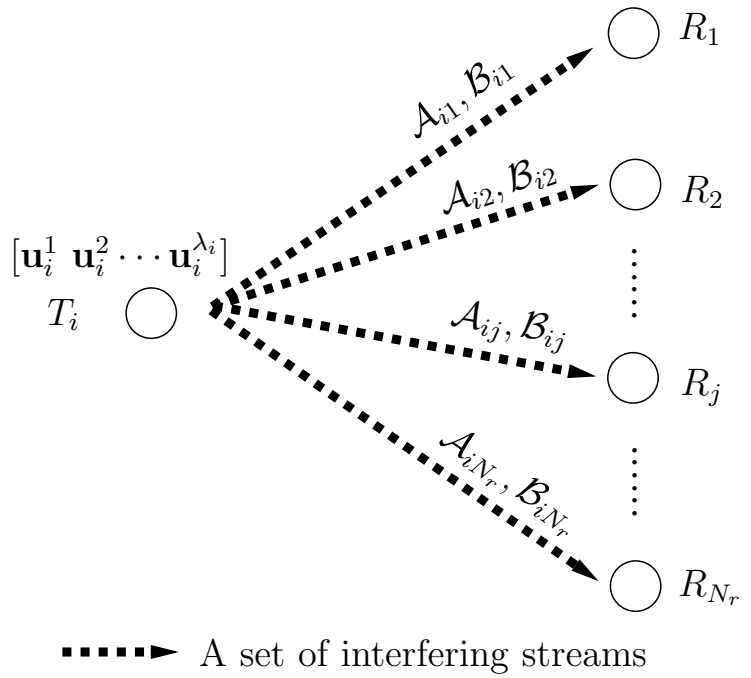


Figure 3.4: IA constraints at transmitter T_i.

and at a receiver, respectively.

3.4.1 IA Constraints at a Transmitter

Consider a transmitter T_i as shown in Fig. 3.4. Based on the definitions of $\mathcal{A}_{ij}$ and $\mathcal{B}_{ij}$, we know $\mathcal{B}_{ij} \subseteq \mathcal{A}_{ij}$. Thus, we have the following constraints at transmitter T_i:

$$\beta_{ij} \leq \alpha_{ij}, \quad j \in \mathcal{I}_i, 1 \leq i \leq N_{\mathrm{T}}, \tag{3.1}$$

where $\mathcal{I}_i$ is the set of nodes within the interference range of node i.

At transmitter T_i in Fig. 3.4, there are λ_i precoding vectors corresponding to λ_i outgoing streams. Since each outgoing stream interferes with all the unintended receivers within transmitter T_i's interference range, the corresponding precoding vector determines the direction of one interfering stream for each of those receivers. For instance, precoding vector $\mathbf{u}_i^1$ determines the directions of the outgoing stream on receiver R_1, R_2, $\cdots$, R_{N_r}, one of which is the intended receiver and the rest are unintended receivers (i.e., interference). However, among the $N_r - 1$ directions for interfering streams, only one of them can be successfully

aligned to a particular direction for IA by constructing $\mathbf{u}_i^1$. Therefore, for the interfering streams from transmitter T_i, at most λ_i interfering streams can be successfully used for IA at their receivers, since there are λ_i precoding vectors at transmitter T_i. Mathematically, we have the following constraints at transmitter T_i:

$$\sum_{j\in\mathcal{I}_i}\beta_{ij}\leq\lambda_i,\qquad 1\leq i\leq N_{\mathrm{T}}. \tag{3.2}$$

At transmitter T_i, the DoF consumption is only for SM. Specifically, the number of DoFs consumed at transmitter T_i is equal to the number of its outgoing data streams (i.e., λ_i). Since the DoFs consumed at a node cannot exceed its total DoFs, we have the following constraints at transmitter T_i:

$$\lambda_i\leq N_{\mathrm{A}},\qquad 1\leq i\leq N_{\mathrm{T}}. \tag{3.3}$$

3.4.2 IA Constraints at a Receiver

Consider a receiver R_j in Fig. 3.5. To ensure (C-1) and (C-2) at receiver R_j, we have the following conditions:

- Based on our definition of $\mathcal{B}_{ij}$, the interfering streams in each $\mathcal{B}_{ij}$ should not occupy "effective" directions at receiver R_j. Therefore, at receiver R_j, each interfering stream in $\cup_{i\in\mathcal{I}_j}\mathcal{B}_{ij}$ can only be aligned to an interfering stream in $\cup_{i\in\mathcal{I}_j}(\mathcal{A}_{ij}\backslash\mathcal{B}_{ij})$.
- To ensure the resolvability of the data streams at each receiver, we must have that any interfering stream in $\mathcal{B}_{ij}$ cannot be aligned to an interfering stream in $\mathcal{A}_{ij}$. To show the reason, suppose that s_{ij}^k in $\mathcal{B}_{ij}$ is aligned to $s_{ij}^{k'}$ in $\mathcal{A}_{ij}$ at receiver R_j. Then, we have $\mathbf{u}_i^k:=\mathbf{H}_{ji}^{-1}\mathbf{H}_{ji}\mathbf{u}_i^{k'}:=\mathbf{u}_i^{k'}$. This means that $\mathbf{u}_i^k$ and $\mathbf{u}_i^{k'}$ are linearly dependent and consequently these two streams are not resolvable at their intended receivers.
- To ensure the resolvability of the data streams at each receiver, we must have that any two interfering streams in $\mathcal{B}_{ij}$ cannot be aligned to the same (a third) interfering stream. To show the reason, suppose that both s_{ij}^k and $s_{ij}^{k'}$ in $\mathcal{B}_{ij}$ are aligned to $s_{i'j}^l$ at receiver R_j. Then, we have $\mathbf{u}_i^k:=\mathbf{H}_{ji}^{-1}\mathbf{H}_{ji'}\mathbf{u}_{i'}^l$ and $\mathbf{u}_i^{k'}:=\mathbf{H}_{ji}^{-1}\mathbf{H}_{ji'}\mathbf{u}_{i'}^l$. We therefore

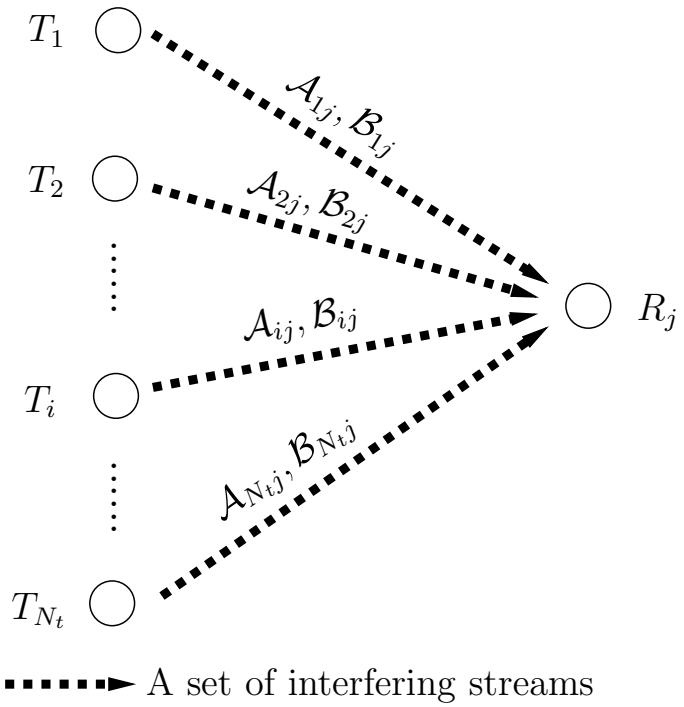


Figure 3.5: IA constraints at receiver R_j.

have $\mathbf{u}_i^k := \mathbf{u}_i^{k'}$, indicating that $\mathbf{u}_i^k$ and $\mathbf{u}_i^{k'}$ are linearly dependent. This means that these two streams are not resolvable at their intended receivers.

We shall show that the above three conditions are all satisfied if the following constraints are satisfied at each receiver R_j:

$$\beta_{ij} \leq \sum_{k \in \mathcal{I}_j}^{k \neq i} (\alpha_{kj} - \beta_{kj}), \qquad i \in \mathcal{I}_j, 1 \leq j \leq N_{\mathrm{R}}. \tag{3.4}$$

At each receiver R_j, its DoFs are consumed for SM and IC. Specifically, the number of DoFs consumed for SM is equal to the number of its incoming data streams (i.e., μ_j); the number of DoFs consumed for IC is equal to the number of "effective" interfering streams at this receiver (i.e., $\sum_{i \in \mathcal{I}_j} (\alpha_{ij} - \beta_{ij})$). Since the DoFs consumed for SM and IC cannot exceed its total DoFs, we have the following constraints at receiver R_j:

$$\mu_j + \sum_{k \in \mathcal{I}_j} (\alpha_{kj} - \beta_{kj}) \leq N_{\mathrm{A}}, \qquad 1 \leq j \leq N_{\mathrm{R}}. \tag{3.5}$$

Collectively, constraints (3.1)–(3.5) characterize an analytical IA model for an MIMO network. A question about this model is its feasibility: For a DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$ that meets these IA constraints, is it also feasible? We answer this question in the following section.

3.5 Feasibility of the IA Model

To prove the feasibility of the proposed analytical IA model, we must first clarify what we mean by "feasibility". The following definition clarifies this issue:

Definition 1. *Suppose that a stream k from transmitter T_i is intended to receiver R_j and $\mathbf{v}_j^l$ is its decoding vector at receiver R_j. Then, DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$ is feasible if there exist precoding vector $\mathbf{u}_i^k$ and decoding vector $\mathbf{v}_j^l$ for each stream k from transmitter T_i, $1 \leq i \leq N_{\mathrm{T}}$, $1 \leq k \leq \lambda_i$, such that*

$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji} \mathbf{u}_i^k = 1, \tag{3.6a}$$

$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0, \tag{3.6b}$$

for $1 \leq k' \leq \lambda_{i'}, i' \in \mathcal{I}_j, (i', k') \neq (i, k)$.

Simply put, we say a DoF vector is feasible if there exist precoding and decoding vectors for each stream so that the stream can be decoded at its intended receiver free of interference. The following theorem is the main result of this section.

Theorem 2. *A DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$ is feasible if it satisfies constraints (3.1)–(3.5) in the IA model.*

It is worth pointing out that for a given DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$, the values of α_{ij}, λ_i, and μ_j in the IA constraints are fixed (i.e., $\lambda_i = \sum_{l \in \mathcal{L}_i^{\mathrm{out}}} z_l$, $\mu_j = \sum_{l \in \mathcal{L}_j^{\mathrm{in}}} z_l$, and $\alpha_{ij} = \sum_{l \in \mathcal{L}_i^{\mathrm{out}}}^{\mathrm{Rx}(l) \neq j} z_l$), while the values of β_{ij} depend on the specific IA scheme that one designs. In the rest of this section, we prove Theorem 2 by construction.

3.5.1 Proof of Theorem 2: A Roadmap

As for notation, we use calligraphic uppercase letter to denote a set of data/interfering streams and use boldface uppercase letter to denote the set of its corresponding precoding vectors. For a set of data streams in $\mathcal{S}_{ij}$, denote $\mathbf{S}_{ij}$ as the corresponding set of precoding vectors. For a set of interfering streams in $\mathcal{A}_{ij}$, denote $\mathbf{A}_{ij}$ as the corresponding set of precoding vectors. For a set of interfering streams in $\mathcal{B}_{ij}$, denote $\mathbf{B}_{ij}$ as the corresponding set of precoding vectors. Mathematically,

$$\begin{aligned}\mathbf{S}_{ij} &= \{\mathbf{u}_i^k : s_{ij}^k \in \mathcal{S}_{ij}\},\\ \mathbf{A}_{ij} &= \{\mathbf{u}_i^k : s_{ij}^k \in \mathcal{A}_{ij}\},\\ \mathbf{B}_{ij} &= \{\mathbf{u}_i^k : s_{ij}^k \in \mathcal{B}_{ij}\}.\end{aligned}$$

Accordingly, we have $|\mathbf{S}_{ij}| = \sigma_{ij}$, $|\mathbf{A}_{ij}| = \alpha_{ij}$, and $|\mathbf{B}_{ij}| = \beta_{ij}$.

Consider receiver R_j shown in Fig. 3.5. Denote $\mathbf{D}_j^{\mathrm{S}}$ as the set of *data stream* directions at receiver R_j. Denote $\mathbf{D}_j^{\mathrm{I}}$ as the set of *interfering stream* directions at receiver R_j. Then we have

$$\begin{aligned}\mathbf{D}_j^{\mathrm{S}} &= \cup_{i\in\mathcal{I}_j}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{S}_{ij}\},\\ \mathbf{D}_j^{\mathrm{I}} &= \cup_{i\in\mathcal{I}_j}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{A}_{ij}\}.\end{aligned}$$

The following lemma shows a sufficient condition for DoF vector φ to be feasible.

Lemma 3. *A DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$ is feasible if there exists a precoding vector $\mathbf{u}_i^k$ for each stream k from transmitter T_i ($1 \le i \le N_{\mathrm{T}}$, $1 \le k \le \lambda_i$), such that*

$$\dim(\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}) = \mu_j + \dim(\mathbf{D}_j^{\mathrm{I}}), \qquad 1 \le j \le N_{\mathrm{R}}, \tag{3.7}$$

where μ_j is the number of incoming data streams at receiver R_j and $\mu_j = \sum_{l\in\mathcal{L}_j^{\mathrm{in}}} z_l$ for $1 \le j \le N_{\mathrm{R}}$.

Proof. We show DoF vector φ is feasible by arguing that if (3.7) is satisfied, then we can find a decoding vector $\mathbf{v}_j^l$ for each stream k from transmitter T_i such that (3.6a) and (3.6b)

are satisfied. Specifically, we show that the following linear system is consistent if (3.7) is satisfied.

$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji} \mathbf{u}_i^k = 1,$$
$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0, \quad i' \in \mathcal{I}_j, 1 \leq k' \leq \lambda_{i'}, (i', k') \neq (i, k).$$

where $\mathbf{v}_j^l$ is variable vector and $\mathbf{H}$'s and $\mathbf{u}$'s are given.

Based on the definition of $\mathbf{D}_j^{\mathrm{S}}$ and $\mathbf{D}_j^{\mathrm{I}}$, we know

$$\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}} = \{\mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} : i' \in \mathcal{I}_j, 1 \leq k' \leq \lambda_{i'}\}.$$

It is easy to see that $\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}$ is the set of coefficient-vectors of this linear system. Moreover, this system has N_{A} free variables and at most N_{A} linearly independent equations. If we can show that the vector $\mathbf{H}_{ji} \mathbf{u}_i^k$ is not a linear combination of other vectors in $\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}$, then this system is consistent. We prove this point by contradiction.

Suppose that $\mathbf{H}_{ji} \mathbf{u}_i^k$ is a linear combination of other vectors in $\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}$. Since $\mathbf{H}_{ji} \mathbf{u}_i^k \in \mathbf{D}_j^{\mathrm{S}}$, we have

$$\dim(\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}) < |\mathbf{D}_j^{\mathrm{S}}| + \dim(\mathbf{D}_j^{\mathrm{I}}) = \mu_j + \dim(\mathbf{D}_j^{\mathrm{I}}).$$

But this contradicts the given condition in (3.7). Therefore, we conclude that the linear system is consistent. □

Intuitively, Lemma 3 tells us that at each receiver R_j, if a data stream lies in an independent direction (i.e., not within the subspace spanned by other data/interfering streams), then this data stream is resolvable. Lemma 3 offers another route for checking the feasibility of a given DoF vector: instead of checking the existence of both precoding and decoding vectors that satisfy (3.6a) and (3.6b) in Definition 1, one only needs to check the existence of the precoding vectors that satisfy (3.7) in Lemma 3.

We give a roadmap for our proof of Theorem 2.

- *Step 1 (Designing An IA Scheme):* Based on the constraints in the IA model, we propose an IA scheme for the network. The objective of this scheme is to ensure that at each receiver R_j, the interfering streams in $\cup_{i \in \mathcal{I}_j} \mathcal{B}_{ij}$ can be successfully aligned to

the interfering streams in $\cup_{i\in\mathcal{I}_j}\mathcal{A}_{ij}\backslash\mathcal{B}_{ij}$. We achieve this objective by addressing two questions: (i) How to select β_{ij} interfering streams from $\mathcal{A}_{ij}$ for $\mathcal{B}_{ij}$ at each transmitter T_i? (ii) How to align the interfering streams in $\mathcal{B}_{ij}$ to other interfering streams at each receiver R_j? Details are given in Section 3.5.2.

- *Step 2 (Constructing Precoding Vectors):* Based on the IA scheme proposed in Step 1, we present an approach to construct the precoding vectors at the transmitters. Specifically, we divide the precoding vectors into two groups: $\mathbf{B}$ and $\mathbf{U}\backslash\mathbf{B}$. For a precoding vector $\mathbf{u}_i^k$ in $\mathbf{U}\backslash\mathbf{B}$, we set $\mathbf{u}_i^k := \mathbf{e}_k$. For the precoding vectors in $\mathbf{B}$, we construct them based on the IA scheme in Step 1. Details are given in Section 3.5.3.
- *Step 3 (Resolving Intended Signals):* We show that the constructed precoding vectors in Step 2 satisfy (3.7) in Lemma 3, thereby concluding that DoF vector $\varphi = (z_1, z_2, \cdots, z_L)$ is feasible. Details are given in Section 3.5.4.

3.5.2 Step 1: Designing An IA Scheme

Based on the constraints in the IA model, we propose an IA scheme at a transmitter and a receiver. The goal of this IA scheme is that at each receiver R_j, the interfering streams in $\cup_{i\in\mathcal{I}_j}\mathcal{B}_{ij}$ can be successfully aligned to the interfering streams in $\cup_{i\in\mathcal{I}_j}\mathcal{A}_{ij}\backslash\mathcal{B}_{ij}$. We present the IA scheme that addresses two questions: (i) At each transmitter T_i, how to select a subset of β_{ij} interfering streams for $\mathcal{B}_{ij}$ from the α_{ij} interfering streams in $\mathcal{A}_{ij}$? (ii) At each receiver R_j, how to align the interfering streams in $\mathcal{B}_{ij}$ to others interfering streams?

Selecting interfering streams for $\mathcal{B}_{ij}$. Consider transmitter T_i in Fig. 3.4. To ensure that the interfering streams in $\cup_{j\in\mathcal{I}_i}\mathcal{B}_{ij}$ can be successfully aligned to particular directions at their receivers, each of them must be corresponding to a unique precoding vector at transmitter T_i. Mathematically, this requirement can be interpreted as

$$\mathbf{B}_{ij_1} \cap \mathbf{B}_{ij_2} = \emptyset, \quad j_1, j_2 \in \mathcal{I}_i, j_1 \neq j_2, 1 \leq i \leq N_{\mathrm{T}}. \tag{3.8}$$

Then, we have the following lemma:

Lemma 4. *For any β_{ij} that meets constraints (3.1) and (3.2), we can select β_{ij} interfering*

streams for $\mathcal{B}_{ij}$ (from the α_{ij} interfering streams in $\mathcal{A}_{ij}$) so that (3.8) is satisfied.

Proof. Proving Lemma 4 is equivalent to solving the precoding vector selection problem (PVS-Problem) as follows:

PVS-Problem: For transmitter T_i and its neighboring receivers as shown in Fig. 3.4, select β_{ij} precoding vectors from $\mathbf{U}_i = \{\mathbf{u}_i^k : 1 \le k \le \lambda_i\}$ for $\mathbf{B}_{ij}$, $j \in \mathcal{I}_i$, such that

$$\mathbf{B}_{ij} \subseteq \mathbf{A}_{ij}, \qquad j \in \mathcal{I}_i^{\mathrm{T}}, \tag{3.9a}$$

$$\mathbf{B}_{ij_1} \cap \mathbf{B}_{ij_2} = \emptyset, \quad j_1, j_2 \in \mathcal{I}_i, j_1 \neq j_2. \tag{3.9b}$$

where $\mathbf{U}_i = \cup_{k\in\mathcal{I}_i}\mathbf{S}_{ik}$, $\mathbf{A}_{ij} = \cup_{k\in\mathcal{I}_i}^{k\neq j}\mathbf{S}_{ik}$, $|\mathbf{S}_{ij}| = \sigma_{ij}$, $\sum_{j\in\mathcal{I}_i}\sigma_{ij} = \lambda_i$, and $\sum_{j\in\mathcal{I}_i}\beta_{ij} \le \lambda_i$.

We solve the PVS-Problem by two steps. First, we propose a greedy algorithm to select precoding vectors for $\mathbf{B}_{ij}$ (for each $j \in \mathcal{I}_i$). Second, we show that the resulting $\mathbf{B}_{ij}$ satisfies constraints (3.9a) and (3.9b).

A greedy algorithm. Without loss of generality, we index the receivers within $\mathcal{I}_i$ from 1 to J, where $J = |\mathcal{I}_i|$. We select precoding vectors for $\mathbf{B}_{ij}$ ($1 \le j \le J$) sequentially. Specifically, we first select β_{i1} precoding vectors for $\mathbf{B}_{i1}$, and then select β_{i2} precoding vectors for $\mathbf{B}_{i2}$, and so forth. In each iteration j, we select β_{ij} precoding vectors for $\mathbf{B}_{ij}$ as follows: For each k within $j < k < J$, we move $(|\mathbf{S}_{ik}| - \sum_{k'=k+1}^{J}\beta_{ik'})^+$ precoding vectors from $\mathbf{S}_{ik}$ to $\mathbf{B}_{ij}$, where $(\cdot)^+ = \max\{\cdot, 0\}$. After that, if $\mathbf{B}_{ij}$ does not have enough precoding vectors, we move the precoding vectors from $\cup_{k\in\mathcal{I}_i}^{k\neq j}\mathbf{S}_{ik}$ to $\mathbf{B}_{ij}$ until $\mathbf{B}_{ij}$ has enough precoding vectors. A pseudo-code for this algorithm is given in Fig. 3.6.

Algorithm analysis. Two observations on the algorithm are in order. First, the resulting solution meets (3.9a), because all precoding vectors in $\mathbf{B}_{ij}$ are selected from $\cup_{k\in\mathcal{I}_i}^{k\neq j}\mathbf{S}_{ik}$ and $\mathbf{A}_{ij} = \cup_{k\in\mathcal{I}_i}^{k\neq j}\mathbf{S}_{ik}$. Second, the resulting solution meets (3.9b), because each precoding vector in $\mathbf{U}_i$ is selected for only one $\mathbf{B}_{ij}$. Therefore, if we can show that the algorithm can successfully select β_{ij} precoding vectors for $\mathbf{B}_{ij}$ in each iteration j, then the PVS-Problem is solved.

Consider the precoding vector selection for $\mathbf{B}_{ij}$ in iteration j. In our algorithm (see Fig. 3.6), any precoding vectors in $\cup_{k\in\mathcal{I}_i}^{k\neq j}\tilde{\mathbf{S}}_{ik}$ can be moved to $\mathbf{B}_{ij}$. Therefore, if we can show that $\tilde{\beta}_{ij} \le \sum_{k\in\mathcal{I}_i}^{k\neq j}\tilde{\sigma}_{ik}$ at the beginning of each iteration j, then the PVS-Problem is solved. We now argue that this is true in different cases.

Algorithm: Solving PVS-Problem at transmitter T_i.

1. $J = |\mathcal{I}_i|$;
2. for $j := 1$ to J {
3. $\quad \tilde{\mathbf{S}}_{ij} = \mathbf{S}_{ij}$; $\mathbf{B}_{ij} = \emptyset$; $\tilde{\beta}_{ij} = \beta_{ij}$; $\tilde{\sigma}_{ij} = \sigma_{ij}$; }
4. for $j := 1$ to J {
5. $\quad$ for $k := j + 1$ to $J - 1$ {
6. $\quad\quad$ if $\tilde{\beta}_{ij} == 0$ {break;}
7. $\quad\quad$ $d = 1^{+}(\tilde{\sigma}_{ik} - \sum_{k'=k+1}^{J} \tilde{\beta}_{ik'})$;
8. $\quad\quad$ move $\min\{\tilde{\beta}_{ij}, d\}$ precoding vectors from $\tilde{\mathbf{S}}_{ik}$ to $\mathbf{B}_{ij}$;
9. $\quad\quad$ $\tilde{\beta}_{ij} := \tilde{\beta}_{ij} - \min\{\tilde{\beta}_{ij}, d\}$;
10. $\quad\quad$ $\tilde{\sigma}_{ik} := \tilde{\sigma}_{ik} - \min\{\tilde{\beta}_{ij}, d\}$; }
11. $\quad$ for $k := 1$ to J {
12. $\quad\quad$ if $\tilde{\beta}_{ij} == 0$ {break;}
13. $\quad\quad$ if $k == j$ {continue;}
14. $\quad\quad$ move $\min\{\tilde{\beta}_{ij}, \tilde{\sigma}_{ik}\}$ precoding vectors from $\tilde{\mathbf{S}}_{ik}$ to $\mathbf{B}_{ij}$;
15. $\quad\quad$ $\tilde{\beta}_{ij} := \tilde{\beta}_{ij} - \min\{\tilde{\beta}_{ij}, \tilde{\sigma}_{ik}\}$;
16. $\quad\quad$ $\tilde{\sigma}_{ik} := \tilde{\sigma}_{ik} - \min\{\tilde{\beta}_{ij}, \tilde{\sigma}_{ik}\}$; }}

Figure 3.6: A pseudo-code for solving PVS-Problem at transmitter T_i.

Case I. $\tilde{\sigma}_{ij} - \sum_{k=j+1}^{J} \tilde{\beta}_{ik} \leq 0$ at the beginning of iteration j. In this case, we have

$$\sum_{k\in\mathcal{I}_i}^{k\neq j} \tilde{\sigma}_{ik} - \tilde{\beta}_{ij} = \sum_{k\in\mathcal{I}_i} \tilde{\sigma}_{ik} - \tilde{\sigma}_{ij} - \tilde{\beta}_{ij} \overset{(a)}{=} \lambda_i - \sum_{k=1}^{j-1} \beta_{ik} - \tilde{\sigma}_{ij} - \tilde{\beta}_{ij} \overset{(b)}{=} \lambda_i - \tilde{\sigma}_{ij} - \sum_{k=1}^{j} \beta_{ik}$$
$$\overset{(c)}{\geq} \sum_{k=j+1}^{J} \beta_{ik} - \tilde{\sigma}_{ij} \overset{(d)}{=} \sum_{k=j+1}^{J} \tilde{\beta}_{ik} - \tilde{\sigma}_{ij} \geq 0,$$

where (a) follows from the fact that $\sum_{k\in\mathcal{I}_i} \tilde{\sigma}_{ik} = \sum_{k\in\mathcal{I}_i} \sigma_{ik} - \sum_{k=1}^{j-1} \beta_{ik} = \lambda_i - \sum_{k=1}^{j-1} \beta_{ik}$ at the beginning of iteration j; (b) and (d) follow from the fact that $\tilde{\beta}_{ik} = \beta_{ik}$ for $j \leq k \leq J$ at the beginning of iteration j; (c) follows from constraint (3.2).

Case II. $\tilde{\sigma}_{ij} - \sum_{k=j+1}^{J} \tilde{\beta}_{ik} > 0$ at the beginning of iteration j. In this case, if there exists a j' such that $j' < j$ and $\tilde{\sigma}_{ij'} = \sum_{k=j'+1}^{J} \tilde{\beta}_{ik}$, then it is easy to see that

$$\sum_{k\in\mathcal{I}_i}^{k\neq j} \tilde{\sigma}_{ik} - \tilde{\beta}_{ij} \geq \tilde{\sigma}_{ij'} - \beta_{ij} \geq 0.$$

Otherwise (i.e., there does not exist such a j'), all precoding vectors in $\cup_{k=1}^{j-1}\mathbf{B}_{ik}$ are from $\tilde{\mathbf{S}}_{ij}$. Then, we have

$$\sum_{k\in\mathcal{I}_i}^{k\neq j} \tilde{\sigma}_{ik} - \tilde{\beta}_{ij} \overset{(a)}{=} \sum_{k\in\mathcal{I}_i}^{k\neq j} \sigma_{ik} - \tilde{\beta}_{ij} \overset{(b)}{=} \alpha_{ij} - \tilde{\beta}_{ij} = \alpha_{ij} - \beta_{ij} \overset{(c)}{\geq} 0,$$

where (a) follows from the fact that $\tilde{\sigma}_{ik} = \sigma_{ik}$ for $j \in \mathcal{I}_i$ and $k \neq j$; (b) follows from the fact that $\sum_{k\in\mathcal{I}_i}^{k\neq j} \sigma_{ik} = \lambda_i - \sigma_{ij} = \alpha_{ij}$; (c) follows from constraint (3.1).

Combining the two different cases, we conclude that the PVS-Problem is solved and Lemma 4 is proved. □

Lemma 4 ensures that each interfering stream in $\mathcal{B}_{ij}$ corresponds to a unique precoding vector and, therefore, each interfering stream in $\mathcal{B}_{ij}$ can be aligned to any particular direction by constructing its corresponding precoding vector.

Aligning the interfering streams in $\mathcal{B}_{ij}$. Consider receiver R_j in Fig. 3.5. For each $i \in \mathcal{I}_j$, we use the following algorithm to align the interfering streams in $\mathcal{B}_{ij}$ at receiver R_j.

At receiver R_j, each interfering stream in $\mathcal{B}_{ij}$ is aligned to a unique interfering stream in $\cup_{k\in\mathcal{I}_j}^{k\neq i}(\mathcal{A}_{kj}\backslash\mathcal{B}_{kj})$, $i \in \mathcal{I}_j$.

Based on (3.4), there are more interfering streams in $\cup_{k\in\mathcal{I}_j}^{k\neq i}(\mathcal{A}_{kj}\backslash\mathcal{B}_{kj})$ than those in $\mathcal{B}_{ij}$. Therefore, each interfering stream in $\mathcal{B}_{ij}$ can be successfully aligned to a unique interfering stream in $\cup_{k\in\mathcal{I}_j}^{k\neq i}(\mathcal{A}_{kj}\backslash\mathcal{B}_{kj})$.

For an interfering stream $s_{ij}^k \in \mathcal{B}_{ij}$, in order to align it to an interfering stream $s_{i'j}^{k'} \in \cup_{k\in\mathcal{I}_j}^{k\neq i}(\mathcal{A}_{kj}\backslash\mathcal{B}_{kj})$ at receiver R_j, we should construct its corresponding precoding vector by $\mathbf{u}_i^k := \mathbf{H}_{ji}^{-1}\mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'}$, which we denote as $\mathbf{u}_i^k \xrightarrow{j} \mathbf{u}_{i'}^{k'}$. The following lemma shows that there is a unique mapping for each precoding vector in $\mathbf{B}_{ij}$.

Lemma 5. *For each $\mathbf{u}_i^k$ in $\mathbf{B}_{ij}$, there exists one and only one $\mathbf{u}_{i'}^{k'}$, such that $\mathbf{u}_i^k \xrightarrow{j} \mathbf{u}_{i'}^{k'}$ with $\mathbf{u}_{i'}^{k'} \in \mathbf{A}_{i'j}\backslash\mathbf{B}_{i'j}$ and $i' \neq i$.*

Lemma 5 is straightforward based on the following two facts: (i) each interfering stream in $\mathcal{B}_{ij}$ is associated with a unique precoding vector (Lemma 4); (ii) each interfering stream in $\mathcal{B}_{ij}$ is aligned to an interfering stream in $\cup_{k\in\mathcal{I}_j}^{k\neq i}(\mathcal{A}_{kj}\backslash\mathcal{B}_{kj})$. We thus omit its proof.

3.5.3 Step 2: Constructing Precoding Vectors

We now explain how to construct the precoding vector for each stream based on the IA scheme in Section 3.5.2. Denote $\mathbf{U}$ as the set of all precoding vectors in the network. Denote $\mathbf{B}$ as the set of the precoding vectors that correspond to the interfering streams for alignment. Mathematically,

$$\mathbf{U} = \{\mathbf{u}_i^k : 1 \leq k \leq \lambda_i, 1 \leq i \leq N_{\mathrm{T}}\},$$
$$\mathbf{B} = \cup_{j\in\mathcal{I}_i, 1\leq i\leq N_{\mathrm{T}}}\mathbf{B}_{ij}.$$

To construct the precoding vectors in $\mathbf{U}$, we divide $\mathbf{U}$ into two groups: $\mathbf{B}$ and $\mathbf{U}\backslash\mathbf{B}$. We first construct the precoding vectors in $\mathbf{U}\backslash\mathbf{B}$ and then construct the precoding vectors in $\mathbf{B}$.

For each precoding vector in $\mathbf{U}\backslash\mathbf{B}$, we construct it as follows:

$$\mathbf{u}_i^k := \mathbf{e}_k, \qquad \text{for } \mathbf{u}_i^k \in \mathbf{U}\backslash\mathbf{B}, \tag{3.10}$$

where $\mathbf{e}_k$ is a vector with the k-th element being 1 and all the others being 0.

For the precoding vectors in $\mathbf{B}$, their construction is more complicated. We describe their construction as follows. Based on Lemma 5, we know that if $\mathbf{u}_{i_1}^{k_1} \in \mathbf{B}$, then there exists a precoding vector $\mathbf{u}_{i_2}^{k_2}$ such that $\mathbf{u}_{i_1}^{k_1} \xrightarrow{j_1} \mathbf{u}_{i_2}^{k_2}$ (i.e., $\mathbf{u}_{i_1}^{k_1} := \mathbf{H}_{j_1 i_1}^{-1} \mathbf{H}_{j_1 i_2} \mathbf{u}_{i_2}^{k_2}$). To construct $\mathbf{u}_{i_1}^{k_1}$, we first need to construct $\mathbf{u}_{i_2}^{k_2}$. If $\mathbf{u}_{i_2}^{k_2} \in \mathbf{U} \backslash \mathbf{B}$, we know that $\mathbf{u}_{i_2}^{k_2}$ has already been constructed by (3.10). Otherwise (i.e., $\mathbf{u}_{i_2}^{k_2} \in \mathbf{B}$), we construct $\mathbf{u}_{i_2}^{k_2}$ in the same way as $\mathbf{u}_{i_1}^{k_1}$, i.e., there exists a precoding vector $\mathbf{u}_{i_3}^{k_3}$ such that $\mathbf{u}_{i_2}^{k_2} \xrightarrow{j_2} \mathbf{u}_{i_3}^{k_3}$. Following the same token, we can establish a chain as follows:

$$\mathcal{C} : \mathbf{u}_{i_1}^{k_1} \xrightarrow{j_1} \mathbf{u}_{i_2}^{k_2} \xrightarrow{j_2} \cdots\cdots \xrightarrow{j_{M-2}} \mathbf{u}_{i_{M-1}}^{k_{M-1}} \xrightarrow{j_{M-1}} \mathbf{u}_{i_M}^{k_M}, \tag{3.11}$$

where $i_m \neq i_{m+1}$ for $m = 1, 2, \cdots, M-1$.

Chain $\mathcal{C}$ terminates if any of the following two cases occurs.

- *Case I:* $\mathbf{u}_{i_M}^{k_M}$ has already been constructed.
- *Case II:* $\mathbf{u}_{i_M}^{k_M}$ appears twice in chain $\mathcal{C}$.

It is easy to see that chain $\mathcal{C}$ will terminate, either by case I or case II. We now show how to construct the precoding vectors in chain $\mathcal{C}$ for the two cases, respectively.

Case I. In this case, chain $\mathcal{C}$ terminates because $\mathbf{u}_{i_M}^{k_M}$ has already been constructed. We can conclude: (i) All other precoding vectors in chain $\mathcal{C}$ have not been constructed. (ii) All precoding vectors in this chain are unique. Thus, we can construct the precoding vectors in chain $\mathcal{C}$ sequentially in the *backward* direction as follows:

$$\begin{aligned}
\mathbf{u}_{i_{M-1}}^{k_{M-1}} &:= \mathbf{H}_{j_{M-1} i_{M-1}}^{-1} \mathbf{H}_{j_{M-1} i_M} \mathbf{u}_{i_M}^{k_M} \ ; \\
\mathbf{u}_{i_{M-2}}^{k_{M-2}} &:= \mathbf{H}_{j_{M-2} i_{M-2}}^{-1} \mathbf{H}_{j_{M-2} i_{M-1}} \mathbf{u}_{i_{M-1}}^{k_{M-1}} \ ; \\
\mathbf{u}_{i_{M-3}}^{k_{M-3}} &:= \mathbf{H}_{j_{M-3} i_{M-3}}^{-1} \mathbf{H}_{j_{M-3} i_{M-2}} \mathbf{u}_{i_{M-2}}^{k_{M-2}} \ .
\end{aligned}$$

Following the same token, we construct all the precoding vectors in chain $\mathcal{C}$.

Case II. In this case, chain $\mathcal{C}$ terminates because $\mathbf{u}_{i_M}^{k_M}$ appears twice. We can conclude: (i) All precoding vectors in chain $\mathcal{C}$ have not been constructed. (ii) All precoding vectors in chain $\mathcal{C}$ are unique except $\mathbf{u}_{i_M}^{k_M}$. (iii) There exists $\hat{m}$ such that $(i_{\hat{m}}, k_{\hat{m}}) = (i_M, k_M)$ and $1 \leq \hat{m} < M$.

To construct the precoding vectors in chain $\mathcal{C}$, we divide chain $\mathcal{C}$ into two sub-chains $\mathcal{C}_1$

and $\mathcal{C}_2$:

$$\begin{aligned}
\mathcal{C}_1 &: \mathbf{u}_{i_1}^{k_1} \xrightarrow{j_1} \mathbf{u}_{i_2}^{k_2} \xrightarrow{j_2} \cdots \xrightarrow{j_{\hat{m}-2}} \mathbf{u}_{i_{\hat{m}-1}}^{k_{\hat{m}-1}} \xrightarrow{j_{\hat{m}-1}} \mathbf{u}_{i_{\hat{m}}}^{k_{\hat{m}}}\ , \\
\mathcal{C}_2 &: \mathbf{u}_{i_{\hat{m}}}^{k_{\hat{m}}} \xrightarrow{j_{\hat{m}}} \mathbf{u}_{i_{\hat{m}+1}}^{k_{\hat{m}+1}} \xrightarrow{j_{\hat{m}+1}} \cdots \xrightarrow{j_{M-2}} \mathbf{u}_{i_{M-1}}^{k_{M-1}} \xrightarrow{j_{M-1}} \mathbf{u}_{i_M}^{k_M}\ ,
\end{aligned}$$

where $(i_{\hat{m}}, k_{\hat{m}}) = (i_M, k_M)$.

For these two sub-chains, we first construct the precoding vectors in $\mathcal{C}_2$ and then construct the precoding vectors in $\mathcal{C}_1$.

Based on the relationships among the vectors in chain $\mathcal{C}_2$, we have:

$$\begin{aligned}
\mathbf{u}_{i_{\hat{m}}}^{k_{\hat{m}}} &:= \mathbf{H}_{j_{\hat{m}} i_{\hat{m}}}^{-1} \mathbf{H}_{j_{\hat{m}} i_{\hat{m}+1}} \mathbf{u}_{i_{\hat{m}+1}}^{k_{\hat{m}+1}}, \\
\mathbf{u}_{i_{\hat{m}+1}}^{k_{\hat{m}+1}} &:= \mathbf{H}_{j_{\hat{m}+1} i_{\hat{m}+1}}^{-1} \mathbf{H}_{j_{\hat{m}+1} i_{\hat{m}+2}} \mathbf{u}_{i_{\hat{m}+2}}^{k_{\hat{m}+2}}, \\
&\vdots \\
\mathbf{u}_{i_{M-2}}^{k_{M-2}} &:= \mathbf{H}_{j_{M-2} i_{M-2}}^{-1} \mathbf{H}_{j_{M-2} i_{M-1}} \mathbf{u}_{i_{M-1}}^{k_{M-1}}, \\
\mathbf{u}_{i_{M-1}}^{k_{M-1}} &:= \mathbf{H}_{j_{M-1} i_{M-1}}^{-1} \mathbf{H}_{j_{M-1} i_M} \mathbf{u}_{i_M}^{k_M}.
\end{aligned} \tag{3.12}$$

Given that $(i_{\hat{m}}, k_{\hat{m}}) = (i_M, k_M)$, we have

$$\mathbf{u}_{i_{\hat{m}}}^{k_{\hat{m}}} = \mathbf{u}_{i_M}^{k_M}. \tag{3.13}$$

(3.12) and (3.13) form a linear equation system, where $\mathbf{H}$'s are given matricies and $\mathbf{u}$'s are variables. It can be verified that a solution to $\mathbf{u}_{i_M}^{k_M}$ in the system is

$$\mathbf{u}_{i_M}^{k_M} := \text{eigvec}\left(\prod_{m=\hat{m}}^{M-1} (\mathbf{H}_{j_m i_m}^{-1} \mathbf{H}_{j_m i_{m+1}}) \right), \tag{3.14}$$

where eigvec$(\cdot)$ is an eigenvector of the square matrix. Once we obtain $\mathbf{u}_{i_M}^{k_M}$, we can sequentially construct all of the other precoding vectors in sub-chain $\mathcal{C}_2$ by (3.12).

After constructing the precoding vectors in sub-chain $\mathcal{C}_2$, we construct the precoding vectors in sub-chain $\mathcal{C}_1$. Since $\mathbf{u}_{i_{\hat{m}}}^{k_{\hat{m}}}$ has already been constructed, we can construct the other precoding vectors in sub-chain $\mathcal{C}_1$ following the same token in Case I.

It is easy to see that, in the end, all precoding vectors in $\mathbf{U}$ will be constructed.

3.5.4 Step 3: Resolving Intended Signals

We now show that the constructed precoding vectors in Step 2 satisfy (3.7) in Lemma 3. First, we present the following lemma:

Lemma 6. *The constructed precoding vectors at each transmitter are linearly independent, i.e.,* $\dim\{\mathbf{u}_i^k : 1 \leq k \leq \lambda_i\} = \lambda_i$ *for* $1 \leq i \leq N_{\mathrm{T}}$.

Proof. Consider transmitter T_i and its neighboring receivers in Fig. 3.4. Let $\mathbf{U}_i = \{\mathbf{u}_i^k : 1 \leq k \leq \lambda_i\}$ and $\mathbf{B}_i = \cup_{j\in\mathcal{I}_i}\mathbf{B}_{ij}$. Then we divide the precoding vectors in $\mathbf{U}_i$ into two groups: $\mathbf{U}_i\backslash\mathbf{B}_i$ and $\mathbf{B}_i$. Recall that in our precoding vector construction, we construct $\mathbf{u}_i^k$ by $\mathbf{u}_i^k := \mathbf{e}_k$ if $\mathbf{u}_i^k \in \mathbf{U}_i\backslash\mathbf{B}_i$ and construct $\mathbf{u}_i^k$ by $\mathbf{u}_i^k := \mathbf{H}_{ji}^{-1}\mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'}$ $(i \neq i')$ if $\mathbf{u}_i^k \in \mathbf{B}_i$. This indicates that the precoding vectors in $\mathbf{U}_i\backslash\mathbf{B}_i$ are independent of the channel matrices and the precoding vectors in $\mathbf{B}_i$ are dependent on the channel matrices. Given that the channel matrices are independent Gaussian random matrices, we have

$$\dim(\mathbf{U}_i) = \dim(\mathbf{U}_i\backslash\mathbf{B}_i) + \dim(\mathbf{B}_i) = |\mathbf{U}_i\backslash\mathbf{B}_i| + \dim(\cup_{j\in\mathcal{I}_i}\mathbf{B}_{ij}), \tag{3.15}$$

almost surely.

Now we analyze the dimension of $\cup_{j\in\mathcal{I}_i}\mathbf{B}_{ij}$. Consider two precoding vectors $\mathbf{u}_i^k \in \mathbf{B}_{ij_1}$ and $\mathbf{u}_i^{k'} \in \mathcal{B}_{ij_2}$ with $j_1 \neq j_2$. In our precoding vector construction, $\mathbf{u}_i^k$ is set to $\mathbf{u}_i^k := \mathbf{H}_{j_1i}^{-1}\mathbf{H}_{j_1i_1}\mathbf{u}_{i_1}^{k_1}$ and $\mathbf{u}_i^{k'}$ is set to $\mathbf{u}_i^{k'} := \mathbf{H}_{j_2i}^{-1}\mathbf{H}_{j_2i_2}\mathbf{u}_{i_2}^{k_2}$ for some i_1, k_1, i_2, and k_2. Hence, $\mathbf{u}_i^k$ is dependent on $\mathbf{H}_{j_1i}$ and $\mathbf{u}_i^{k'}$ is dependent on $\mathbf{H}_{j_2i}$. Given that $\mathbf{H}_{j_1i}$ and $\mathbf{H}_{j_2i}$ are two independent Gaussian random matrices, we have

$$\dim(\cup_{j\in\mathcal{I}_i}\mathbf{B}_{ij}) = \sum_{j\in\mathcal{I}_i}\dim(\mathbf{B}_{ij}), \tag{3.16}$$

almost surely.

We now analyze the dimension of $\mathbf{B}_{ij}$. Based on (3.11), each precoding vector $\mathbf{u}_i^k \in \mathbf{B}_{ij}$ is constructed in the form of

$$\mathbf{u}_i^k = \left(\prod_{m=1}^{M-1}(\mathbf{H}_{j_mi_m}^{-1}\mathbf{H}_{j_mi_{m+1}})\right)\mathbf{u}_{i_M}^{k_M},$$

where $(i_1, k_1) = (i, k)$, $M \geq 2$, and $\mathbf{u}_{i_M}^{k_M}$ is constructed either by (3.10) or (3.14). Let $\mathbf{G}_i^k = \prod_{m=1}^{M-1}(\mathbf{H}_{j_m i_m}^{-1}\mathbf{H}_{j_m i_{m+1}})$. We call $\mathbf{G}_i^k$ the "effective channel" for $\mathbf{u}_i^k$. We divide the precoding vectors in $\mathbf{B}_{ij}$ into subsets such that the precoding vectors in the same subset have the same "effective channel". Denote the subsets as $\mathbf{B}_{ij}^n$, $1 \leq n \leq N_{ij}$. Since $\mathbf{H}_{ij}$'s are independent Gaussian random matrices, any the "effective channels" are independent random matrices. Thus, we have

$$\dim(\mathbf{B}_{ij}) = \sum_{n=1}^{N_{ij}} \dim(\mathbf{B}_{ij}^n). \tag{3.17}$$

For each $\mathbf{u}_i^k \in \mathbf{B}_{ij}^n$, it is determined by its corresponding precoding vector $\mathbf{u}_{i_M}^{k_M}$ and $\mathbf{u}_{i_M}^{k_M}$ is constructed either by (3.10) or (3.14). Denote $\tilde{\mathbf{B}}_{ij}^n$ as the set of precoding vectors $\mathbf{u}_{i_M}^{k_M}$ corresponding to the precoding vectors in $\mathbf{B}_{ij}^n$. Then we have $\dim(\tilde{\mathbf{B}}_{ij}^n) = |\tilde{\mathbf{B}}_{ij}^n|$ based on three facts: (i) the precoding vectors in $\tilde{\mathbf{B}}_{ij}^n$ are at the same transmitter; (ii) the precoding vectors constructed by (3.10) are linearly independent; (iii) there are N_{A} linearly independent solutions (eigenvectors) to (3.14). Thus, we have

$$\dim(\mathbf{B}_{ij}^n) = \dim(\tilde{\mathbf{B}}_{ij}^n) = |\tilde{\mathbf{B}}_{ij}^n| = |\mathbf{B}_{ij}^n|, \tag{3.18}$$

where the first equation follows from the fact that the "effective channel" has full rank.

Based on (3.17) and (3.18), we have

$$\dim(\mathbf{B}_{ij}) = \sum_{n=1}^{N_{ij}} \dim(\mathbf{B}_{ij}^n) = \sum_{n=1}^{N_{ij}} |\mathrm{B}_{ij}^n| = |\mathrm{B}_{ij}|. \tag{3.19}$$

Based on (3.15), (3.16), and (3.19), we have

$$\begin{aligned}\dim(\mathbf{U}_i) &= |\mathbf{U}_i \backslash \mathbf{B}_i| + \dim(\cup_{j\in\mathcal{I}_i}\mathbf{B}_{ij}) = |\mathbf{U}_i \backslash \mathbf{B}_i| + \cup_{j\in\mathcal{I}_i}\dim(\mathbf{B}_{ij}) \\ &= |\mathbf{U}_i \backslash \mathbf{B}_i| + \cup_{j\in\mathcal{I}_i}|\mathbf{B}_{ij}| = \lambda_i.\end{aligned}$$

Therefore, Lemma 6 is proved. □

Denote $\mathbf{D}_j^{\mathrm{I,eff}}$ as the set of "effective" interfering stream directions at receiver R_j. Denote $\mathbf{D}_j^{\mathrm{I,algn}}$ as the set of interfering stream directions for alignment at receiver R_j. Mathematically,

we have

$$\mathbf{D}_j^{\mathrm{I,eff}} = \cup_{i\in\mathcal{I}_j}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{A}_{ij}\backslash\mathbf{B}_{ij}\},$$
$$\mathbf{D}_j^{\mathrm{I,algn}} = \cup_{i\in\mathcal{I}_j}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{B}_{ij}\}.$$

Based on the precoding vector construction procedure, we know that for each $\mathbf{H}_{ji}\mathbf{u}_i^k \in \mathbf{D}_j^{\mathrm{I,algn}}$, there exists a $\mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'} \in \mathbf{D}_j^{\mathrm{I,eff}}$ such that $\mathbf{H}_{ji}\mathbf{u}_i^k := \mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'}$. Thus we have

$$\mathrm{span}(\mathbf{D}_j^{\mathrm{I,algn}}) \subseteq \mathrm{span}(\mathbf{D}_j^{\mathrm{I,eff}}). \tag{3.20}$$

For the number of vectors in $\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I,eff}}$, we have

$$|\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I,eff}}| = \mu_j + \sum_{i\in\mathcal{I}_j}(\alpha_{ij} - \beta_{ij}) \le N_{\mathrm{A}}, \tag{3.21}$$

where the inequality follows from (3.5).

The dimension of signal and interference space at receiver R_j is:

$$\begin{aligned}
\dim(\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}) &\overset{(a)}{=} \dim(\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I,eff}}) \\
&= \dim \cup_{i\in\mathcal{I}_j} \{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{S}_{ij} \cup \mathbf{A}_{ij}\backslash\mathbf{B}_{ij}\} \\
&\overset{(b)}{=} \sum_{i\in\mathcal{I}_j} \dim\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathbf{S}_{ij} \cup \mathbf{A}_{ij}\backslash\mathbf{B}_{ij}\} \\
&\overset{(c)}{=} \sum_{i\in\mathcal{I}_j} \dim(\mathbf{S}_{ij} \cup \mathbf{A}_{ij}\backslash\mathbf{B}_{ij}) \\
&\overset{(d)}{=} \sum_{i\in\mathcal{I}_j} |\mathbf{S}_{ij} \cup \mathbf{A}_{ij}\backslash\mathbf{B}_{ij}| \\
&\overset{(e)}{=} \sum_{i\in\mathcal{I}_j} |\mathbf{S}_{ij}| + \sum_{i\in\mathcal{I}_j} |\mathbf{A}_{ij}\backslash\mathbf{B}_{ij}| \\
&= \mu_j + \sum_{i\in\mathcal{I}_j}(\alpha_{ij} - \beta_{ij}),
\end{aligned} \tag{3.22}$$

where (a) follows from (3.20). (b) follows from two facts: (i) The number of elements in $\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I,eff}}$ is bounded by N_{A} as shown in (3.21). (ii) The matrices in $\{\mathbf{H}_{ji} : i \in \mathcal{I}_j\}$ are Gaussian random matrices and are independent of each other. (c) follows from our assumption that $\mathbf{H}_{ji}$ is of full rank, which is usually the case in practical networks. (d) follows from Lemma 6. (e) follows from $\mathbf{S}_{ij} \cap \mathbf{A}_{ij} = \emptyset$ and $\mathbf{B}_{ij} \subseteq \mathbf{A}_{ij}$.

Similarly, the dimension of interference subspace at receiver R_j is:

$$\dim(\mathbf{D}_j^{\mathrm{I}}) = \dim(\mathbf{D}_j^{\mathrm{I,eff}}) = \sum_{i \in \mathcal{I}_j} (\alpha_{ij} - \beta_{ij}). \tag{3.23}$$

Based on (3.22) and (3.23), we have

$$\dim(\mathbf{D}_j^{\mathrm{S}} \cup \mathbf{D}_j^{\mathrm{I}}) = \mu_j + \dim(\mathbf{D}_j^{\mathrm{I}}), \tag{3.24}$$

which indicates that the constructed precoding vectors satisfy (3.7) in Lemma 3.

3.6 An Optimization Framework

In this section, we apply the new analytical IA model and develop a cross-layer optimization framework for a multi-hop MIMO network. Denote $\mathcal{N}$ as the set of nodes in the network with $N = |\mathcal{N}|$, each of which is equipped with N_{A} antennas. Denote $\mathcal{F}$ the set of sessions in the network with $F = |\mathcal{F}|$. Denote $r(f)$ as the data rate of session $f \in \mathcal{F}$. Denote $\mathrm{src}(f)$ and $\mathrm{dst}(f)$ as the source and the destination nodes of session $f \in \mathcal{F}$, respectively. To transport data flow f from $\mathrm{src}(f)$ to $\mathrm{dst}(f)$, we allow flow splitting inside the network for better load balancing and network resource utilization. We assume that a time frame consists of K time slots.

Half Duplex Constraints. We assume that a node cannot transmit and receive in the same time slot. Denote $x_i(t)$ as a binary variable to indicate whether node $i \in \mathcal{N}$ is a transmitter in time slot t, i.e., $x_i(t) = 1$ if node i is a transmitter in time slot t and 0 otherwise. Similarly, denote $y_i(t)$ as a binary variable to indicate whether node $i \in \mathcal{N}$ is a receiver in time slot t. Then the half duplex constraints can be written as

$$x_i(t) + y_i(t) \leq 1, \quad (1 \leq i \leq N, 1 \leq t \leq K). \tag{3.25}$$

Node Activity Constraints. Denote $z_l(t)$ as the number of data streams on link $l \in \mathcal{L}$ in time slot t. If node i is a transmitter, we have $1 \leq \sum_{l \in \mathcal{L}_i^{\mathrm{out}}} z_l(t) \leq N_{\mathrm{A}}$. Otherwise (i.e., node i is either a receiver or inactive), we have $\sum_{l \in \mathcal{L}_i^{\mathrm{out}}} z_l(t) = 0$. Combining the two cases,

we have the following constraints:

$$x_i(t) \leq \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) \leq N_{\text{A}} \cdot x_i(t), \quad (1 \leq i \leq N, 1 \leq t \leq K). \tag{3.26}$$

Similarly, by considering whether or not node i is a receiver, we have the following constraints:

$$y_j(t) \leq \sum_{l \in \mathcal{L}_j^{\text{in}}} z_l(t) \leq N_{\text{A}} \cdot y_j(t), \quad (1 \leq j \leq N, 1 \leq t \leq K). \tag{3.27}$$

General IA Constraints at a Node. In Section 3.4 and 3.5, we developed IA constraints at a transmitter and a receiver. Here, we can rewrite these constraints at a node based on the node status variables.

Based on (3.1) in our IA model, if node i is a transmitter and node j is a receiver in time slot t, then we have $\beta_{ij}(t) \leq \alpha_{ij}(t)$ for each $j \in \mathcal{I}_i$. Otherwise (i.e., $x_i(t) = 0$ or $y_j(t) = 0$), we have $\beta_{ij}(t) = 0$ and $\alpha_{ij}(t) = 0$. Combining these two cases, constraint (3.1) can be rewritten as

$$\beta_{ij}(t) \leq \alpha_{ij}(t), \quad (j \in \mathcal{I}_i, 1 \leq i \leq N, 1 \leq t \leq K). \tag{3.28}$$

Based on (3.2) in our IA model, if node i is a transmitter in time slot t, then we have $\sum_{j \in \mathcal{I}_i} \beta_{ij}(t) \leq \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t)$ as $\lambda_i = \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t)$. Otherwise (i.e., $x_i = 0$), we have $\sum_{j \in \mathcal{I}_i} \beta_{ij}(t) = 0$ and $\sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) = 0$. Combining these two cases, constraint (3.2) can be rewritten as

$$\sum_{j \in \mathcal{I}_i} \beta_{ij}(t) \leq \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t), \quad (1 \leq i \leq N, 1 \leq t \leq K). \tag{3.29}$$

Based on (3.3) in our IA model, if node i is a transmitter in time slot t, we have $\sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) \leq N_{\text{A}}$ as $\lambda_i = \sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t)$. Otherwise (i.e., $x_i = 0$), we have $\sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) = 0$. Combining these two cases, constraint (3.3) can be rewritten as

$$\sum_{l \in \mathcal{L}_i^{\text{out}}} z_l(t) \leq N_{\text{A}} \cdot x_i(t), \quad (1 \leq i \leq N, 1 \leq t \leq K). \tag{3.30}$$

Based on (3.4) in our IA model, if node j is a receiver in time slot t, we have $\beta_{ij}(t) \leq \sum_{k \in \mathcal{I}_j}^{k \neq i} [\alpha_{kj}(t) - \beta_{kj}(t)]$ for each $i \in \mathcal{I}_j$. Otherwise (i.e., $y_j = 0$), we have $\beta_{ij}(t) = 0$ and

$\alpha_{ij}(t) = 0$ for each $i \in \mathcal{I}_j$. Combining these two cases, constraint (3.4) can be rewritten as

$$\beta_{ij}(t) \leq \sum_{k \in \mathcal{I}_j}^{k \neq i} \left[\alpha_{kj}(t) - \beta_{kj}(t)\right], \qquad (i \in \mathcal{I}_j, 1 \leq j \leq N, 1 \leq t \leq K). \tag{3.31}$$

Based on (3.5) in our IA model, if node j is a receiver in time slot t, we have $\sum_{l \in \mathcal{L}_j^{\text{in}}} z_l(t) + \sum_{i \in \mathcal{I}_j} \left[\alpha_{ij}(t) - \beta_{ij}(t)\right] \leq N_{\text{A}}$. Otherwise (i.e., $y_j = 0$), we have $z_l(t) = 0$ for $l \in \mathcal{L}_j^{\text{in}}$ and $\alpha_{ij}(t) = \beta_{ij}(t) = 0$ for each $i \in \mathcal{I}_j$. Combining these two cases, constraint (3.5) can be rewritten as

$$\sum_{l \in \mathcal{L}_j^{\text{in}}} z_l(t) + \sum_{i \in \mathcal{I}_j} \left[\alpha_{ij}(t) - \beta_{ij}(t)\right] \leq N_{\text{A}} \cdot y_j(t), \qquad (1 \leq j \leq N, 1 \leq t \leq K). \tag{3.32}$$

Finally, we characterize the relationship between $\alpha_{ij}(t)$ and $z_l(t)$. If node i is a transmitter and node j is a receiver in time slot t, we have $\alpha_{ij}(t) = \sum_{l \in \mathcal{L}_i^{\text{out}}}^{\text{Rx}(l) \neq j} z_l(t)$, where $\text{Rx}(l)$ is the receiver of link l. Otherwise (i.e., $x_i(t) = 0$ or $y_j(t) = 0$), we have $\alpha_{ij}(t) = 0$. In general, we have the following constraints:

$$\alpha_{ij}(t) = y_j(t) \cdot \sum_{l \in \mathcal{L}_i^{\text{out}}}^{\text{Rx}(l) \neq j} z_l(t), \qquad (j \in \mathcal{I}_i, 1 \leq i \leq N, 1 \leq t \leq K). \tag{3.33}$$

Link Capacity Constraints. Denote $r_l(f)$ as the amount of data rate on link l that is attributed to session $f \in \mathcal{F}$. For ease of calculation, we assume that fixed modulation and coding scheme (MCS) is used for data transmission and one data stream in one time slot corresponds to one unit data rate. Then the average rate of link l over K time slots is $\frac{1}{K} \sum_{t=1}^{K} z_l(t)$. Since the aggregate data rates cannot exceed the average link rate, we have

$$\sum_{f=1}^{F} r_l(f) \leq \frac{1}{K} \sum_{t=1}^{K} z_l(t), \qquad (1 \leq l \leq L). \tag{3.34}$$

Flow Routing Constraints. At each node, flow conservation must be observed. At a source node, we have

$$\sum_{l \in \mathcal{L}_i^{\text{out}}} r_l(f) = r(f), \qquad (i = \text{src}(f), 1 \leq f \leq F). \tag{3.35}$$

At an intermediate relay node, we have

$$\sum_{l\in\mathcal{L}_i^{\text{in}}} r_l(f) = \sum_{l\in\mathcal{L}_i^{\text{out}}} r_l(f), \qquad (1 \le i \le N, i \ne \text{src}(f), i \ne \text{dst}(f), 1 \le f \le F). \tag{3.36}$$

At a destination node, we have

$$\sum_{l\in\mathcal{L}_i^{\text{in}}} r_l(f) = r(f), \qquad (i = \text{dst}(f), 1 \le f \le F). \tag{3.37}$$

It can be easily verified that if (3.35) and (3.36) are satisfied, then (3.37) is also satisfied. Therefore, it suffices to include only (3.35) and (3.36).

Collectively, constraints (3.25)–(3.36) constitute our cross-layer optimization framework with the IA model for a multi-hop MIMO network. In particular, (3.29), (3.32), and (3.33) are coupling constraints that involve both the physical layer and the link layer; (3.34) is coupling constraints that involve the physical layer, the link layer, and the network layer.

3.7 Performance Evaluation

In this section, we apply the IA optimization framework to study a throughput maximization problem in a multi-hop MIMO network. Our goals are twofold. First, we want to see how IA is performed in a network setting through a case study. Second, we want to have a quantitative comparison between the IA framework and the case when IA is not used, thereby affirming the significant advantage of exploiting IA in a network environment.

3.7.1 A Throughput Maximization Problem

We define the objective function to be maximization of the minimum rate among all sessions.[3] Denote the objective variable as $r_{\min}$. Then we have

$$r_{\min} \le r(f), \qquad 1 \le f \le F. \tag{3.38}$$

We have the following formulation:

[3]Other objective functions such as maximizing sum of weighted rates or a proportional scaling of all session rates belongs to the same category of linear function and can be solved following the same token.

OPT-IA$^{\mathbf{raw}}$	Max	$r_{\min}$
	S.t.	Half duplex constraints: (3.25);
		Node activity constraints: (3.26), (3.27);
		General IA constraints: (3.28)–(3.33);
		Link capacity constraints: (3.34);
		Flow routing constraints: (3.35), (3.36);
		Min rate constraints: (3.38).

Among the constraints, (3.33) is nonlinear and all other constraints are linear. We can employ the Reformulation-Linearization Technique [76] to linearize (3.33). By analyzing the relationship between $\alpha_{ij}(t)$ and $\sum_{l\in\mathcal{L}_i^{\text{out}}}^{\text{Rx}(l)\neq j} z_l(t)$ in (3.33), we construct two new sets of constraints (3.39) and (3.40) to replace (3.33):

$$0 \leq \sum_{l\in\mathcal{L}_i^{\text{out}}}^{\text{Rx}(l)\neq j} z_l(t) - \alpha_{ij}(t) \leq (1-y_j(t))\cdot B, \quad (j\in\mathcal{I}_i, 1\leq i\leq N, 1\leq t\leq K), \tag{3.39}$$

$$0 \leq \alpha_{ij}(t) \leq y_j(t)\cdot B, \quad (j\in\mathcal{I}_i, 1\leq i\leq N, 1\leq t\leq K), \tag{3.40}$$

where B is a fixed integer (e.g., $B = N_{\text{A}}$). It can be verified that the combination of (3.39) and (3.40) is equivalent to (3.33) in terms of maximizing $r_{\min}$.

By replacing nonlinear constraint (3.33) with (3.39) and (3.40), we have the following problem formulation:

OPT-IA	Max	$r_{\min}$
	S.t.	(3.25)–(3.32), (3.34)–(3.36), (3.38)–(3.40),

where $x_i(t)$ and $y_i(t)$ are binary variables; $z_l(t)$, $\alpha_{ij}(t)$, and $\beta_{ij}(t)$ are non-negative integer variables; $r(f)$ and $r_l(f)$ are non-negative variables; N_{A}, N, L, F, K, and B are constants.

OPT-IA is a cross-layer optimization formulation with the objective of maximizing the session throughput by fully exploiting the potential of IA at the physical layer. Recall that β_{ij} is the variable that characterizes the IA capability at the physical layer. So it suffices to disable IA capability in the network by simply setting $\beta_{ij} = 0$ for $i, j \in \mathcal{N}$. Therefore, the same network maximization problem without using IA can be formulated the following formulation:

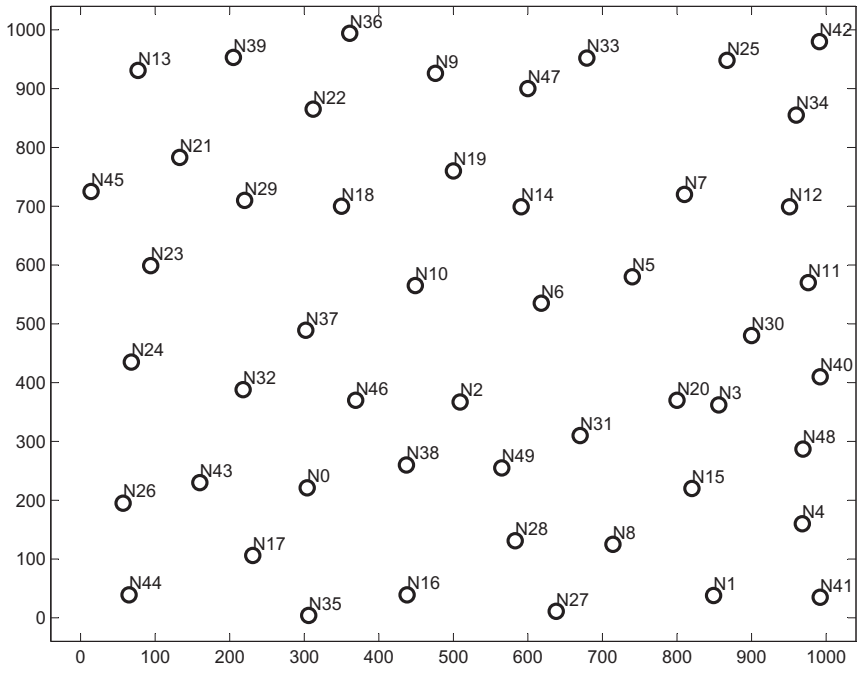


Figure 3.7: A 50-node network topology.

OPT-base	Max	$r_{\min}$
	S.t.	(3.25)–(3.32), (3.34)–(3.36), (3.38)–(3.40),
		$\beta_{ij} = 0$ for $i, j \in \mathcal{N}$.

Both OPT-IA and OPT-base are mixed integer linear program (MILP). Although the theoretical worst-case complexity of solving a general MILP problem is exponential [19, 74], there exist highly efficient optimal and approximation algorithms (e.g., branch-and-bound with cutting planes [75]) and heuristic algorithms (e.g., sequential fixing algorithm [34, 32]). For small to moderate network size, an off-the-shelf solver such as CPLEX [104] may also be effective. Since the main goal of this chapter is to present a new analytical IA model multi-hop MIMO networks (rather than developing a solution procedure), we will employ CPLEX solver for numerical results.

3.7.2 A Case Study

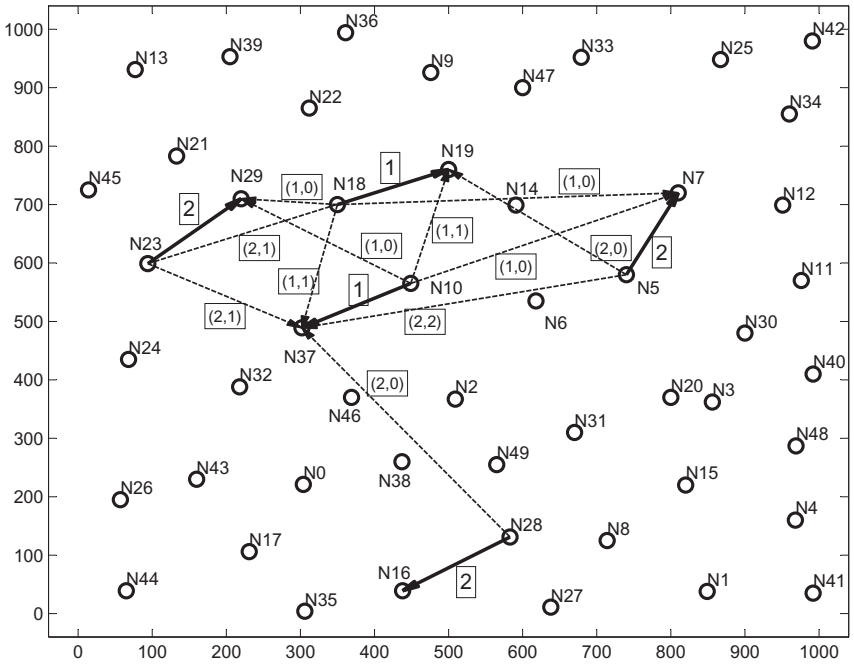
Without loss of generality, we normalize all units for distance, data rate, bandwidth, time and power with appropriate dimensions. We consider a randomly generated multi-hop MIMO network with 50 nodes (see Fig. 3.7), which are distributed in a 1000×1000 square region. Each node in the network is equipped with 4 antennas. We assume that all nodes have the same transmission range 250 and interference range 500.

In this network, there are 4 active sessions: N_{10} to N_{43}, N_{23} to N_{47}, N_{30} to N_{16}, and N_2 to N_7 in Fig. 3.7. For ease of illustration, we assume that there are 4 time slots in a time frame. By solving OPT-IA, we obtain the optimal objective (i.e., the maximum throughput) of 0.50.

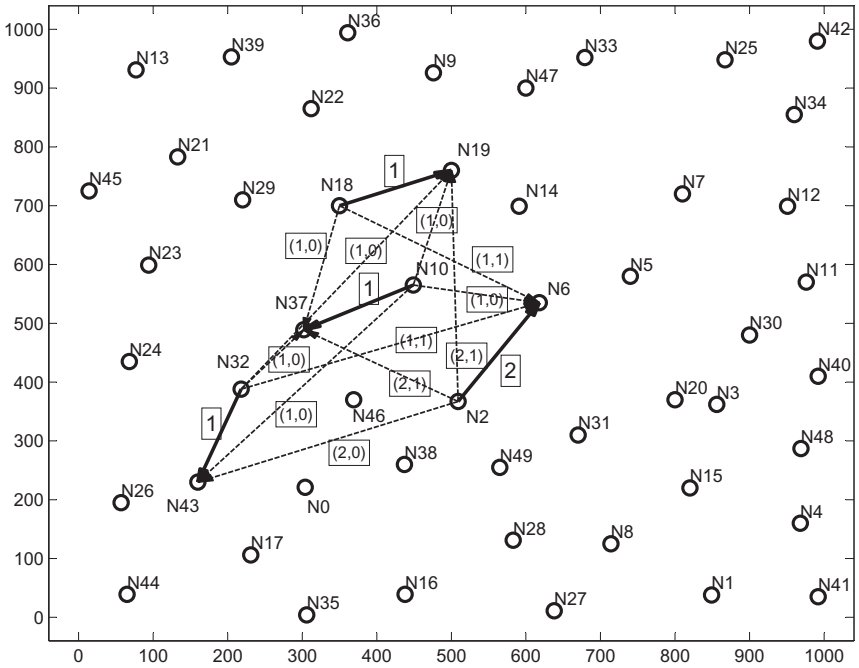
Fig. 3.8 shows the transmission/reception pattern, interference pattern, and IA scheme in each time slot. Specifically, a solid line with arrow represents a directed link (with the number of data streams on this link shown in a box, i.e., z_l). A dashed line with arrow represents interference, with the total number of interfering streams and the number of subset interfering streams chosen for IA shown in a box, i.e., $(\alpha_{ij}, \beta_{ij})$. For example, in Fig. 3.8(a), on the dashed line between N_5 and N_{37}, $(2, 2)$ in the box represents that $\alpha_{5,37} = 2$ and $\beta_{5,37} = 2$, i.e., there are two interfering streams from node N_5 to node N_{37} and both of them are selected for IA at node N_{37} in our solution.

As an example to illustrate how IA is performed in the network, let's take a look at N_{37} in time slot 1 (see Fig. 3.8(a)). At node N_{37}, there is a total of 7 interfering streams (from transmitting nodes N_5, N_{18}, N_{23}, and N_{28}). In our solution, we find that for the 2 interfering streams from node N_5, both of them are aligned to the interfering streams from node N_{28}. Similarly, for the interfering stream from node N_{18}, it has been aligned to an interfering stream from node N_{28}. For the 2 interfering streams from node N_{23}, one of them has been aligned to the interfering streams from node N_{28}. That is, for the 7 interfering streams at node N_{37}, 4 of them have been successfully aligned to the remaining 3 interfering streams. As a result, node N_{37} only needs to consume 3 DoFs to cancel the 7 interfering streams.

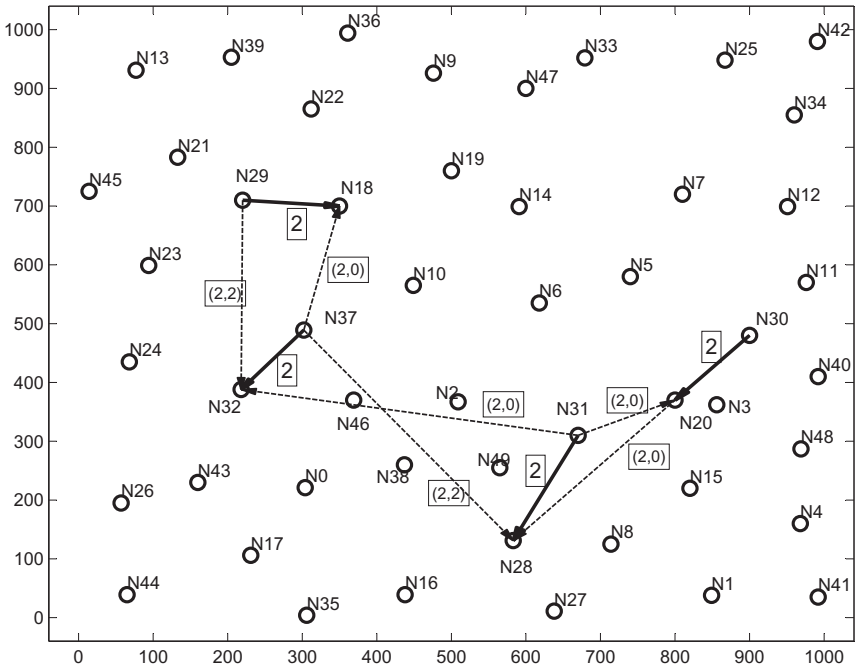
Table 3.2 summarizes the savings of DoFs in IC due to IA at each receiving node in



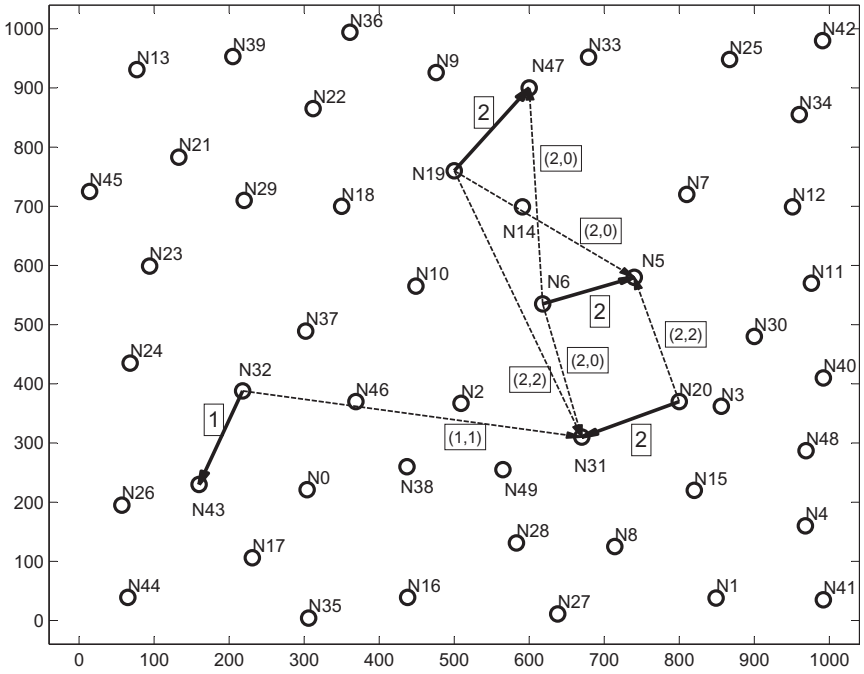
(a) Time slot 1



(b) Time slot 2



(c) Time slot 3



(d) Time slot 4

Figure 3.8: Transmission and interference pattern in each time slot.

Table 3.2: A comparison between $P(N_j)$ and $Q(N_j)$.

Time slot 1			Time slot 2		
Rx	$P(Rx)$	$Q(Rx)$	Rx	$P(Rx)$	$Q(Rx)$
N_7	2	2	N_6	3	1
N_{16}	0	0	N_{19}	4	3
N_{19}	5	3	N_{37}	4	3
N_{29}	2	2	N_{43}	3	3
N_{37}	7	3			
Time slot 3			Time slot 4		
Rx	$P(Rx)$	$Q(Rx)$	Rx	$P(Rx)$	$Q(Rx)$
N_{18}	2	2	N_5	4	2
N_{20}	2	2	N_{31}	5	2
N_{28}	4	2	N_{43}	0	0
N_{32}	4	2	N_{47}	2	2

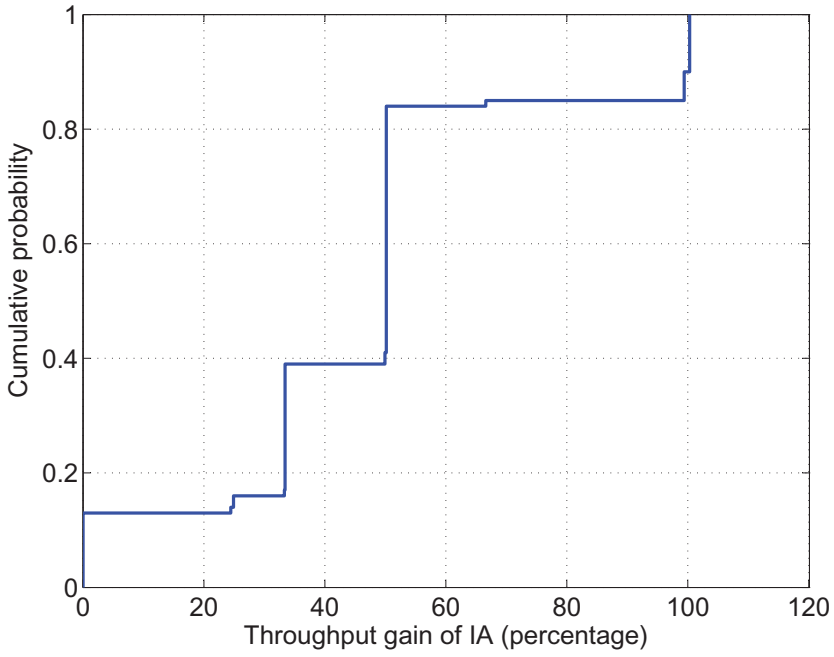


Figure 3.9: The CDF of throughput gain of IA in multi-hop MIMO networks.

each time slot. To abbreviate notation in the table, denote $P(N_j)$ as the total number of interfering streams at node N_j, i.e., $P(N_j) = \sum_{i \in \mathcal{I}_j} \alpha_{ij}$. Denote $Q(N_j)$ as the total number of DoFs that are consumed by node N_j for IC, i.e., $Q(N_j) = \sum_{i \in \mathcal{I}_j} (\alpha_{ij} - \beta_{ij})$. Then the difference between $P(N_j)$ and $Q(N_j)$ is the saving in DoFs at node N_j due to IA. Note that savings in DoFs are directly translated into improvement of network throughput.

To compare to the case when IA is not applied in the network, we formulate a throughput maximization problem and denote it as OPT-base. By solving OPT-base with CPLEX, we find that the objective is only 0.25 (comparing to 0.50 under OPT-IA).

3.7.3 Complete Results

The case study discussed in the last section gives results from one 50-node network instance. In this section, we perform the same drill for 100 network instances. Here, a time frame has 6 time slots. Figure 3.9 presents the CDF of the throughput gain of IA (i.e., the optimal objective value of OPT-IA over that of OPT-base). Based on our simulation results, the

average throughput improvement due to the use of IA (over 100 instances) is 48%.

3.8 Related Work

The concept of IA was coined in a seminar paper by Jafar and Shamai for the two-user X channel [39]. Since then, results for IA have been developed for a variety of channels and networks in increasingly sophisticated forms, such as the K-user interference channel [9], the cellular network [84, 85], the MIMO Y channel [48], ergodic capacity in fading channel [65], the X network with arbitrary number of users, and the complex interference channel. A distributed IA scheme was proposed by Gomadam et al. in [24]. The feasibility of IA in signal vector space for K-user MIMO interference channel was studied by Yetis et al. in [96], and blind IA (no CSI at transmitter) was studied in [92]. A tutorial on IA from information theory perspective is [40].

In wireless communications and networking communities, efforts on IA have been mainly limited to validations on small toy networks [15, 23, 53]. In [15], El Ayach et al. did an experimental study of IA in MIMO-OFDM interference channels and showed that IA achieves the theoretical throughput gains. In [23], Gollakotta et al. demonstrated that the combination of IA and IC increases the average throughput by 1.5 times on the downlink and 2 times on the uplink in a 2×2 MIMO WLAN. In [53], Lin et al. proposed a distributed random access protocol (called 802.11n$^+$) based on IA and demonstrated that the system can double the average network throughput in a small network with three pairs of nodes. None of these prior efforts have made advances to extend IA technique in a network setting as we have done in this chapter.

3.9 Chapter Summary

The goal of this chapter is to make a concrete step forward in advancing IA technique in multi-hop MIMO networks. We developed an analytical IA model consisting of a set of constraints at a transmitter and a receiver. We also proved the feasibility of the IA model

by showing that each DoF vector satisfying the constraints in the IA model is feasible at the physical layer. Based on this IA model, we developed a cross-layer optimization framework for IA in a multi-hop MIMO network. We anticipate that this framework or its variants will be adopted by the networking community to study IA in a multi-hop network environment.

As an application of this optimization framework, we studied a network throughput optimization problem and compared performance objectives with our IA model and that without IA. Simulation results showed that the use of IA in a multi-hop MIMO network can significantly reduce DoF consumption for IC at the receivers, thereby improving network throughput.

Chapter 4

Spatial Interference Alignment for Multicast Communications

4.1 Introduction

Interference alignment (IA) is a promising interference management technique in wireless networks that may help increase throughput dramatically. Since its inception, IA has received much attention in the information theory (IT) community and has been widely applied to a wide range of channels and networks (see, e.g., [9, 15, 23, 39, 84]). The most significant theoretical result of IA so far was that developed by Cadambe and Jafar in [9], which showed that the K-user interference channel could achieve $K/2$ degrees of freedom (DoFs). This suggests that the total DoFs of the interference channel can increase linearly with the number of users. In addition to many theoretical advances, there are also active efforts that are devoted to the feasibility and performance of IA in practical networks. For instance, Gollakotta et al. in [23] demonstrated that IA was feasible for WLAN and could increase the network throughput significantly. El Ayach et al. in [15] experimentally showed that IA could achieve its theoretical throughput gain in MIMO-OFDM system.

To date, the benefits of IA has been successfully demonstrated from both theoretical and practical perspectives. A natural question to ask is in what environment the benefits of IA

will be most profound. Based on our observation and understanding, we believe that IA is most effective when there are a large number of interfering links in the network (so that there are more opportunities for the interferences to be aligned). One of such environments that we are particularly interested in is *multicast communications,* where there are multiple concurrent multicast groups, with each group consisting of many destination nodes. A large number of interfering links are a common place in this environment, due to simultaneous activation of many links. In addition to being a good candidate for IA, multicast is also a general form of communications, as it encompasses both unicast and broadcast as its special cases. These reasons motivate us to study IA for multicast communications.

Given that IA in the spatial domain (using MIMO) is considered most practical, we will focus on spatial-domain IA in MIMO networks. Within this universe, although there are a large volume of research efforts on spatial-domain IA in the literature, there is no study so far on multicast IA in multi-hop MIMO networks (see Section 4.7). This is not surprising, as there are a number of technical barriers to achieve multicast IA in a multi-hop MIMO network, which we describe as follows. First, ensuring feasibility of IA for multicast at the physical layer is not a trivial problem. By feasibility at the physical layer, we mean that there exist precoding and decoding vectors for every data stream in the network so that it can be transported successfully. Since the construction of precoding and decoding vectors requires complex matrix manipulations, it is not an easy job to ensure feasibility, especially in a multi-hop network environment. Second, maintaining tractability (i.e., being analyzable with low complexity) on top of feasibility adds another level of challenge. Design of precoding and decoding vectors for each data stream would require complex matrix manipulations that soon becomes intractable. For networking research, what we need is a simple mathematical abstraction of IA property without being involved in tedious matrix manipulations. This would call for a simple model that is also provable to be implementable at the physical layer (i.e., feasible). So far such a simple model does not exist for multicast IA and needs to be developed. Last but not least, the optimal decision for IA in multicast is always coupled with MIMO's interference cancellation (IC) and spatial multiplexing (SM), as well as time slot

scheduling in the network. Such a cross-layer design is attractive from optimality perspective as it typically offers much better performance than a decoupled design. But it also comes with much complexity in problem formulation and solution procedure.

To address these challenges, we classify interferences at a receive node into "strong" interferences and "weak" interferences. The strong interferences are handled by MIMO's IA and IC capabilities as well as time slot scheduling, while the weak interferences are simply treated as noise at each receiver in its SINR calculation. Based on this classification, we make the following contributions in this chapter:

- Instead of constructing precoding and decoding vectors in the network, we develop a set of multicast IA constraints at both transmitter and receiver sides. These constraints characterize a feasible design space for multicast IA. Based on the feasible design space, we develop a set of DoF constraints for each multicast link to characterize its number of achievable data streams. We show that as long as these constraints are satisfied, then there exist precoding and decoding vectors at the physical layer to guarantee feasibility at the physical layer, i.e., data streams on each multicast link are transported free of strong interference. Moreover, the multicast IA and DoF constraints only requires addition and subtraction algebraic operations.
- Based on multicast IA and DoF constraints, we develop a set of cross-layer constraints to characterize the interaction of IA, IC, SM, and time slot scheduling. We formulate a multicast throughput maximization problem to study the impact of IA. To handle nonlinear constraints in the formulation, we propose to linearize the nonlinear constraints through reformulation and approximation. We show that our reformulation and approximation can achieve a $(1-\varepsilon)$-optimality of the original problem.
- We conduct simulation to evaluate the throughput gain of IA by comparing the multicast throughput with IA against that without IA. Simulation results show that the use of IA can significantly increase multicast throughput. Furthermore, the throughput gain of IA increases with the size of multicast group and the number of multicast sessions, echoing the motivating conjecture that led us to this work.

The remainder of this chapter is organized as follows. In Section 4.7, we review related work. In Section 4.2, we offer a motivating example to show the benefits of IA for multicast communications. In Section 4.3, we show how to find a feasible design space for multicast IA through a set of linear constraints without getting into the details of precoding/decoding vector design. Section 4.4 formulates a multicast throughput maximization problem to study the impact of IA. Section 4.5 shows how to linearizes the nonlinear constraints in the optimization problem. Section 4.6 presents our simulation results. Section 4.8 concludes the chapter. Table 4.1 lists the notation used in this chapter.

4.2 A Motivating Example

IA is a powerful techniques for interference management. In the spatial domain (MIMO), it allows multiple interfering signals from different transmitters to overlap in the same direction at a receiver so that they can be canceled collectively by just one DoF. This idea has been demonstrated successfully in a number of prior works (see, e.g., [9, 15, 23, 84]). We observe that IA is most effective when (i) the number of interfering signals at a receiver is significant, and (ii) the interfering signals can be aligned into some common directions. A natural question to ask is in what environment these two conditions are most likely to occur. One answer we can think of immediately is *multicast communications,* where there are multiple concurrent multicast groups, with each group consisting of many users. In what follows, we use an example to illustrate IA in multicast communications. The positive impact from the number of multicast sessions and multicast group size on IA will be demonstrated by numerical results in Section 4.6.

Consider the network in Fig. 4.1, where each node has 2 antennas (DoFs). We use solid arrow line to represent an intended transmission and a dashed arrow line to represents interference in all figures throughout the chapter. Here, we have three multicast groups: from source N_1 to destination group $\{N_4, N_5\}$, from N_2 to $\{N_4, N_6, N_8\}$, and N_3 to $\{N_7, N_8\}$, respectively. For each multicast group, one data stream is sent from the source to its destination nodes in the group. We assume that the interference from N_1 to N_8 and from N_3

Table 4.1: Notation for spatial IA in multicast communications.

Constant symbols	
B	A large enough constant integer
D	The strong interference range
F	The number of sessions in the network
L	The number of links in the network
N	The number of nodes in the network
A	The number of antennas at a node
T	The number of time slots in a frame
W	The network bandwidth
P_{s}	The transmit power of an active transmitter
P_{n}	The noise power at a receiver
$\mathcal{P}_j$	The set of nodes that may strongly interfere with node j
$\mathcal{Q}_i$	The set of nodes that may be strongly interfered by node i
Optimization variables	
$r(f)$	The data rate of session f
$c_i(t)$	The achievable outgoing rate of node i in time slot t
$c_i^k(t)$	The achievable rate of stream k at node i in time slot t
$p_i^k(t)$	The transmit power for the kth stream at node i in time slot t
$x_i(t)$	The activity of node i in time slot t
$z_i(t)$	The number of outgoing data streams at node i in time slot t
$\lambda_i^k(t)$	The activity of the kth stream at node i in time slot t
$\alpha_{ik}(t)$	The number of interfering streams from node i that to be aligned at node k in time slot t
$\gamma_{ij}^k(t)$	The effective SINR at node j for receiving the kth stream from node i in time slot t

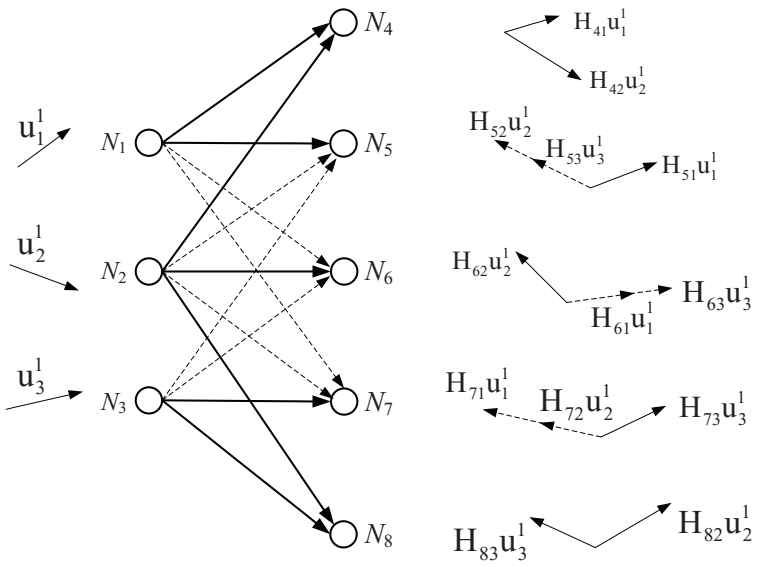


Figure 4.1: An example that illustrates IA for multicast.

to N_4 is negligible due to the large distance (beyond interference range). In this network, nodes N_5, N_6, and N_7 receive one desired data stream and two interfering streams from the three transmitters. Without using IA, the desired data stream at N_5, N_6, and N_7 cannot be decoded free of interference, due to the presence of two independent interfering streams.

To show how IA makes it possible to send three data streams in the three multicast groups, we design the precoding vectors for the three transmitters (i.e., N_1, N_2, and N_3) as follows:

$$\begin{aligned} \mathbf{u}_1^1 &:= \text{eigvec}(\mathbf{H}_{61}^{-1}\mathbf{H}_{63}\mathbf{H}_{53}^{-1}\mathbf{H}_{52}\mathbf{H}_{72}^{-1}\mathbf{H}_{71}) \ , \\ \mathbf{u}_2^1 &:= \mathbf{H}_{72}^{-1}\mathbf{H}_{71}\mathbf{u}_1^1 \ , \\ \mathbf{u}_3^1 &:= \mathbf{H}_{63}^{-1}\mathbf{H}_{61}\mathbf{u}_1^1 \ , \end{aligned} \tag{4.1}$$

where $\mathbf{u}_i^k$ is the precoding vector of the kth transmitted data stream at node N_i ($i = 1, 2, 3$); $\mathbf{H}_{ji}$ is the channel matrix between node j and node i, which is assumed to have full rank throughout the chapter; eigvec($\mathbf{H}$) is an eigenvector of square matrix $\mathbf{H}$; and operation ":=" means that two nonzero vectors are in the same direction, that is, $\mathbf{u}_1 := \mathbf{u}_2$ if and only if there exists a nonzero complex number a such that $\mathbf{u}_1 = a\mathbf{u}_2$.

By using the above precoding vectors at transmitters N_1, N_2, and N_3, we have the fol-

lowing observations on the interference at receivers N_5, N_6, and N_7. At receiver N_5, the interference from N_2 is in the direction of $\mathbf{H}_{52}\mathbf{u}_2^1$ while the interference from N_3 is in the direction of $\mathbf{H}_{53}\mathbf{u}_3^1$. It is easy to check that these two interferences now align in the same direction (i.e., $\mathbf{H}_{52}\mathbf{u}_2^1 := \mathbf{H}_{53}\mathbf{u}_3^1$), due to the special design of precoding vectors in (4.1). Similarly, at receiver N_6, based on (4.1), one can verify that the interferences from N_1 and N_3 are aligned in the same direction, i.e., $\mathbf{H}_{61}\mathbf{u}_1^1 := \mathbf{H}_{63}\mathbf{u}_3^1$. Finally, at receiver N_7, one can verify that $\mathbf{H}_{72}\mathbf{u}_2^1 := \mathbf{H}_{71}\mathbf{u}_1^1$, indicating that the two interfering streams from N_1 and N_2 are aligned in the same direction, again due to the special design of the precoding vectors in (4.1). In summary, the IA scheme resulting from the precoding vectors in (4.1) is shown in the figure, where the two interfering streams at receive nodes N_5, N_6, and N_7 is successfully aligned to the same direction.

We now show that by using the precoding vectors in (4.1) at the transmitters, the desired data streams can be decoded free of interference at each receiver (i.e., N_4, N_5, N_6, N_7, and N_8). Denote $\mathbf{v}_j^l$ as the decoding vector of the lth data stream at receive node j, $l = 1, 2$ and $j = 4, 5, \cdots, 8$. In particular, receivers N_4 and N_8 have 2 decoding vectors since they have 2 desired data streams. To decode the desired data streams at each receiver, one set of decoding vectors are as follows:

$$\begin{aligned}
(\mathbf{v}_4^1)^T &:= [1\ 0][\mathbf{H}_{41}\mathbf{u}_1^1\ \mathbf{H}_{42}\mathbf{u}_2^1]^{-1}, \\
(\mathbf{v}_4^2)^T &:= [0\ 1][\mathbf{H}_{41}\mathbf{u}_1^1\ \mathbf{H}_{42}\mathbf{u}_2^1]^{-1}, \\
(\mathbf{v}_5^1)^T &:= [1\ 0][\mathbf{H}_{51}\mathbf{u}_1^1\ \mathbf{H}_{52}\mathbf{u}_2^1]^{-1}, \\
(\mathbf{v}_6^1)^T &:= [0\ 1][\mathbf{H}_{61}\mathbf{u}_1^1\ \mathbf{H}_{62}\mathbf{u}_2^1]^{-1}, \\
(\mathbf{v}_7^1)^T &:= [0\ 1][\mathbf{H}_{71}\mathbf{u}_1^1\ \mathbf{H}_{73}\mathbf{u}_3^1]^{-1}, \\
(\mathbf{v}_8^1)^T &:= [1\ 0][\mathbf{H}_{81}\mathbf{u}_2^1\ \mathbf{H}_{83}\mathbf{u}_3^1]^{-1}, \\
(\mathbf{v}_8^2)^T &:= [0\ 1][\mathbf{H}_{81}\mathbf{u}_2^1\ \mathbf{H}_{83}\mathbf{u}_3^1]^{-1}.
\end{aligned} \tag{4.2}$$

One can verify that the precoding vectors in (4.1) and the decoding vectors in (4.2) satisfy the zero-forcing IC requirements:

$$(\mathbf{v}_j^l)^T\mathbf{H}_{ji}\mathbf{u}_i^k = \begin{cases} 1 & \text{if } \mathbf{v}_j^l \text{ is designed for the data stream of } \mathbf{u}_i^k, \\ 0 & \text{otherwise.} \end{cases}$$

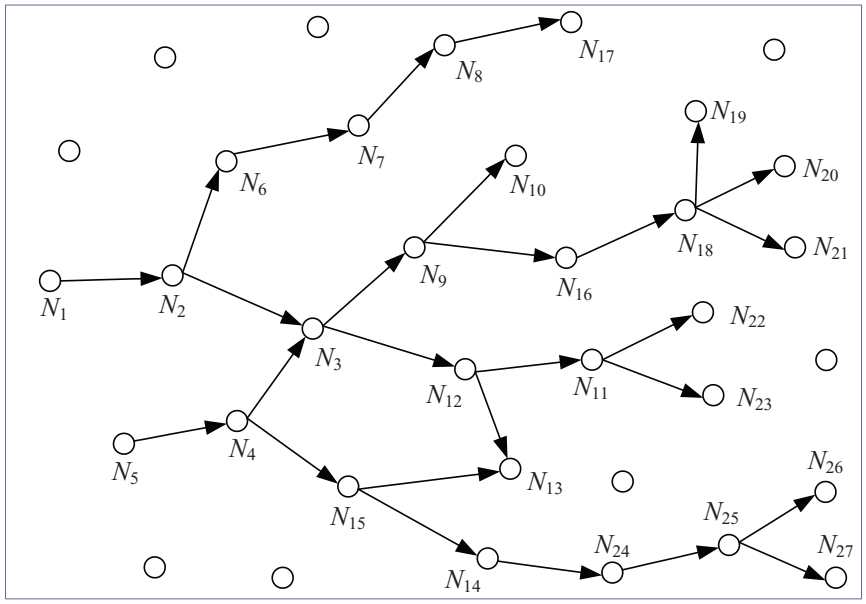
Therefore, by using IA, three data streams can be sent to the three multicast groups successfully.

4.3 Multicast IA: Characterizing Feasible Design Space

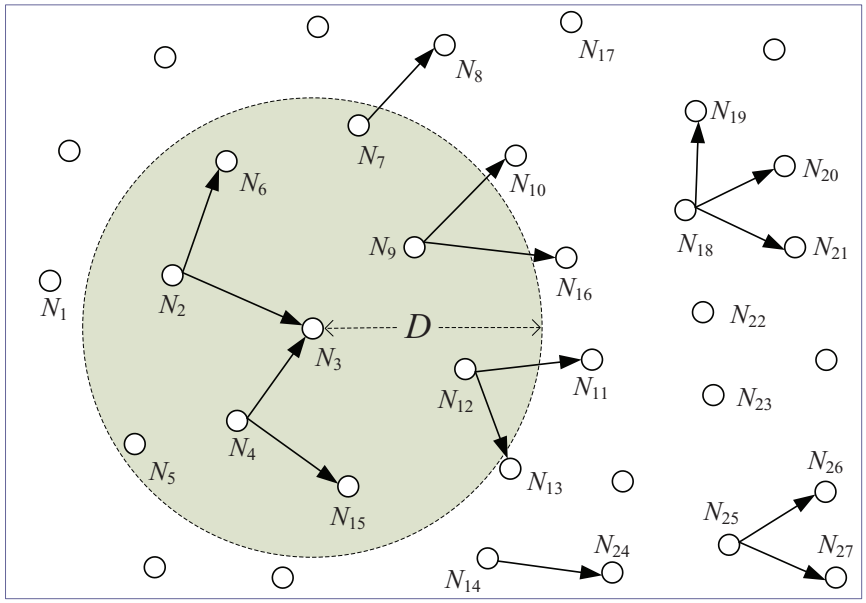
In the previous section, we demonstrated that a special design of precoding vectors at the transmitters could align the interference from different transmitters to the same direction at each receiver, but we did not explain where these precoding vectors came from. In fact, the design of these precoding vectors requires complex matrix manipulations at the physical layer and is not an easy task. The good news is that, for networking research, it is not necessary to design these precoding vectors in order to ensure feasibility at the physical layer. In this section, we will show that the physical-layer feasible design space for IA can be characterized by a set of simple constraints, with only "+" and "−" algebraic manipulations. As long as the desired IA falls within the space that is defined by these constraints, such IA is guaranteed to be feasible at the physical layer (with a corresponding set of feasible precoding vectors).

Note that we have been limiting our discussion of the design space to be feasible, without mentioning "optimal." This is because an optimal IA design space for a general network topology remains unknown in the information theory community [16]. Therefore, pursuing an optimal design space for multicast IA appears futile and a more reasonable goal at present is to pursue a feasible design space, as we are doing in this section.

Consider a multi-hop multicast network in Fig. 4.2(a), where each node is equipped with A antennas. Among the nodes there are a set of multicast sessions, where each session's multicast tree is computed through some multicast routing protocol (e.g., MOSPF [62]). Denote $\mathcal{N}$ as the set of nodes associated with the multicast routing trees, with N being its cardinality (i.e., $N = |\mathcal{N}|$). Assume transmission scheduling is done within a time frame that consists of T time slots. Within a time slot t, only a subset of nodes may be active due to half-duplex constraints and interference constraints. Among the active nodes, a transmit node may send its data information to multiple receive nodes (in one-hop multicast branch)



(a) Routing trees of two multicast sessions



(b) Active links in a time slot

Figure 4.2: An example of multicast communications in a multi-hop network.

simultaneously, as shown in Fig. 4.2(b).

Strong Interference vs. Weak Interference. Referring to Fig. 4.2(b), for a receiver in a time slot (see, e.g., N_3), it is being interfered with by all its unintended transmit nodes in the network. Interferences from these different nodes have different strength at receive node N_3. Those interfering nodes that are closer to node N_3 will have stronger interference on N_3 than those that are far away. To distinguish this difference in interference strength, we classify the interferences at a receiver into two groups: *strong interferences* and *weak interferences.* For strong interferences, we will (zero-forcing) nullify them by MIMO's IA and IC capabilities. For weak interferences, we will simply treat them as noise in the SINR calculation.

The next question is: how do we classify an interference as strong or weak interference with respect to a receive node? A natural approach here is to employ an interference range, denoted as D. That is, if the interfering transmitter is within a radius of D from the receiver, then the interference is considered as a strong interference. Otherwise, the interference is considered as a weak interference. For example, for receive node N_3 in Fig. 4.2(b), the interferences from the transmitters within the shadowed area (e.g., N_7, N_9, and N_{12}) are considered as strong interferences while the interferences from the transmitters outside the disk (e.g., N_{14}, N_{18}, and N_{25}) are considered as weak interferences.

Obviously, the setting of interference range D is important. A similar problem has been explored by Shi et al. in [79], which gave a guideline on the optimal setting of interference range. In Section 4.6, we will also explore the optimal setting of D for multicast IA via numerical results.

Multicast IA Constraints: Transmitter Side. Consider a transmit node $i \in \mathcal{N}$ in time slot t in Fig. 4.3. Denote $z_i(t)$ as the number of outgoing data streams from transmit node i. Denote $\mathcal{R}_i$ as the set of node i's immediate next-hop nodes along i's downstream multicast tree(s). For example, for node N_2 in Fig. 4.2(a), we have $\mathcal{R}_2 = \{N_3, N_6\}$. For the receive nodes in $\mathcal{R}_i$, we should ensure that all of them can successfully receive the data information from transmit node i.[1] Based on the interference range D, denote $\mathcal{Q}_i$ as the set

[1]We note that multicast communication differs from multi-user MIMO communication. In multicast

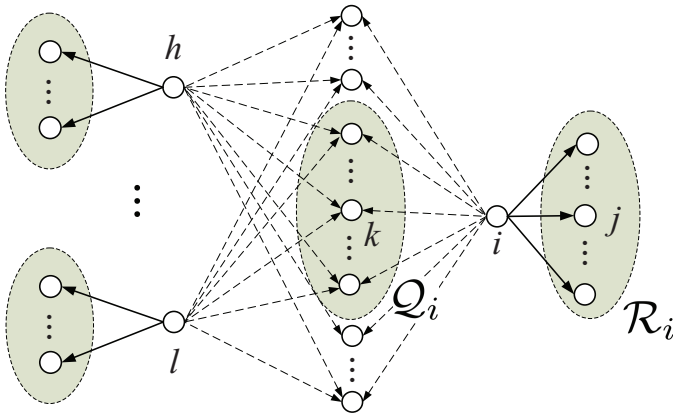


Figure 4.3: An illustration of multicast IA constraints at transmit node i.

of nodes that are strongly interfered with by transmit node i (see Fig. 4.3). Receive nodes outside $\mathcal{Q}_i$ are therefore considered to be weakly interfered by transmit node i.

We now consider a receive node $k \in \mathcal{Q}_i$ in Fig. 4.3. It is interfered with by $z_i(t)$ streams from transmit node i. It may be also interfered by the interfering streams from some other unintended transmitters (h and l), as shown in Fig. 4.3. As demonstrated in Section 4.2, by constructing the precoding vectors at transmit node i, we can align some (or all) of these $z_i(t)$ interfering streams to other interfering streams (e.g., from h and l) at receive node k. Among these $z_i(t)$ interfering streams, denote $\alpha_{ik}(t)$ as the number of interfering streams that can be successfully aligned to other interfering streams (e.g., from h and l) at receive node k. Then, at receive node k, the number of "effective" interfering streams from transmit node i is reduced from $z_i(t)$ to $[z_i(t) - \alpha_{ik}(t)]$. Note that variable $\alpha_{ik}(t)$ characterizes the IA space for the $z_i(t)$ interfering streams from transmit node i to receive node k. A larger value of $\alpha_{ik}(t)$ means that more interfering streams from transmit node i can be successfully aligned to other interfering streams at receive node k. Next, we derive constraints for α_{ik} to characterize the IA design space at transmit node i.

Referring to Fig. 4.3, let's consider one of the outgoing stream at transmit node i and its

communication, every receive node in $\mathcal{R}_i$ will receive the same (all) outgoing data streams from transmit node i. In contrast, in multi-user MIMO, each node in $\mathcal{R}_i$ will receive different subset of outgoing data streams from node i and consider the rest as interference.

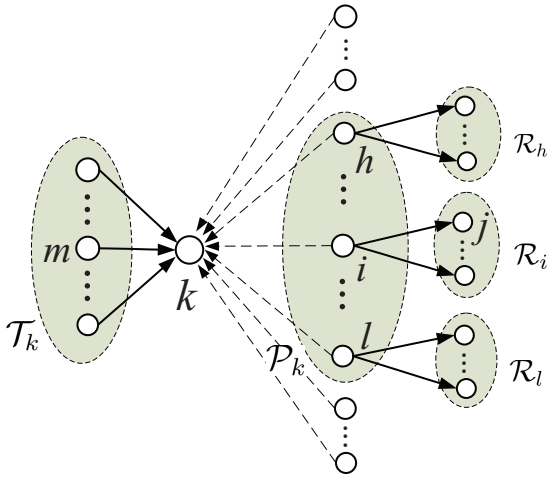


Figure 4.4: An illustration of multicast IA constraints at receive node k.

precoding vector, say $\mathbf{u}_i^1$. This outgoing stream will strongly interfere with all the receive nodes in $\mathcal{Q}_i$. At receive node $k \in \mathcal{Q}_i$, the direction of this interfering stream is $\mathbf{H}_{ki}\mathbf{u}_i^1$. By properly constructing $\mathbf{u}_i^1$, we can align this interfering stream to any arbitrary direction at receive node k. Suppose that $\mathbf{u}_i^1$ is constructed to make this interfering stream be aligned to a particular direction at receive node k. Then, at the receive nodes in $\mathcal{Q}_i \backslash \{k\}$, the direction of this interfering stream is also fixed and has no freedom for alignment. This indicates that the construction of a precoding vector can only guarantee one of its corresponding interfering streams to be successfully aligned (at one receive node). Since there are $z_i(t)$ precoding vectors at transmit node i, we can align $z_i(t)$ interfering streams to any particular directions among the receive nodes in $\mathcal{Q}_i$. That is, the total number of "alignable" interfering streams from transmit node i (i.e., $\sum_{k\in\mathcal{Q}_i} \alpha_{ik}(t)$) cannot exceed the number of the precoding vectors (i.e., $z_i(t)$). Then we have

$$\sum_{k\in\mathcal{Q}_i} \alpha_{ik}(t) \leq z_i(t), \quad 1 \leq i \leq N, 1 \leq t \leq T. \tag{4.3}$$

Multicast IA Constraints: Receiver Side. Consider a receive node, say k, in Fig. 4.4. Denote $\mathcal{P}_k$ as the set of nodes that may strongly interfere with receive node k (e.g., nodes i, h, and l in Fig. 4.4). Transmit nodes outside $\mathcal{P}_k$ are considered to be weak interferer and their interferences will be treated as noise in the SINR calculation at receive node k. Among

the $z_i(t)$ interfering streams from transmit node $i \in \mathcal{P}_k$, constraint (4.3) ensures that there are α_{ik} interfering streams that can be aligned to any particular directions at receive node k. However, to ensure the resolvability of the data streams at the receive nodes in $\mathcal{R}_i$, any two interfering streams from transmit node i cannot be aligned to the same direction. The following lemma offers a sufficient condition to ensure this will not happen:

Lemma 7. *For a receive node k, if*

$$\alpha_{ik}(t) \leq \sum_{h \in \mathcal{P}_k}^{h \neq i} [z_h(t) - \alpha_{hk}(t)], \qquad 1 \leq k \leq N, i \in \mathcal{P}_k, 1 \leq t \leq T \ , \tag{4.4}$$

then there exists an IA scheme in which any two interfering streams from transmit node $i \in \mathcal{P}_k$ are not being aligned to the same direction at receive node k.

Proof. We show that the interfering streams from transmit node i can be aligned successfully at receive node k by construction. For the interfering streams from other transmit nodes in $\mathcal{P}_k$ (e.g., h and l in Fig. 4.4), they can be successfully aligned in the same way as transmit node i.

At receive node k, denote $\mathcal{Z}_i$ as the set of its interfering streams from transmit node i. Among the interfering streams in $\mathcal{Z}_i$, denote $\mathcal{A}_{ik}$ as the subset of interfering streams that should be aligned (to other interfering streams) at receive node k. By definition, we have $|\mathcal{Z}_i| = z_i(t)$ and $|\mathcal{A}_{ik}| = \alpha_{ik}(t)$. To align the interfering streams in $\mathcal{A}_{ik}$ at receive node k, we employ the following IA scheme: for each interfering stream in $\mathcal{A}_{ik}$, we align it to a *unique* interfering stream in $\cup_{h \in \mathcal{P}_k}^{h \neq i} \mathcal{Z}_h \backslash \mathcal{A}_{hk}$. By "unique" we mean that any two interfering streams in $\mathcal{A}_{ik}$ are not allowed to align to the same (a third) interfering stream at receive node k. Based on the given constraint (4.4), we know that the interfering streams in $\cup_{h \in \mathcal{P}_k}^{h \neq i} \mathcal{Z}_h \backslash \mathcal{A}_{hk}$ are more than that in $\mathcal{A}_{ik}$. So every interfering stream in $\mathcal{A}_{kj}$ can be successfully aligned in this IA scheme. Furthermore, it is not difficult to see that this IA scheme meet the requirement. Therefore, this lemma holds. □

In (4.4), the left hand side (LHS) is the number of "alignable" interfering streams from transmit node i and the right hand side (RHS) is the number of directions occupied by the

interference from the transmit nodes in $\mathcal{P}_k$ other than transmit node i. Lemma 7 says that, at receive node k, if the "alignable" interfering streams from transmit node i are no more than the directions occupied by the interference from the transmit nodes in $\mathcal{P}_k \backslash \{i\}$, then every two interfering streams from transmit node $i \in \mathcal{P}_k$ will not be aligned to the same direction.

Total DoF Constraints. Constraints (4.3) and (4.4) characterize how many interfering streams can be successfully aligned for each interfering node pair (i, k). Based on (4.3) and (4.4), we derive the constraints to characterize how many data streams at each transmitter can be successfully transported to its receiver(s). In a time slot t, a node can be a transmitter, a receiver, or idle. For a node $k \in \mathcal{N}$, we define a binary variable $y_k(t)$ to indicate whether or not node k is a receiver in time slot t. Specifically, $y_k(t) = 1$ if node k is a receiver in time slot t and 0 otherwise. If node k is a receiver, then it may have multiple incoming links (e.g., N_3 in Fig. 4.2(a) and (b)) as it may be traversed by multiple multicast trees. Denote $\mathcal{T}_k$ as the set of node k's one-hop upstream nodes from all the multicast trees that contain k. For example, for N_3 in Fig. 4.2(a), we have $\mathcal{T}_3 = \{N_2, N_4\}$. Then we have the following analysis on the desired data streams and interfering streams at node k:

- *Node k is a receiver in time slot t, i.e., $y_k(t) = 1$.* In this case, node k receives $z_m(t)$ data streams from node $m \in \mathcal{T}_k$ and $z_i(t)$ interfering streams from node $i \in \mathcal{P}_k$. Based on (4.3) and (4.4), the $z_i(t)$ interfering streams from node $i \in \mathcal{P}_k$ actually occupy $z_i(t) - \alpha_{ik}(t)$ new directions at node k because $\alpha_{ik}(t)$ interfering streams have been successfully aligned to other interfering streams at node k. Therefore, to ensure the total DoF constraint at node k is satisfied, we must have: $\sum_{m \in \mathcal{T}_k} z_m(t) + \sum_{i \in \mathcal{P}_k} [z_k(t) - \alpha_{ik}(t)] \leq A$.
- *Node k is not a receiver in time slot t, i.e., $y_k(t) = 0$.* In this case, node k does not have any desired data streams from node $m \in \mathcal{T}_k$. So there is no restriction on the number of effective interfering directions at node k. That is, there should be no effective upper bound on $\sum_{i \in \mathcal{P}_k} [z_k(t) - \alpha_{ik}(t)]$.

Combining these two cases, we have

$$\sum_{m\in\mathcal{T}_k} z_m(t) + \sum_{i\in\mathcal{P}_k}[z_i(t) - \alpha_{ik}(t)] \le A + [1 - y_k(t)] \cdot B, \qquad 1 \le k \le N, 1 \le t \le T, \qquad (4.5)$$

where B is a large enough constant (e.g., $B = N \cdot A$) that can be used as a loose upper bound of $\sum_{i\in\mathcal{P}_k}[z_i(t) - \alpha_{ik}(t)]$.

Summary and Example. Collectively, constraints (4.3)–(4.5) characterize a feasible IA space for a multicast network in a time slot. Note that these three constraints are linear and only require simple integer addition and subtraction. By using these constraints, we can avoid designing those complex precoding vectors at the physical layer while still ensuring that our results are feasible. This is a significant advantage, and is the basis for our investigation of multicast IA in multi-hop wireless networks. We summarize our discussions by the following theorem:

Theorem 3. *If constraints (4.3), (4.4), and (4.5) are satisfied, then there exist precoding and decoding vectors at the physical layer so that each transmit node i can send $z_i(t)$ data streams to its one-hop multicast receive nodes in $\mathcal{R}_i$ free of strong interference.*

As an example of how to use constraints (4.3)–(4.5) to ensure feasibility, we return to the example in Fig. 4.1 in Section 4.2. Without causing ambiguity, we omit time slot index t for notation simplicity. In Fig. 4.1, since each transmitter has 1 outgoing data stream, DoF vector of this network is $[z_1, z_2, z_3] = [1, 1, 1]$. Based on the network topology, constraint (4.3) at transmitters N_1, N_2, and N_3 can be written as

$$\text{At transmitter } N_1\text{: } \alpha_{16} + \alpha_{17} \le 1;$$
$$\text{At transmitter } N_2\text{: } \alpha_{25} + \alpha_{27} \le 1;$$
$$\text{At transmitter } N_3\text{: } \alpha_{35} + \alpha_{36} \le 1.$$

Similarly, constraint (4.4) at receivers N_5, N_6, and N_7 can be written as

$$\text{At receiver } N_5\text{: } \alpha_{25} \le z_3 - \alpha_{35} \text{ and } \alpha_{35} \le z_2 - \alpha_{25};$$
$$\text{At receiver } N_6\text{: } \alpha_{16} \le z_3 - \alpha_{36} \text{ and } \alpha_{36} \le z_1 - \alpha_{16};$$

At receiver N_7: $\alpha_{17} \leq z_2 - \alpha_{27}$ and $\alpha_{27} \leq z_1 - \alpha_{17}$.

The above 9 inequalities (constraints) define the IA space for this network: [α_{16}, α_{17}, α_{25}, α_{27}, α_{35}, α_{36}]. Based on Theorem 3, any point $[\alpha_{16}, \alpha_{17}, \alpha_{25}, \alpha_{27}, \alpha_{35}, \alpha_{36}]$ that falls in this space is guaranteed to be feasible at the physical layer. As a validation, consider point $[1, 0, 0, 1, 1, 0]$. It is easy to check that this point meets all 9 constraints and thus falls in the space defined by the 9 constraints. So there exist feasible precoding and decoding vectors at the physical layer to achieve the desired IA. For this point, "$\alpha_{16} = 1$" means that, at receiver N_6, the interfering stream from transmitter N_1 is successfully aligned to the interfering stream from transmitter N_3; "$\alpha_{27} = 1$" means that, at receiver N_7, the interfering stream from transmitter N_2 is successfully aligned to the interfering stream from transmitter N_1; "$\alpha_{35} = 1$" means that, at receiver N_5, the interfering stream from transmitter N_3 is successfully aligned to the interfering stream from transmitter N_2. The IA scheme of this point is exact what happened by using the precoding vectors in (4.1).

Now we check whether or not the point $[1, 0, 0, 1, 1, 0]$ meets the total DoF constraint (4.5) at each receiver. At receive nodes N_4 and N_8, since they do not have interference, it is easy to verify that both nodes meet (4.5). At receive node N_5, we have $\sum_{m \in \mathcal{T}_5} z_m = z_1 = 1$ and $\sum_{i \in \mathcal{P}_5} [z_i - \alpha_{i5}] = (z_1 - \alpha_{15}) + (z_3 - \alpha_{35}) = 1$. Since each node has two antennas (i.e., $A = 2$), receive node N_5 meets (4.5). Similarly, one can verify that receive nodes N_6 and N_7 also meet (4.5). So all five receive nodes meet the total DoF constraint (4.5). Based on Theorem 3, since constraints (4.3), (4.4), and (4.5) are all satisfied, the DoF vector $[z_1, z_2, z_3] = [1, 1, 1]$ is guaranteed to be feasible at the physical layer. Therefore, by using the set of linear constraints in (4.3), (4.4), and (4.5), one can completely avoid dealing with the complex designs of precoding and decoding vectors in (4.1) and (4.2).

4.4 Problem Formulation

Based on the feasible multicast IA space defined by (4.3), (4.4), and (4.5) in the previous section, we study a multicast throughput maximization problem in multi-hop MIMO

networks. Consider a multi-hop MIMO network consisting of a set of nodes as shown in Fig. 4.2(a). Among the nodes there are F multicast sessions. For each session f, it has one (and only one) source node, denoted as S_f, but a group of destination nodes, denotes as set $\mathcal{D}_f$. We assume each multicast tree is computed based on some multicast routing protocol (e.g., MOSPF [62]). Denote $r(f)$ as the achievable data rate of session f. Denote $r_{\min}$ as the minimum achievable data rate among all sessions, i.e., $r_{\min} = \min_{1\leq f\leq F}\{r(f)\}$. Then our objective is to maximize the minimum achievable rate ($r_{\min}$) among all sessions. For this effort, we will focus on the mathematical underpinning of finding the minimum achievable rate. The details of protocol design and overhead evaluation are beyond the scope of this chapter.

Multicast Node Constraints. Denote $x_i(t)$ a binary variable to indicate whether or not node $i \in \mathcal{N}$ is a transmitter in time slot t. Specifically, $x_i(t) = 1$ if node i is a transmitter in time slot t and 0 otherwise. Recall that $y_i(t)$ is a binary variable to indicate whether or not node i is a receiver in time slot t. For each node, we assume that it operates on half duplex.[2] That is, a node cannot be a transmitter and a receiver in the same time slot. Then we have:

$$x_i(t) + y_i(t) \leq 1, \quad 1 \leq i \leq N, 1 \leq t \leq T. \tag{4.6}$$

Consider a node $i \in \mathcal{N}$ in time slot t, if it is a transmitter (i.e., $x_i(t) = 1$), then there is at least one outgoing data stream at node i, i.e., $z_i(t) \geq 1$. Otherwise (i.e., $x_i(t) = 0$), there is no outgoing data streams at node i, i.e., $z_i(t) = 0$. Combining these two cases, we have

$$x_i(t) \leq z_i(t) \leq A \cdot x_i(t), \quad 1 \leq i \leq N, 1 \leq t \leq T\,. \tag{4.7}$$

In a time slot t, if $x_i(t) = 1$ (i.e., node $i \in \mathcal{N}$ is a transmitter), then each of its next-hop nodes in $\mathcal{R}_i$ (see Fig. 4.3) should be a receiver, i.e., $y_j(t) = 1$ for $j \in \mathcal{R}_i$. Otherwise (i.e., $x_i(t) = 0$), $y_j(t)$ can be either 0 or 1 for $j \in \mathcal{R}_i$, i.e., no restriction on $y_j(t)$. Combining these two cases, we have

$$x_i(t) \leq y_j(t), \quad 1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq t \leq T\,. \tag{4.8}$$

[2]Although there is significant recent advance on full duplex on single-antenna node, practical design of full duplex for a MIMO node remains at research frontier.

One-hop Multicast Rate Constraints. At node i, there are $z_i(t)$ outgoing data streams. In particular, if node i is not a transmitter, then $z_i(t) = 0$. Denote $P_{\rm s}$ as the total transmit power of all outgoing data streams at node i. We assume $P_{\rm s}$ is the same for all transmit nodes. Among the $z_i(t)$ outgoing data streams at node i, we denote p_i^k as the transmit power that is allocated to its kth outgoing data stream. Then we have

$$\sum_{k=1}^{z_i(t)} p_i^k(t) = P_{\rm s} \cdot x_i(t), \quad 1 \le i \le N, 1 \le t \le T\,, \tag{4.9}$$

where $x_i(t)$ is the binary variable indicating whether or not node i is an active transmitter.

Referring to see Fig. 4.3, consider one-hop multicast data transmission from transmit node i and the receive nodes in $\mathcal{R}_i$. Due to wireless multicast advantage [33, 93], the transmitted data streams from node i can be received by all the receive nodes in $\mathcal{R}_i$ simultaneously. But the distances from node i to different nodes in $\mathcal{R}_i$ are likely different. As a result, the received signal strengths at different nodes in $\mathcal{R}_i$ are likely different, leading to different achievable data rates. Therefore, the achievable data rate is limited by the receive node in $\mathcal{R}_i$ that has the smallest SINR. Denote $c_i^k(t)$ as the achievable data rate of transmit node i's kth outgoing data stream in a one-hop multicast. Then we have

$$c_i^k(t) = \min_{j \in \mathcal{R}_i} \left\{ W \cdot \log_2 \left(1 + \frac{G_{ij} \cdot p_i^k(t)}{\sum_{h \in \mathcal{N} \setminus \mathcal{P}_j}^{h \neq i} G_{hj} \cdot P_{\rm s} \cdot x_h(t) + P_{\rm n}} \right) \right\},$$
$$1 \le i \le N, 1 \le k \le z_i(t), 1 \le t \le T, \tag{4.10}$$

where W is the channel bandwidth, G_{ij} is the path loss from transmit node i to receive node j, $P_{\rm n}$ is the noise power at the receiver. It should be noted that when calculating the effective SINR at receive node j in (4.10), we only need to consider the weak interference from the unintended transmitters in $\mathcal{N} \setminus \mathcal{P}_j$, since the strong interference from the unintended transmitters in $\mathcal{P}_j$ has been nullified in the spatial domain by IA and IC.

Denote $c_i(t)$ as the aggregate achievable data rate at transmit node i over its $z_i(t)$ outgoing data streams in time slot t. Then we have

$$c_i(t) = \sum_{k=1}^{z_i(t)} c_i^k(t), \quad 1 \le i \le N, 1 \le t \le T\,. \tag{4.11}$$

Multicast Capacity Constraints. Consider the multicast tree of a session f. It has a root node, a set of internal nodes, and a set of leaves. The root node is the source node of the session and each leaf is a destination node of the session. However, an internal node can be either an intermediate node that helps relay traffic or a destination node of the session. Based on previous discussion on one-hop multicst rate constraints, the data rate from a node $i \in \mathcal{N}$ per (4.9), (4.10), and (4.11) will be received by all of its next-hop nodes in $\mathcal{R}_i$. So considering a single multicast tree f, to achieve an average rate of $r(f)$ for the entire tree, we must have the root node and each of the internal nodes in the tree send out an average rate of $r(f)$ over T time slots.

When there are multiple multicast sessions in the network, a node $i \in \mathcal{N}$ may be a root node, an internal node, or a leaf node on one or multiple trees. Only the first two cases will contribute to our discussions here (i.e., a root node or an internal node). Denote $\mathcal{F}_i$ as the set of trees for which node i is used either as a root or an internal node. Then the aggregate data rate that node i needs to send out in T time slots is $\sum_{f\in\mathcal{F}_i} r(f)$. Since $\sum_{f\in\mathcal{F}_i} r(f)$ is upper bounded by the average achievable (outgoing) rate at node i, we have

$$\sum_{f\in\mathcal{F}_i} r(f) \le \frac{1}{T}\sum_{t=1}^{T} c_i(t), \quad 1 \le i \le N\,. \tag{4.12}$$

Finally, since $r_{\min}$ is the minimum achievable data rate among all the multicast trees, we have:

$$r(f) \ge r_{\min}, \quad 1 \le f \le F\,. \tag{4.13}$$

In summary, we have the following problem formulation for multicast throughput maximization (MTM):

$\mathbf{MTM}_1$	max	$r_{\min}$
	s.t.	Multicast IA constraints: (4.3)–(4.5);
		Multicast node constraints: (4.6)–(4.8);
		One-hop multicast rate constraints: (4.9)–(4.11);
		Multicast capacity constraints: (4.12)–(4.13);

where N, T, F, B, W, A, $P_{\rm n}$, $P_{\rm s}$, and G_{ij} are constants; $x_i(t)$ and $y_i(t)$ are binary variables; $z_i(t)$ and $\alpha_{ij}(t)$ are nonnegative integer variables; $r(f)$, $c_i(t)$, $p_i^k(t)$, and $c_i^k(t)$ are continuous variables.

This optimization problem is in the form of mixed integer nonlinear program (MINLP), which is intractable. In the next section, we show how to eliminate the nonlinear constraints in problem MTM_1 through reformulation and approximation.

4.5 Problem Reformulation and $(1-\varepsilon)$-Optimal Approximation

In problem MTM_1, (4.3)–(4.4), (4.6)–(4.8), and (4.12)–(4.13) are linear constraints. But (4.9)–(4.11) are nonlinear constraints. The goal of this section is to perform reformulation and approximation so that the final optimization problem does not have any nonlinear constraints and achieves a $(1-\varepsilon)$-optimality of the original problem (MTM_1).

4.5.1 Problem Reformulation

Note that in (4.9), (4.10), and (4.11), $z_i(t)$ is not a constant but rather an optimization variable. This prevents all three constraints from being linear. In addition to variable $z_i(t)$, constraint (4.10) also has nonlinear function that further prevents it from being linear.

To linearize these constraints at transmit node i, we first focus on the reformulation of $z_i(t)$ in (4.9), (4.10), and (4.11). For transmit node i, we introduce $[A - z_i(t)]$ dummy outgoing data streams and add constraints to force the data rate of each dummy stream to zero. To distinguish a dummy stream from a real data strem, we define a new binary variable $\lambda_i^k(t)$ to indicate whether or not data stream k is dummy at node i in time slot t. Specifically, $\lambda_i^k(t) = 0$ if data stream k is dummy and 1 otherwise. Therefore, we have

$$\sum_{k=1}^{A} \lambda_i^k(t) = z_i(t), \quad 1 \le i \le N, 1 \le t \le T\,. \tag{4.14}$$

Consider the k outgoing data stream at transmit node i. If $\lambda_i^k(t) = 0$ (i.e., the kth

stream is dummy), then the transmit power allocated for this stream is 0. Otherwise (i.e., $\lambda_i^k(t) = 1$), the transmit power allocated for this data stream is upper bounded by the total transmit power $P_{\rm s}$. Therefore, we have

$$0 \le p_i^k(t) \le P_{\rm s} \cdot \lambda_i^k(t),\ 1 \le i \le N, 1 \le k \le A, 1 \le t \le T\,. \tag{4.15}$$

Similarly, we use the following constraints to force the achievable data rate of a dummy data stream to zero:

$$0 \le c_i^k(t) \le B \cdot \lambda_i^k(t), \quad 1 \le i \le N, 1 \le k \le A, 1 \le t \le T\,, \tag{4.16}$$

where B is a large enough constant number that we defined earlier.

The new constraints (4.14)–(4.16) ensure that a dummy data stream has zero transmit power and zero data rate. With the notion of dummy stream and constraints (4.14)–(4.16), constraint (4.9) can be rewritten as:

$$\sum_{k=1}^{A} p_i^k(t) = P_{\rm s} \cdot x_i(t), \quad 1 \le i \le N, 1 \le t \le T\,. \tag{4.17}$$

constraint (4.11) can be rewritten as:

$$c_i(t) = \sum_{k=1}^{A} c_i^k(t), \quad 1 \le i \le N, 1 \le t \le T\,. \tag{4.18}$$

constraint (4.10) can be rewritten as:

$$c_i^k(t) = \min_{j \in \mathcal{R}_i} \left\{ W \cdot \log_2 \left(1 + \frac{G_{ij} \cdot p_i^k(t)}{\sum_{h \in \mathcal{N} \setminus \mathcal{P}_j}^{h \ne i} G_{hj} \cdot P_{\rm s} \cdot x_h(t) + P_{\rm n}} \right) \right\},$$
$$1 \le i \le N, 1 \le k \le A, 1 \le t \le T. \tag{4.19}$$

Now except (4.19), all other constraints (4.14)–(4.18) are linear. We now focus on (4.19), which contains both a fraction and a log function. We consider linearization of the fraction first, and will address the log function later in Section 4.5.2.

Denote $\gamma_{ij}^k(t)$ as:

$$\gamma_{ij}^k(t) = \frac{G_{ij} \cdot p_i^k(t)}{\sum_{h \in \mathcal{N} \setminus \mathcal{P}_j}^{h \ne i} G_{hj} \cdot P_{\rm s} \cdot x_h(t) + P_{\rm n}},$$

$$1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, 1 \leq t \leq T\ . \tag{4.20}$$

Then (4.20) can be equivalently rewritten as:

$$\sum_{h \in \mathcal{N} \setminus \mathcal{P}_j}^{h \neq i} G_{hj} \cdot P_{\mathrm{s}} \cdot x_h(t) \cdot \gamma_{ij}^k(t) + P_{\mathrm{n}} \cdot \gamma_{ij}^k(t) = G_{ij} \cdot p_i^k(t),$$
$$1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, 1 \leq t \leq T. \tag{4.21}$$

Constraint (4.21) is still nonlinear because it has nonlinear term $x_h(t) \cdot \gamma_{ij}^k(t)$. We use the Reformulation-Linearization Technique (RLT) to handle this product of variables [35, Chapter 6]. Define a new variable $w_{ij}^{hk}(t) = x_h(t) \cdot \gamma_{ij}^k(t)$. Then (4.21) can be rewritten as:

$$\sum_{h \in \mathcal{N} \setminus \mathcal{P}_j}^{h \neq i} G_{hj} \cdot P_{\mathrm{s}} \cdot w_{ij}^{hk}(t) + P_{\mathrm{n}} \cdot \gamma_{ij}^k(t) = G_{ij} \cdot p_i^k(t),$$
$$1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, 1 \leq t \leq T. \tag{4.22}$$

In order to ensure that $w_{ij}^{hk}(t) = x_h(t) \cdot \gamma_{ij}^k(t)$ holds, we need to add the following two sets of additional linear constraints:

$$0 \leq w_{ij}^{hk}(t) \leq \gamma_{ij}^k(t), \qquad 1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, h \in \mathcal{N} \setminus \mathcal{P}_j \setminus \{i\}, 1 \leq t \leq T. \tag{4.23}$$

$$\gamma_{ij}^k(t) - [1 - x_h(t)] \cdot B \leq w_{ij}^{hk}(t) \leq x_h(t) \cdot B,$$
$$1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, h \in \mathcal{N} \setminus \mathcal{P}_j \setminus \{i\}, 1 \leq t \leq T, \tag{4.24}$$

where B is a large enough number that we defined earlier. Now (4.21) can be replaced by constraints (4.22)–(4.24). Based on the definition of $\gamma_{ij}^k(t)$, nonlinear constraint (4.19) can be rewritten as:

$$c_i^k(t) = \min_{j \in \mathcal{R}_i} \Big\{ W \cdot \log_2 \big(1 + \gamma_{ij}^k(t)\big) \Big\}, \qquad 1 \leq i \leq N, 1 \leq k \leq A, 1 \leq t \leq T,$$

which is equivalent to:

$$c_i^k(t) \leq W \cdot \log_2 \big(1 + \gamma_{ij}^k(t)\big), \qquad 1 \leq i \leq N, j \in \mathcal{R}_i, 1 \leq k \leq A, 1 \leq t \leq T. \tag{4.25}$$

By replacing (4.9)–(4.11) with (4.14)–(4.18) and (4.22)–(4.25), problem MTM_1 is reformulated as:

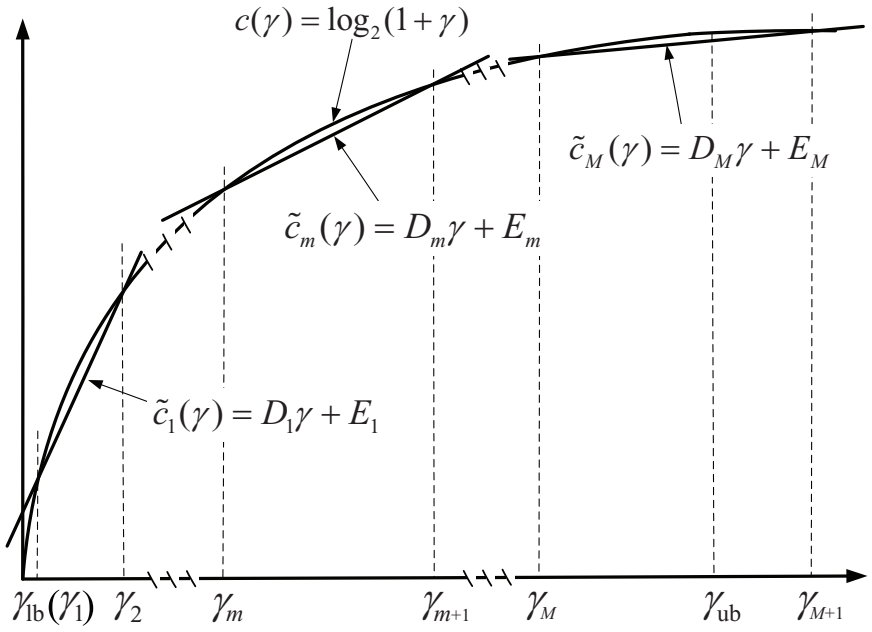


Figure 4.5: Linear approximation for log function $c_1(\gamma) = \log_2(1+\gamma)$.

MTM$_2$	max	$r_{\min}$
	s.t.	Multicast IA constraints: (4.3)–(4.5);
		Multicast node constraints: (4.6)–(4.8);
		Multicast capacity constraints: (4.12)–(4.13);
		One-hop multicast rate constraints: (4.14)–(4.18), (4.22)–(4.25).

Denote $r^*_{\min}(\text{MTM}_1)$ and $r^*_{\min}(\text{MTM}_2)$ as the optimal objective values of MTM_1 and MTM_2, respectively. We have the following lemma:

Lemma 8. *Problems MTM$_1$ and MTM$_2$ have the same optimal objective value, i.e.,* $r^*_{\min}(\text{MTM}_1) = r^*_{\min}(\text{MTM}_2)$.

A proof of this lemma is based on solution construction. Specifically, for an optimal solution to problem MTM_1, we can always construct a feasible solution to problem MTM_2 that achieves the optimal objective value of MTM_1, and vice versa. We omit its proof to save space.

4.5.2 $(1-\varepsilon)$-Optimal Approximation

In problem MTM_2, (4.25) still involves log function and is the only nonlinear constraint. To linearize the log function in (4.25), we will resort to linear approximation. The goal is to replace the log function with a minimum number of linear constraints while ensuring the gap between the two does not exceed ε, a target performance gap for the objective value.

Consider the log function in (4.25). Its variable γ_{ij}^k is the SINR of a data stream from transmit node i to its intended receive node j (see (4.20)). To linearize the log function, we first characterize the range of its variable γ_{ij}^k. Note that the range for each link may differ due to difference in link distance. So instead of finding the range for each individual link (i,j), we use a universal lower bound γ_{lb} and a universal upper bound γ_{ub} for all links. The range of γ_{ij}^k for each link (i,j) will fall in $[\gamma_{\text{lb}},\ \gamma_{\text{ub}}]$. Therefore, instead of designing a unique linear approximation for the log function for each individual link (i,j), we will use one (identical) linear approximation for the log function, over the range of $[\gamma_{\text{lb}},\ \gamma_{\text{ub}}]$, for all links. Note that, as we will prove in Theorem 4, the employment of such an identical linear approximation for all links will not induce any infeasible solution. For an upper bound γ_{ub}, we can use the best possible scenario where there is no interference at receive node j. Then, $\gamma_{\text{ub}} = \max_{i,j\in\mathcal{N}}\{G_{ij}P_{\text{s}}/P_{\text{n}}\}$. For a lower bound γ_{lb}, we can use the worst-case scenario where all other nodes in the network are transmitters and interfere with receive node j. Then, $\gamma_{\text{lb}} = \min_{i,j\in\mathcal{N}}\left\{G_{ij}P_{\text{s}}/(\sum_{k\in\mathcal{N}}^{k\neq i,j} G_{kj}P_{\text{s}} + P_{\text{n}})\right\}$.

As illustrated in Fig. 4.5, to approximate the log function $c(\gamma) = \log_2(1+\gamma)$ over $[\gamma_{\text{lb}},\ \gamma_{\text{ub}}]$, we need multiple lines. Denote the set of lines as $\left\{\tilde{c}_m(\gamma) = D_m\cdot\gamma + E_m : 1 \le m \le M\right\}$, where m is the line sequence number, D_m is the line slope, and E_m is the vertical starting point of the line. M is the minimum number of required lines for the approximation, which will be calculated later. To start with, let's consider the first line $\tilde{c}_1(\gamma) = D_1\cdot\gamma + E_1$. Denote γ_1 and γ_2 as the two points on $\log_2(1+\gamma)$ where the line intersects (see Fig. 4.5). Then $\gamma_1 = \gamma_{\text{lb}}$. For γ_2, we shall find its maximum value so that the performance gap of this line approximation is upper bounded by ε, i.e.,

$$0 \le \frac{c(\gamma) - \tilde{c}_1(\gamma)}{c(\gamma)} \le \varepsilon, \quad \forall\gamma \in [\gamma_1,\ \gamma_2]\,. \tag{4.26}$$

Through algebraic derivations, we find that

$$\gamma_2 = b_1 + \sqrt{b_1^2 - 4b_2} - \gamma_1 \ ,$$

where $b_1 = 2\gamma_1 + \varepsilon(1+\gamma_1)\ln(1+\gamma_1)$ and $b_2 = \gamma_1^2 - \varepsilon(1+\gamma_1)\ln(1+\gamma_1)$. Then we can find D_1 and E_1 in $\tilde{c}_1(\gamma) = D_1 \cdot \gamma + E_1$ as:

$$D_1 = \frac{\log_2(1+\gamma_2) - \log_2(1+\gamma_1)}{\gamma_2 - \gamma_1} \ ,$$
$$E_1 = \log_2(1+\gamma_1) - D_1 \cdot \gamma_1 \ .$$

After obtaining the first line, we can compute the second line by following the same token (but starting from point $(\gamma_2, \log_2(1+\gamma_2))$. Sequentially, we can compute all the lines, until the whole range of $[\gamma_{\rm lb},\ \gamma_{\rm ub}]$ is covered. The procedure of computing the parameters of the approximation lines is given in Alg. 1. We note that our approach yields the minimum number of lines for the linear approximation of the log function. We also note that the last line will exceed $\gamma_{\rm ub}$ almost surely (see Fig. 4.5). As we shall prove in Theorem 4, this extra range (i.e., $[\gamma_{\rm ub},\ \gamma_{M+1}]$) will not induce any infeasible solution to the problem.

We now employ Alg. 1 to linearize the log function in (4.25) in MTM_2. For (4.25), its feasible region is within the shadowed area underneath the log curve as shown in Fig. 4.6(a). We approximate this area by a polygon defined by the lines generated in Alg. 1 over $[\gamma_{lb}, \gamma_{ub}]$, as shown in Fig. 4.6(b). Therefore, (4.25) is approximated by the following constraint:

$$c_i^k(t) \le W \cdot \big(D_m \gamma_{ij}^k(t) + E_m\big), \qquad 1 \le i \le N, j \in \mathcal{R}_i, 1 \le k \le A, 1 \le t \le T, 1 \le m \le M \ , \quad (4.27)$$

where M, D_m, and E_m are computed in Alg. 1.

By replacing (4.25) in MTM_2 with (4.27), the resulting optimization problem, which we denote as MTM_3, can be written as

MTM$_3$	max	$r_{\min}$
	s.t.	Multicast IA constraints: (4.3)–(4.5);
		Multicast node constraints: (4.6)–(4.8);
		Multicast capacity constraints: (4.12)–(4.13);
		One-hop multicast rate constraints: (4.14)–(4.18), (4.22)–(4.24), (4.27).

Algorithm 1 Computing D_m, E_m, and M.

// Initialization

$\gamma_{\rm lb} = \max_{i,j\in\mathcal{N}}\{G_{ij}P_{\rm s}/P_{\rm n}\}$;

$\gamma_{\rm ub} = \min_{i,j\in\mathcal{N}}\{G_{ij}P_{\rm s}/(\sum_{k\in\mathcal{N}}^{k\neq i,j} G_{kj}P_{\rm s} + P_{\rm n})\}$;

$m = 0$;

$\gamma_1 = \gamma_{\rm lb}$;

while $\gamma_m < \gamma_{\rm ub}$ **do**

$m \leftarrow m + 1$;

$\gamma_{m+1} = b_{m,1} + \sqrt{b_{m,1}^2 - 4b_{m,2}} - \gamma_m$ where;

$b_{m,1} = 2\gamma_m + \varepsilon(1+\gamma_m)\ln(1+\gamma_m)$;

$b_{m,2} = \gamma_m^2 - \varepsilon(1+\gamma_m)\ln(1+\gamma_m)$;

$D_m = \frac{\log_2(1+\gamma_{m+1})-\log_2(1+\gamma_m)}{\gamma_{m+1}-\gamma_m}$;

$E_m = \log_2(1+\gamma_m) - \frac{\gamma_m[\log_2(1+\gamma_{m+1})-\log_2(1+\gamma_m)]}{\gamma_{m+1}-\gamma_m}$;

end while

$M = m$;

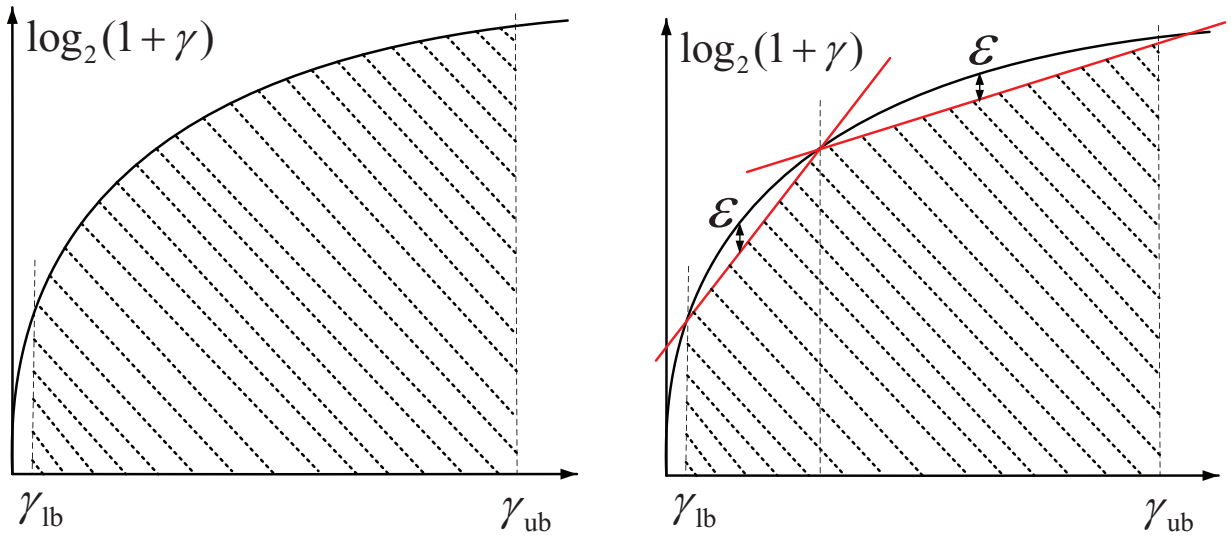
Denote $r^*_{\min}(\mathrm{MTM}_3)$ as the optimal objective value of MTM_3. Then we have the following theorem:

Theorem 4. *The optimal objective value of MTM$_3$ is within* $(1-\varepsilon)$*-optimal of the objective value of MTM$_2$, i.e.,* $(1-\varepsilon)\cdot r^*_{\min}(\mathrm{MTM}_2) \le r^*_{\min}(\mathrm{MTM}_3) \le r^*_{\min}(\mathrm{MTM}_2)$.

Proof. We prove this theorem by two steps.

Step 1. We prove $(1-\varepsilon)\cdot r^*_{\min}(\mathrm{MTM}_2) \le r^*_{\min}(\mathrm{MTM}_3)$ by showing that if $r_{\min}$ is the optimal objective value of MTM_2, then $(1-\varepsilon)r_{\min}$ is an achievable objective value of MTM_3. Suppose that $r_{\min}$ is the optimal objective value of MTM_2, i.e., $r_{\min} = r^*_{\min}(\mathrm{MTM}_2)$. Then its corresponding solution $\varphi = [r_{\min}, r(f), c_i(t), c_i^k(t), \gamma_{ij}^k(t), other\_variables]$ is obviously feasible and satisfies all the constraints in MTM_2. Based on φ, we construct a solution $\hat{\varphi}$ to MTM_3 as follows:

$$\hat{\varphi} = \big[(1-\varepsilon)r_{\min},\ (1-\varepsilon)r(f),\ (1-\varepsilon)c_i(t),\ (1-\varepsilon)c_i^k(t),\ \gamma_{ij}^k(t),\ other\_variables\big].$$



(a) Original feasible region

(b) Approximated feasible region

Figure 4.6: Original and approximated feasible regions.

We now show that $\hat{\varphi}$ is a feasible solution to MTM$_3$. Since φ satisfies (4.3)–(4.8), (4.12)–(4.18), and (4.22)–(4.24), it is not difficult to see that $\hat{\varphi}$ also satisfies these constraints. Since φ satisfies (4.25) in MTM$_2$, we have $W\log_2\left(1+\gamma_{ij}^k(t)\right) \geq c_i^k(t)$. Based on Alg. 1, we know that $D_m\gamma_{ij}^k(t)+E_m \geq (1-\varepsilon)\log_2(1+\gamma_{ij}^k(t))$ holds for $m=1,2,\cdots,M$. We therefore have $W\left(D_m\gamma_{ij}^k(t)+E_m\right) \geq (1-\varepsilon)c_i^k(t)$ for $m=1,2,\cdots,M$. This indicates that $\hat{\varphi}$ satisfies (4.27). Since $\hat{\varphi}$ satisfies all the constraints in MTM$_3$, $\hat{\varphi}$ is a feasible solution to MTM$_3$ and its corresponding objective value is $(1-\varepsilon)r_{\min}$. So $(1-\varepsilon)r_{\min}$ is an achievable objective value of MTM$_3$. Therefore, $(1-\varepsilon)\cdot r_{\min}^*(\text{MTM}_2) \leq r_{\min}^*(\text{MTM}_3)$ holds.

Step 2. We prove $r_{\min}^*(\text{MTM}_3) \leq r_{\min}^*(\text{MTM}_2)$ by showing that if $r_{\min}$ is the optimal objective value of MTM$_3$, then $r_{\min}$ is an achievable objective value of MTM$_2$. Suppose that $r_{\min}$ is the optimal objective value of MTM$_3$, i.e., $r_{\min}=r_{\min}^*(\text{MTM}_3)$. Then its corresponding solution $\varphi=[r_{\min}, r(f), c_i(t), c_i^k(t), \gamma_{ij}^k(t), other\_variables]$ is obviously feasible and satisfies all the constraints in MTM$_3$.

We now show that φ is a feasible solution to MTM$_2$. Since φ is a feasible solution to MTM$_3$, we know that φ satisfies (4.3)–(4.8), (4.12)–(4.18), (4.22)–(4.24), and (4.27). Since

φ satisfies (4.27), we have $c_i^k(t) \leq W\big(D_m\gamma_{ij}^k(t) + E_m\big)$ for $m = 1, 2, \cdots, M$. Based on Alg. 1, we know that $D_m\gamma_{ij}^k(t) + E_m \leq \log_2(1+\gamma_{ij}^k(t))$ holds for some m. We therefore have $c_i^k(t) \leq W \cdot \log_2\big(1+\gamma_{ij}^k(t)\big)$. This indicates that φ satisfies (4.25). Since φ satisfies all the constraints in MTM_2, φ is a feasible solution to MTM_2 and its corresponding objective value is $r_{\min}$. So $r_{\min}$ is an achievable objective value of MTM_2. Therefore, $r_{\min}^*(\mathrm{MTM}_3) \leq r_{\min}^*(\mathrm{MTM}_2)$ holds. □

Combining Theorem 4 and Lemma 8, we have the following corollary:

Corollary 1. *The optimal objective value of MTM$_3$ is within* $(1-\varepsilon)$*-optimal of the objective value of MTM$_1$, i.e.,* $(1-\varepsilon) \cdot r_{\min}^*(\mathrm{MTM}_1) \leq r_{\min}^*(\mathrm{MTM}_3) \leq r_{\min}^*(\mathrm{MTM}_1)$.

This corollary is straightforward and thus we omit its proof.

To conclude, we have successfully transformed MTM_1, a MINLP, to MTM_3, which is mixed integer linear program (MILP). Although the theoretical worst-case complexity of solving a general MILP problem is exponential [74], there exist highly efficient optimal algorithms (e.g., branch-and-bound with cutting planes [75]) and heuristic algorithms (e.g., sequential fixing algorithm [34]). For most of practical-sized networks, an off-the-shelf solver such as CPLEX [104] may also be effective. Since the goal of this chapter is to explore the throughput gain of multicast IA in a multi-hop MIMO network, any of the above techniques in [34, 74, 75] may be employed.

4.6 Performance Evaluation

In this section, we use numerical results to demonstrate the benefits of IA in multicast communications. For this purpose, we will use CPLEX [104] to solve problem MTM_3 and compare its optimal objective value against that when IA is not used.

4.6.1 Simulation Setting

We consider a multi-hop MIMO network with 30 nodes, which are randomly distributed in a 1000 m $\times$ 1000 m square area. For each node in the network, we assume it is equipped with four antennas and has the transmit power of 24 dBm when transmitting. The power attenuation from node i to node j is calculated as per the path loss formula in LTE specification (3GPP TR 25.951 V10.0.0), which is $G_{ij} = 10^{-L_{\text{PL}}/10}$ with $L_{\text{PL}} = 32.9 + 37.5 \log_{10}(d_{ij})$ and d_{ij} being the distance between nodes i and j. The noise power at a receive node is -104 dBm. The transmission range of a transmit node is 300 m. We will discuss interference range setting shortly. Transmission scheduling is done in time slots. We assume there are four time slots in a frame and the channel bandwidth is 10 MHz.

For each multicast session in the network, its source node is selected randomly and so are its destination nodes. The routing tree for the multicast session is computed by the MOSPF protocol [62]. The approximation error ε is set to 0.05.

4.6.2 Impact of Interference Range D

In Section 4.3 we use interference range D to classify interferences from different transmit nodes as either a strong or a weak interference at a receiver. The strong interference will be handled by MIMO's IA and IC capabilities, while the weak interference will be treated as noise in the SINR calculation. Therefore, the setting of D is important in solving problem MTM_3.

To see the impact of D on the objective value in MTM_3, consider an example network instance shown in Fig. 4.7. Here, we have one multicast session with source node (denoted as $S[0]$ in the figure) and 15 destination nodes (denoted as $D[0]$ in the figure). The multicast routing tree in Fig. 4.7 was found by MOSPF [62]. In Fig. 4.8(a), we present the normalized throughput w.r.t. bandwidth (in bits/sec/Hz) under different settings of D. The normalized throughput is obtained by solving the problem MTM_3 using CPLEX. In Fig. 4.8(a), $D = 0$ represents the special case where all interferences are treated as noise. In this case, neither IA nor IC will be used. Without IA and IC, those strong interferences will be treated as

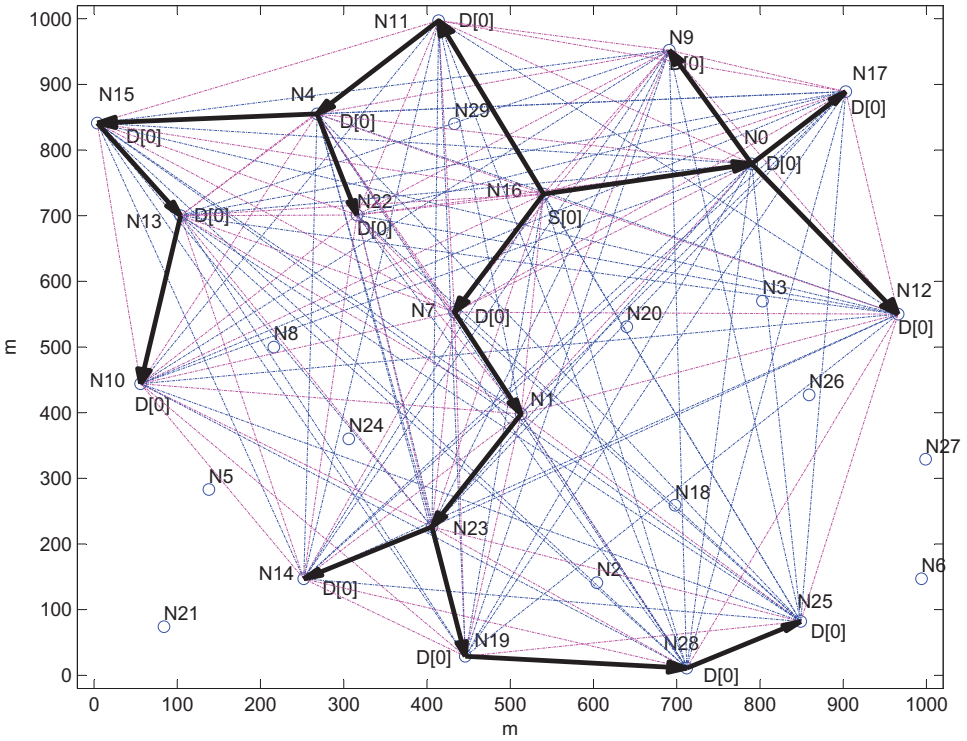
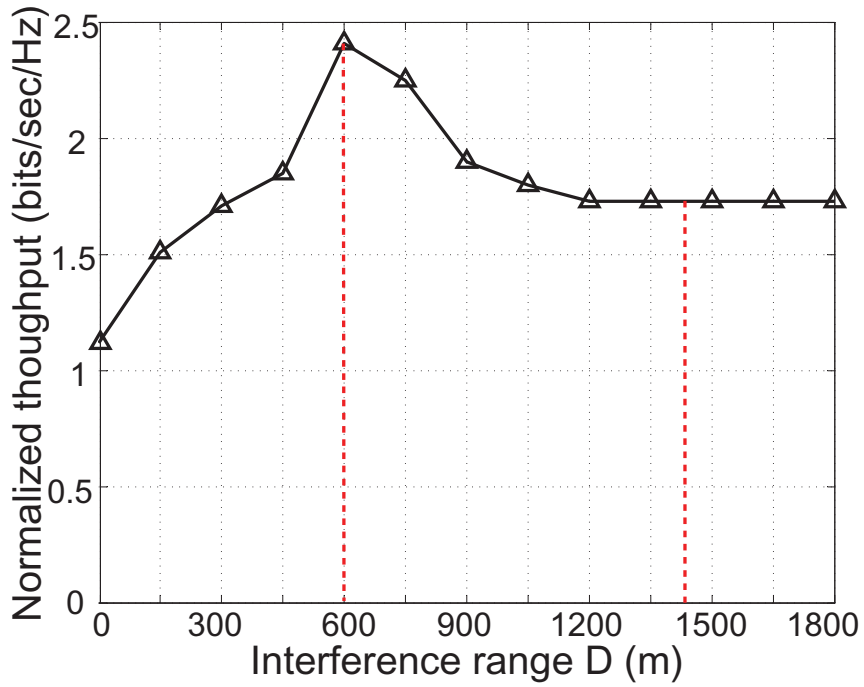


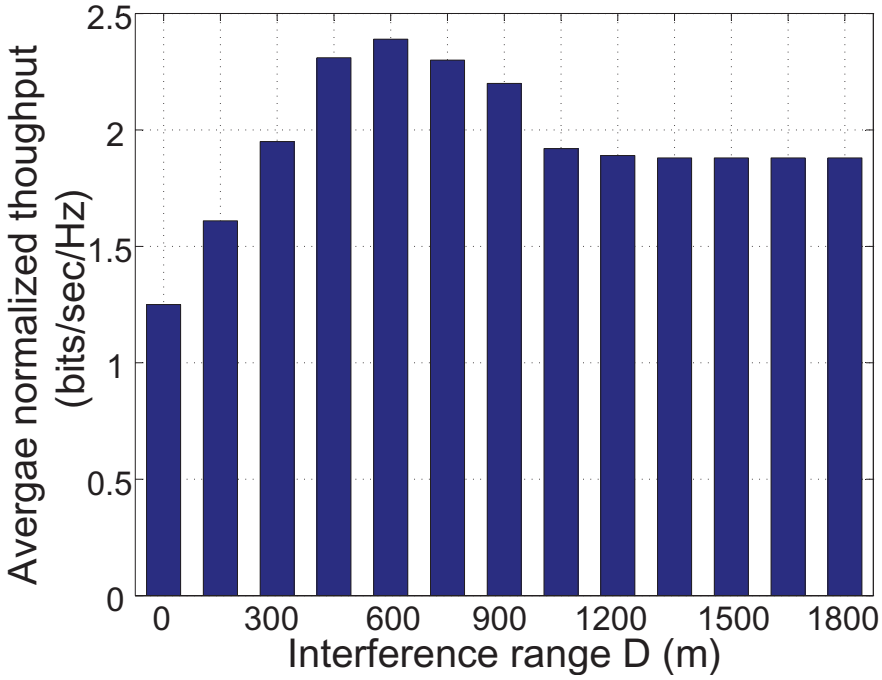
Figure 4.7: A network instance for the case study.

noise in the SINR calculation at each receiver, which will severely reduce the achievable throughput. This point ($D = 0$) represents the smallest throughput for all settings of D.

When $D > 0$, the normalized throughput varies depending on the setting of D. Note that the peak occurs at $D = 600$ m, which is twice the transmission range (300 m). This is not surprising, and serves as a validation of the findings by Shi et al. in [79]. When $0 \leq D \leq 600$, the normalized throughput is strictly increasing with the value of D. This is because the increase of D allows more strong interferences to be included for possible nullification by MIMO's IA and IC capabilities and thus SINR will increase, so will the achievable rate of a session. When $600 \leq D \leq 1414$, the normalized throughput is decreasing with the value of D. This is because the increase of D results in a waste of MIMO's DoFs on canceling weak interference, leaving fewer number of DoFs available for data transmission. Finally, when $D \geq 1414$, the normalized throughput flattens out. Since 1414 is the farthest possible distance between any two nodes in a 1000×1000 area, all the interferences at a receiver



(a) One network instance



(b) 50 network instances (average)

Figure 4.8: Impact of D on the normalized throughput.

will be considered as strong interference and will be handled by IA and IC when $D \geq 1414$. Further increase of D will not change the classification of strong or weak interference in the network. Therefore, the normalized throughput will stay flat.

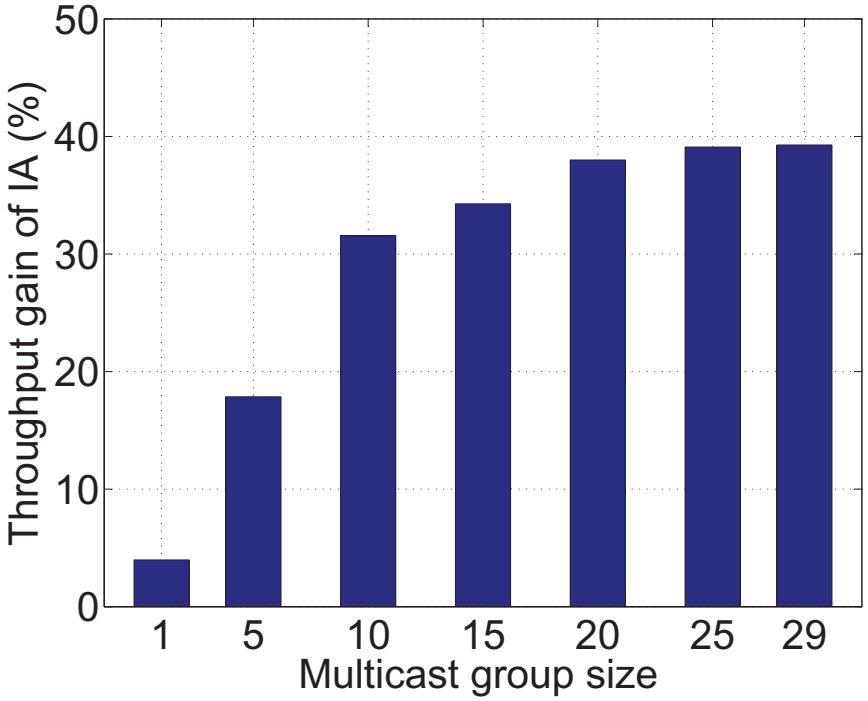
In Fig. 4.8(b), we present the average throughput (normalized w.r.t. bandwidth) over 50 network instances under different settings of D. This average result is consistent with our findings in Fig. 4.8(a).

4.6.3 Throughput Gain of Multicast IA

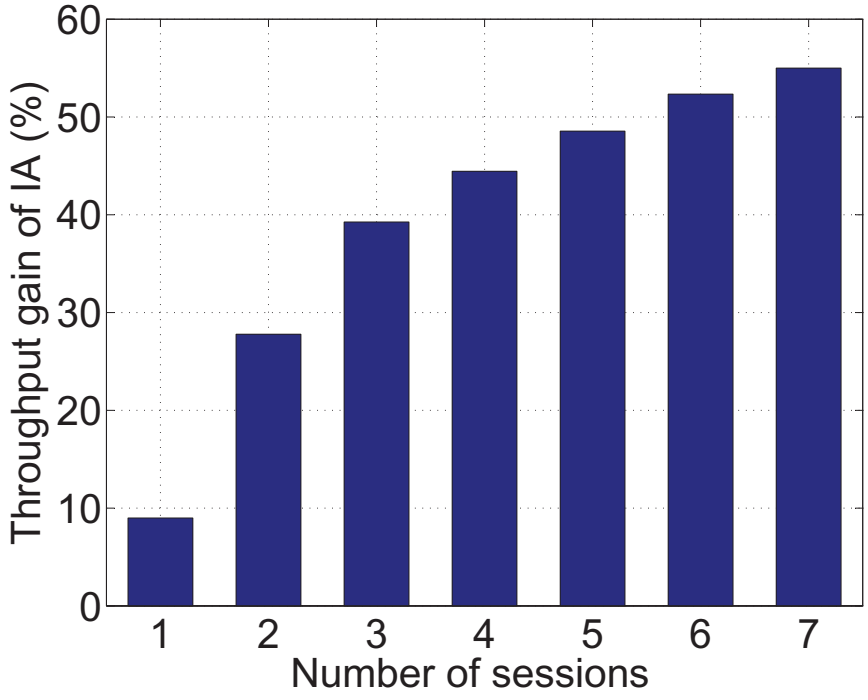
For the rest of this section, we set $D = 600$. We will compare the multicast throughput of a network with IA (obtained by solving MTM_3) to the throughput of the same network without IA. The formulation of the latter problem, denoted as MTM-noIA. Recall that in MTM_3, α_{ik} is the optimization variable that characterizes the IA capability at the physical layer. So it suffices to disable the IA capability at the physical layer by simply fixing $\alpha_{ik} = 0$ for $i, k \in \mathcal{N}$. As such, the same throughput maximization problem without using IA can be formulated as follows:

MTM-noIA	max	$r_{\min}$
	s.t.	Multicast IA constraints: (4.3)–(4.5);
		Multicast node constraints: (4.6)–(4.8);
		Multicast capacity constraints: (4.12)–(4.13);
		One-hop multicast constraints: (4.14)–(4.18), (4.22)–(4.24), (4.27);
		$\alpha_{ik} = 0$ for $i, k \in \mathcal{N}$.

Impact of Multicast Group Size. We first study throughput under different multicast group sizes (i.e., the number of destination nodes in a multicast session). We consider 50 randomly generated 30-node network instances. There is one multicast session and its destination group size varies from 1 (unicast case), 5, 10, $\cdots$, to 29 (broadcast case). Table 4.2(a) presents the average multicast throughput for different group sizes, where the average throughput is taken over the 50 network instances. Figure 4.9(a) presents the corresponding multicast throughput gain of IA under different multicast group sizes. For example,



(a) Throughput gain of IA under different multicast group sizes



(b) Throughput gain of IA under different session numbers

Figure 4.9: Percentage of multicast throughput gain of IA.

Table 4.2: Average multicast throughput over 50 network instances with and without IA.

(a) Normalized throughput under different multicast group sizes

Multicast group size	1	5	10	15	20	25	29
Throughput without IA	6.28	2.81	2.06	1.78	1.50	1.33	1.23
Throughput with IA	6.53	3.30	2.71	2.39	2.07	1.85	1.72

(b) Normalized throughput under different multicast session numbers

# of multicast sessions	1	2	3	4	5	6	7
Throughput without IA	3.56	1.80	1.07	0.72	0.552	0.407	0.28
Throughput with IA	3.88	2.30	1.49	1.04	0.82	0.62	0.434

when the multicast group size is 15, the average multicast throughput of the 50 network instances is 1.78 (without IA) and 2.39 (with IA), as shown in Table 4.2(a). Therefore, the percentage of multicast throughput gain due to IA is $(2.39 - 1.78)/1.78 \approx 34\%$, as shown in Fig. 4.9(a).

From Table 4.2(a) we can see that the multicast throughput of a network, with or without IA, decreases as the multicast group size increases. However, from Fig. 4.9(a) we can see that the multicast throughput gain of IA becomes more significant as the increase of multicast group size. This is not surprising, as larger multicast group size leads to more interference among the links, thereby providing more opportunities for IA.

Impact of Multicast Session Number. We now study throughput when the number of multicast session increases. Again we consider 50 randomly generated 30-node network instances. For each network instance, we vary the number of multicast sessions. For

each multicast session, it has three destination nodes, again chosen randomly. Table 4.2(b) presents the average multicast throughput for different numbers of multicast sessions, where the average throughput is taken over the results from 50 network instances. Figure 4.9(b) presents the corresponding multicast throughput gain of IA. For example, when the number of sessions is 4, the average multicast throughput is 0.72 (without IA) and 1.04 (with IA), as shown in Table 4.2(b). Then the percentage of multicast throughput gain due to IA is $(1.04 - 0.72)/0.72 \approx 44\%$, as shown in Fig. 4.9(b).

From Table 4.2(b) we can see that the multicast throughput of a network, with or without IA, decreases as the number of sessions increases. If we keep increasing the number of sessions in the network, the throughput (the optimal objective value of MTM_3) will eventually go to 0 (either with IA or without IA). This is because if there are too many links on the multicast trees, the MIMO's IA and IC capabilities as well as the time slot resource cannot support the activation of every link. But within the schedulable region (nonzero throughput region) of the network, from Fig. 4.9(b) we can see that the multicast throughput gain of IA becomes more significant as the number of multicast sessions increases. This is intuitive, as the more multicast sessions in the network, the more interference among the links, thereby rendering more room for IA among the interferences.

4.7 Related Work

We review the related work along the following two research lines.

IA in multi-hop networks. The concept of IA was coined by Jafar and Shamai for the two-user X channel in [39]. The most significant result was developed by Cadambe and Jafar [9], in which they showed that the K-user interference channel could achieve $K/2$ DoFs. Since then, the results of IA have been developed for a variety of channels and networks, such as the MIMO-OFDM channel [15], the cellular network [84], and the WLAN [23]. While there is a large volume of papers that studied IA for single-hop communications from information-theoretic perspective, the advance of IA in multi-hop and MIMO networks remains scarce. In [51], Li et al. attempted to explore IA in a multi-hop MIMO network. The

idea of IA was described in several examples in the paper to illustrate its benefits. However, the key concept of IA was not incorporated into their proposed algorithm and was absent in the final solution. In [1], Abdel-Hadi and Vishwanath studied multicast IA for "multihop" single-antenna networks. There, "multihop" was only limited to two-hop networks and it was unclear how to extend their multicast IA scheme to general multi-hop (more than two hops) networks. In [100], Zeng et al. developed an IA model for MIMO networks and used that model to solve network throughput problem in a multi-hop MIMO network. The efforts in [100] laid the foundation for studying multicast IA in this chapter.

Muticast in MIMO networks. In the literature, there is a large body of efforts that studied multicast communication in multi-hop wireless networks (see, e.g., [4, 33, 99]). However, most of them focused on multicast in single-antenna wireless networks. To date, results on multicast in MIMO networks remain limited. In [20], Ge et al. studied multicast communications in both single-antenna and MIMO networks, with the objective of improving multicast throughput and reliability. However, their analysis and proposed multicast transmission scheme were limited to single-hop networks. In [95], Xu et al. proposed an adaptive resource allocation (ARA) scheme for multicast communication in MIMO-OFDM cellular networks. Their ARA scheme was tailored for cellular networks with a BS and multiple users and could not be applied to multi-hop MIMO networks. In [18], Gao et al. studied a multicast communication problem in a multi-hop MIMO network where each node was equipped with a cognitive radio (CR), with the objective of minimizing the required bandwidth while meeting the throughput requirement of each session. They formulated this problem as a mixed-integer optimization problem and then developed a heuristic algorithm to solve it. In [18], only MIMO's IC capability was considered whereas MIMO's IA capability was not explored. To the best of our knowledge, our work in this chapter is the first one that studies multicast communication by exploiting MIMO's IA and IC capacities in general multi-hop networks.

4.8 Chapter Summary

IA is an important advance in the information theory community. Its potential is most profound when there are enough opportunities to align interfering signals at a receiver. As such, multicast communications offer a natural environment to exploit the power of IA. In this chapter, we offer a systematic study of IA for multicast communications in a multi-hop MIMO network. Instead of dealing with complex design of precoding and decoding vectors, we developed a set of linear constraints at both transmitter and receiver to characterize a feasible design space for multicast IA. We showed that, for a set of data streams, as long as the set of linear constraints are satisfied, there exist precoding and decoding vectors at the physical layer to guarantee their feasibility. The set of proposed constraints render a simple abstraction to study multicast IA without being buried by the physical layer details. Based on these constraints, we formulated a multicast throughput maximization problem and showed the significant throughput gain offered by IA. We also find that within schedulable region (with non-zero throughput), the throughput gain of IA increases with the size of multicast group and the number of multicast sessions.

Chapter 5

Spectral Interference Alignment in Cellular Networks

5.1 Introduction

In recent years, interference alignment (IA) has become a key technique for interference management in wireless networks. The basic idea of IA is to construct the signals at the transmitters so that at each receiver, the undesired signals (interference) are overlapping in predefined subspace while the desired signals remain separable and decodable. It was shown in [9] that IA allows the aggregate degrees-of-freedom (DoFs) of K-user interference channel to increase linearly with the number of users K (rather than being a constant). Since its inception, the benefits of IA have been recognized and exploited for a variety of interference channels and networks, such as the K-user $M \times N$ interference channel [26], the X network [10], and the MIMO Y channel [48].

The potential benefits of IA have also been studied for the cellular networks [45, 66, 84, 85, 89]. The most significant results in this area were developed by Suh and Tse in [84], where they showed that an IA scheme can achieve $K/(\sqrt[G-1]{K}+1)^{G-1}$ DoFs for each cell, where G is the number of cells and K is the number of users in a cell. As the number of per-cell users is large enough ($K \to +\infty$), each cell can almost achieve one DoF, meaning

that each cell (base station) can serve its users as if there were no interference in the network.

The results in [84] are significant from information theoretic perspective. However, the underlying network settings and assumptions are far from what may happen in practice. Specifically, the work in [84] was based on the following assumptions. (i) Each user in the network is restricted to one data stream. This is not likely to hold in a practical cellular network, particularly when the demand from each user may vary widely due to different applications. (ii) The number of users served by each BS is identical. This assumption makes it convenient to design an IA scheme at a BS, but it is not likely to happen in practice. (iii) The number of available frequency subcarriers in the network is equal to the number of users under a BS plus one. The authors assume this setting so that all of the interference streams are aligned on the same (one) direction at a BS. But in reality, the number of available frequency subcarriers in the network is not dependent on the number of users under a BS. (iv) Each receiver is in the interference range of all transmitters, i.e., symmetric interference pattern. But this is hardly true for a cellular network in practice, where a user (or BS) is only within the interference range of a subset of BSs (or users). In summary, due to the above assumptions in [84], there remains a major technical gap between the findings in [84] and how IA can be applied to a cellular network in practice.

The goal of this chapter is to bridge the gap between the information theoretical results in [84] and how IA can be exploited for more practical settings (without the assumptions in (i) to (iv)). We consider a single-antenna cellular network consisting of a set of BSs and a set of users. User population is randomly distributed in the area. A user may fall into the service areas of multiple BSs and will choose one BS as its service provider. Therefore, for the uplink, a user will interfere with those BSs (other than its chosen service BS) within its interference range. Likewise, for the downlink, a BS will interfere with those users (not served by this BS) in its interference range. A user can transmit/receive any number of data streams, which are only limited by subcarrier resources. The total number of available subcarriers are user-independent and depends on only the communication standards (e.g., 1024 in LTE). Our objective is to exploit IA in the spectral (frequency) domain so as to

maximize the uplink/downlink user throughput in the cellular networks.

Under the above practical settings, this chapter makes the following contributions:

- For the uplink, we propose an IA scheme for each user (transmitter) and BS (receiver). At each user, we propose an approach to determine which subset of its interfering streams should be selected for alignment at a BS. At each BS, we propose a procedure for IA so that the desired data streams remain resolvable. For the proposed IA scheme, we develop a set of IA constraints for each user and BS. We also prove the feasibility of the proposed IA scheme at the physical layer.
- Based on the proposed IA scheme, we develop a mathematical model for the uplink user throughput maximization problem, This model incorporates BS selection in the formulation. To remove nonlinear terms in the formulation, we employ the Reformulation-Linearization Technique. We show that the final formulation is in a form that is suitable for a commercial solver.
- We study the performance of our proposed IA scheme by solving the uplink user throughput maximization problem for different network instances. For comparison, we compare it to two other schemes: "no-IA" scheme and "crude-IA" scheme. Results from 100 network instances show that our IA scheme achieves an average 98% throughput improvement over the no-IA scheme, and an average 39% improvement over the crude-IA scheme.
- For the downlink problem, we show that its IA scheme can be developed based on the same IA scheme for the uplink by reversing the role of a user and BS. Further, the downlink user throughput maximization problem has the same formulation as the uplink problem and can thus be solved in the same way.

The remainder of this chapter is organized as follows. Section 5.2 offers some essential background on IA in cellular networks. Section 5.3 describes a user throughput maximization problem. In Section 5.4, we develop an IA scheme and prove its feasibility. In Section 5.5, we incorporate BS selection into our IA scheme. In Section 5.6, we formulate the uplink user throughput maximization problem. In Section 5.7, we offer numerical results to show the

Table 5.1: Notation for spectral IA in cellular networks.

Symbol	Definition
$\mathcal{A}_{ij}$	The subset of interfering streams at user i that are aligned at unintended BS j, $\mathcal{A}_{ij} \subseteq \mathcal{S}_i$, $\lvert\mathcal{A}_{ij}\rvert = \alpha_{ij}$
$\mathcal{B}_i$	The subset of interfering streams at user i that are not aligned at any BSs, $\mathcal{B}_i = \mathcal{S}_i \backslash (\cup_{j \in \mathcal{I}_i^{\text{bs}}} \mathcal{A}_{ij})$, $\lvert\mathcal{B}_i\rvert = \beta_i$
$\mathcal{C}_i^{\text{bs}}$	The set of BSs within the transmission range of user i
$\mathcal{C}_j^{\text{usr}}$	The set of users that can choose BS j as service provider
$\mathcal{E}^{A_{ij}}$	The subset of encoding vectors that correspond to the interfering streams in $\mathcal{A}_{ij}$
$\mathcal{E}^{B_i}$	The subset of encoding vectors that correspond to the interfering streams in $\mathcal{B}_i$
$\mathcal{E}^{S_i}$	The set of encoding vectors that correspond to the streams in $\mathcal{S}_i$
$\mathbf{H}_{ji}$	Channel matrix from user i to BS j, $\mathbf{H}_{ji} \in \mathbb{C}^{K \times K}$
$\mathcal{I}_i^{\text{bs}}$	The set of BSs that are interfered by user i
$\mathcal{I}_j^{\text{usr}}$	The set of users that are interfering with BS j
K	The number of subcarriers in the network, $K = \lvert\mathcal{K}\rvert$
$\mathcal{K}$	The set of subcarriers in the nework
M	The number of BSs in the network, $M = \lvert\mathcal{M}\rvert$
$\mathcal{M}$	The set of BSs in the network
N	The number of users in the network, $N = \lvert\mathcal{N}\rvert$
$\mathcal{N}$	The set of users in the network
$\mathcal{O}_i^{\text{bs}}$	The set of BSs within the interference range of user i (but out of its transmission range)
$\mathcal{O}_j^{\text{usr}}$	The set of users that interfere with BS j but out of BS j's service area
$\mathcal{Q}_j^{\text{T}}$	The set of directions for desired data streams at BS j
$\mathcal{Q}_j^{\text{I}}$	The set of directions for interfering streams at BS j

Table 5.1: Continued.

$\mathcal{Q}_j^{\mathrm{I,Eff}}$	The set of directions for "effective" interfering streams at BS j
$\mathcal{Q}_j^{\mathrm{I,Ali}}$	The set of directions for aligned interfering streams at BS j
$\mathcal{Q}_j^{\mathrm{I,Def}}$	The set of predefined directions for interference at BS j
s_i^k	The k-th stream at user i, $1 \leq k \leq \sigma_i$
$\mathcal{S}_i$	The set of streams at user i, $\lvert\mathcal{S}_i\rvert = \sigma_i$
$\mathcal{T}_j^{\mathrm{usr}}$	The set of users that choose BS j as service provider
$\mathbf{u}_i^k$	The encoding vector for stream s_i^k over the subcarriers, $\mathbf{u}_i^k \in \mathbb{C}^{K\times 1}$
$\mathbf{v}_j^l$	The decoding vector for stream s_i^k over the subcarriers, $\mathbf{v}_j^l \in \mathbb{C}^{K\times 1}$
$r_{\min}$	The minimum data rate among all users
x_{ij}	A binary variable indicating whether user i chooses BS j as its service provider
y_{ij}	An opposite binary variable of variable x_{ij}, $x_{ij} + y_{ij} = 1$
α_{ij}	The cardinality of $\mathcal{A}_{ij}$, $\alpha_{ij} = \lvert\mathcal{A}_{ij}\rvert$
β_i	The cardinality of $\mathcal{B}_i$, $\beta_i = \lvert\mathcal{B}_i\rvert$
σ_i	The number of streams at user i, $\sigma_i = \lvert\mathcal{S}_i\rvert$
λ_{ij}	Linearization variable, $\lambda_{ij} = \alpha_{ij} \cdot y_{ij}$
μ_{ij}	Linearization variable, $\mu_{ij} = \beta_i \cdot y_{ij}$

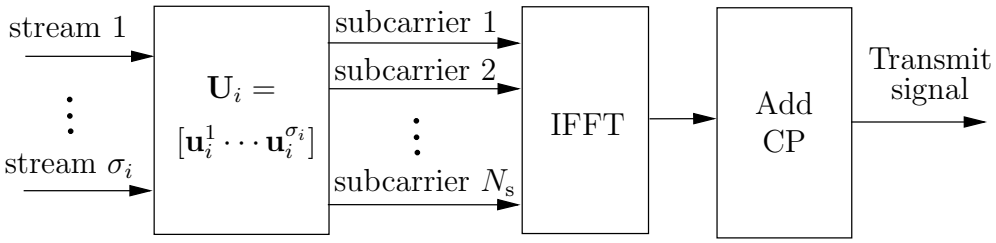


Figure 5.1: Schematic diagram of user i.

efficacy of the IA scheme. Section 5.8 studies the downlink user throughput maximization problem. Section 5.9 presents the related work on IA and Section 5.10 concludes this chapter. Table 5.1 lists the notation that we will use in this chapter.

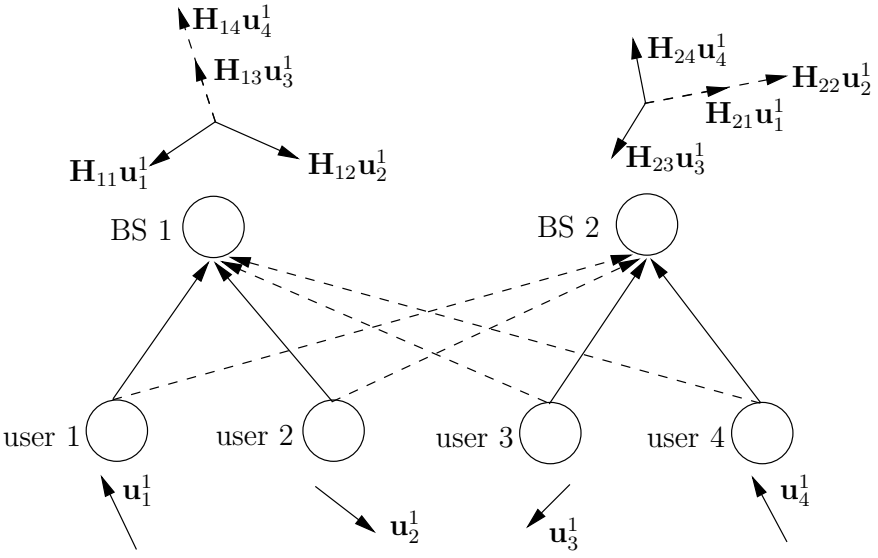
5.2 IA in Cellular Networks: A Primer

In general, IA refers to the construction of transmit signals so that (i) they overlap at the *unintended* receivers, and (ii) they remain resolvable (i.e., not being overlapped by other streams) at the *intended* receivers. In the context of cellular networks, we consider an IA scheme in the frequency domain by mapping each transmit stream onto all of the available subcarriers (e.g., 1024 subcarriers in OFDM). In general, suppose that there are K (e.g., 1024) subcarriers available in the network, then the encoding vector for each outgoing stream has a dimension of $K \times 1$. At each transmitter, one needs to design encoding vector over the K subcarriers, as shown in Fig. 5.1, to ensure that the outgoing signals can overlap at their unintended receivers.

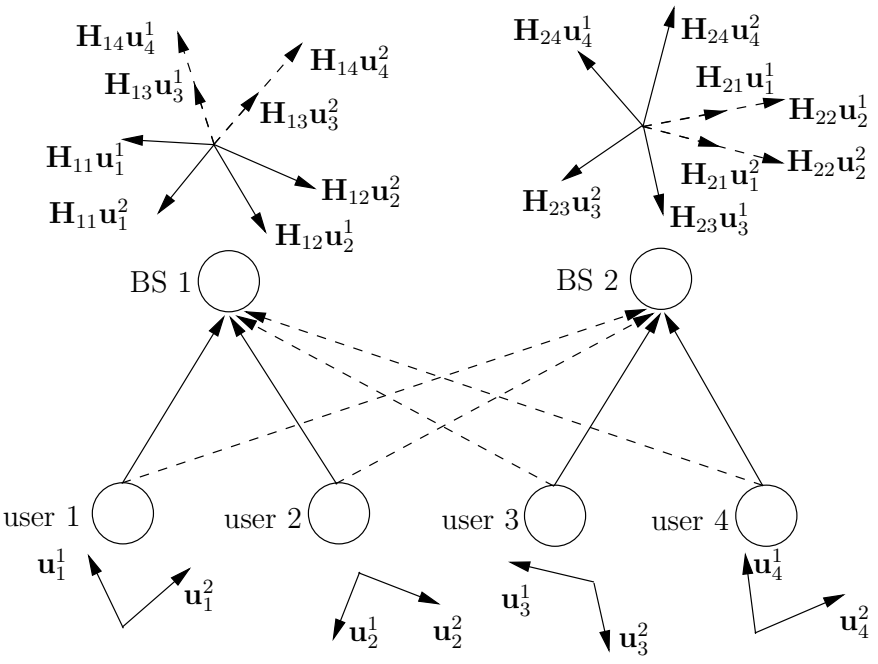
An Example. Consider a small cellular network with two BSs and four users in Fig. 5.2. Each user and BS have a single antenna. Suppose the uplink/downlink channels of the two BSs are synchronized as in [84]. We consider the uplink case in this example.[1] In Fig. 5.2, a solid arrow line represents a directed link and a dashed arrow line represents a directed interference.

To show the benefits of IA, we start with a simple example by assuming $K = 3$. Note

[1]The downlink case can be done similarly, as we will show in Section 5.8.



(a) Three-subcarrier case



(b) Six-subcarrier case

Figure 5.2: An example of IA in the spectral domain.

that we take $K = 3$ only for ease of illustration and we will consider a larger value of K (as in a practical system) later in the example. With $K = 3$, we show that by using IA, a total of four data streams can be transmitted from the users to their BSs, with 1 data stream from each user.

To show this, we need some notation. For vectors $\mathbf{a}$ and $\mathbf{b}$, denote $\mathbf{a} := \mathbf{b}$ if there exists a nonzero complex number c such that $\mathbf{a} = c\mathbf{b}$, i.e., $\mathbf{a}$ and $\mathbf{b}$ are in the same direction. Denote $\{\mathbf{u}_{\mathrm{b}}^1 \cdots \mathbf{u}_{\mathrm{b}}^K\}$ as a set of linearly independent basis vectors with dimension K and nonzero entries. Denote $\mathbf{H}_{ji}$ as the (frequency-domain) channel matrix between user i to BS j. Denote $\mathbf{u}_i^k$ as the encoding vector for the k-th outgoing stream at user i. We construct the encoding vectors at user 1 and user 2 as follows: let $\mathbf{u}_1^1 := \mathbf{u}_{\mathrm{b}}^1$ and let $\mathbf{u}_2^1 := \mathbf{H}_{22}^{-1}\mathbf{H}_{21}\mathbf{u}_1^1$. As a result, at BS 2, the interfering stream from user 1 is aligned to the interfering stream from user 2, as shown in Fig. 5.2. Likewise, we construct the encoding vectors at user 3 and user 4 as follows: let $\mathbf{u}_3^1 := \mathbf{u}_{\mathrm{b}}^2$ and let $\mathbf{u}_4^1 := \mathbf{H}_{24}^{-1}\mathbf{H}_{23}\mathbf{u}_3^1$. Then at BS 1, the interfering stream from user 3 is aligned to the interfering stream from user 4, as shown in Fig. 5.2. By using the above encoding vectors at the four users, the received data and interfering streams at each BS are on three different directions, indicating that three subcarriers are sufficient to support four data streams. However, when IA is not used, three subcarriers can support at most three data streams from the four users (with any combinations), since putting more than one data stream on a subcarrier will have interference on that subcarrier.

Now we increase the total number of subcarriers in the network to 6, i.e., $K = 6$. We show that by using IA, 8 data streams can be transmitted from the users to their BSs, with 2 data stream at each user. We construct the encoding vectors at user 1 and user 2 as follows: let $[\mathbf{u}_1^1\ \mathbf{u}_1^2] := [\mathbf{u}_{\mathrm{b}}^1\ \mathbf{u}_{\mathrm{b}}^2]$ and let $[\mathbf{u}_2^1\ \mathbf{u}_2^2] := \mathbf{H}_{22}^{-1}\mathbf{H}_{21}[\mathbf{u}_1^1\ \mathbf{u}_1^2]$. As a result, at BS 2, the two interfering streams from user 1 are aligned to the two interfering streams from user 2, as shown in Fig. 5.2(b). Likewise, we construct the encoding vectors at user 3 and user 4 as follows: let $[\mathbf{u}_3^1\ \mathbf{u}_3^2] := [\mathbf{u}_{\mathrm{b}}^3\ \mathbf{u}_{\mathrm{b}}^4]$ and let $[\mathbf{u}_4^1\ \mathbf{u}_4^2] := \mathbf{H}_{24}^{-1}\mathbf{H}_{23}[\mathbf{u}_3^1\ \mathbf{u}_3^2]$. As a result, at BS 1, the two interfering streams from users 3 are aligned to the two interfering streams from user 4, as shown in Fig. 5.2(b). By using those encoding vectors at the 4 users, the received data and

interfering streams at each BS are on 6 directions, indicating that 6 subcarriers are sufficient to support 8 data streams. However, when IA is not used, 6 subcarriers can support at most 6 data streams from the four users (with any combinations), since putting more than one data stream on a subcarrier will lead to interference.

When the total number of subcarriers in the network is increased to 12, i.e., $K = 12$. Following the same token, by using IA, 16 data streams can be transmitted from the 4 users to the BSs (with 4 data streams from each user). In comparison, when IA is not used, at most 12 data streams can be transmitted from the four users to their BSs. When K is increased to 1024. With IA, at least 1364 data streams can be transmitted from the four users to the BSs (with 341 from each user). In comparison, when IA is not used, at most 1024 data streams can be transmitted.

5.3 Problem Statement and Challenges

5.3.1 Goals and Problem Statement

The goal of this chapter is to exploit the benefits of spectral-domain (frequency-domain) IA for increasing user throughput in a cellular network. Instead of following an information theoretic approach as in [84], we are interested in addressing more practical problems, which we contrast as follows:

- The number of users served by each BS can be arbitrary (instead of being equal).
- A user's transmitter only interferes with those BSs' receivers within its interference range (rather than all BSs' receivers).
- The number of data streams at each user can vary, depending on application requirements (rather than being identical).
- The number of total available subcarriers is limited, e.g., 256, 1024, or 2048 based on cellular network standard (rather than going to infinity).

Note that the consideration of these practical issues is an important step to bridge the gap between results in IT to practical problems for future cellular networks. As such, new

mathematical models and optimization problems need to be developed.

We consider a cellular network consisting of a set of BSs and a set of users (see e.g., Fig. 5.6). A user may fall into the service area of multiple BSs but can only choose one BS as its service provider. For the uplink, a user may interfere with those BSs in its interference range that are not its home BS. Likewise, for the downlink, a BS may also interfere with those users who are not served by this BS. Our objective is to exploit IA in the spectral (frequency) domain so as to maximize the uplink/downlink minimum throughput among all the users in a cellular network.

5.3.2 Challenges

Given that the uplink and downlink are structured separately in a frame, we study the above throughput maximization problem for uplink and downlink separately. As it turns out, our results for the uplink problem can be easily carried to the downlink problem, as we shall show in Section 5.8. So for now we focus on the uplink problem.

We identify a number of challenges in the uplink user throughput maximization problem as follows:

- **IA scheme.** How to perform IA at the users is not a trivial problem, as the signal alignment behavior from a user's data stream is different at different BSs. Therefore, at each user (transmitter), one needs to determine which subset of its interfering streams should be selected for alignment and at which BS within its interference range. Further, one needs to design an alignment scheme at each BS so that the desired data streams are resolvable at all BSs while the undesirable data streams can be aligned to some predefined directions whenever possible.
- **IA Feasibility.** While designing an IA scheme, one needs to ensure its feasibility at the physical layer. This is not a trivial problem either and an IA scheme can only be validated by a feasibility proof. An improperly designed IA scheme may be shown to be infeasible at the physical layer. To prove feasibility of an IA scheme, one needs to show that there exist an encoding vector (at a user) and a decoding vector (at the BS) for

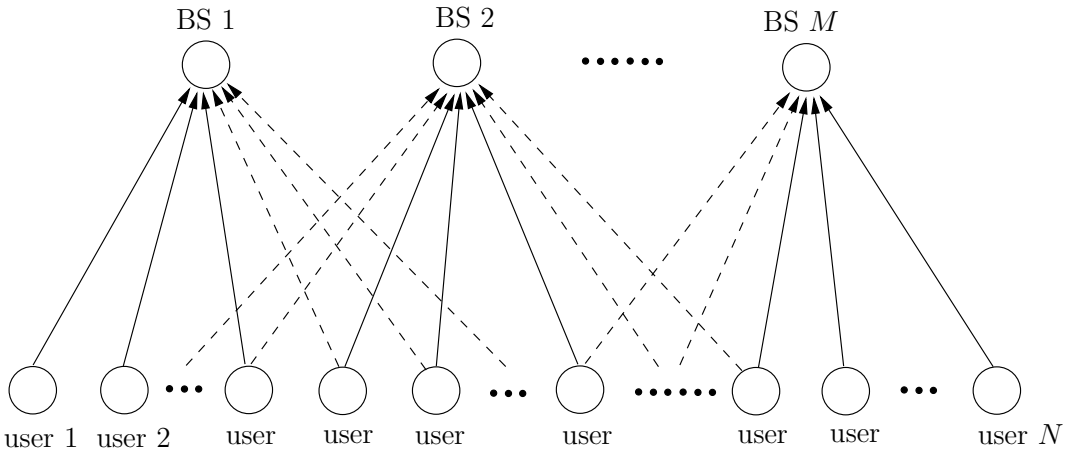


Figure 5.3: The uplink transmission in a cellular network.

each data stream such that all of the data streams in the network can be transported free of interference.

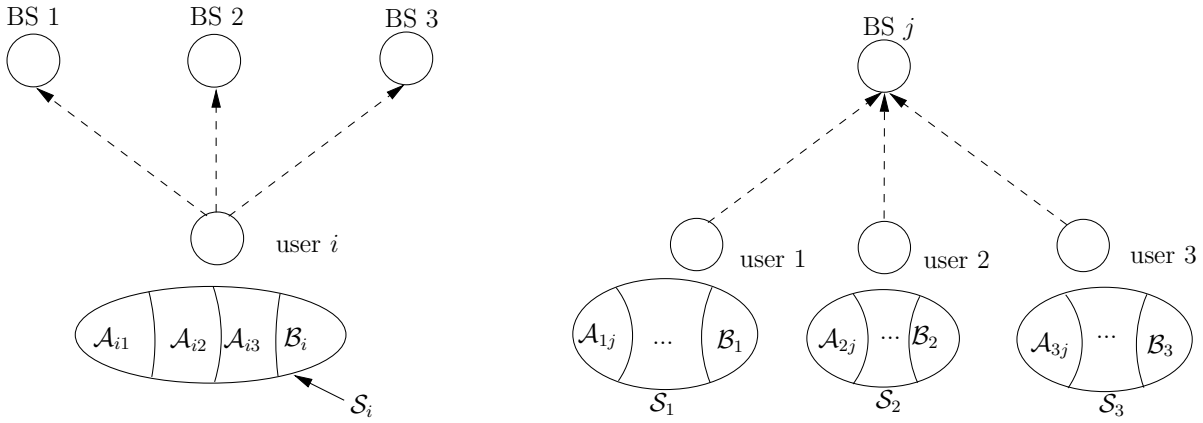
- **BS Selection.** In a cellular network, a user may fall into the service area of multiple BSs. Since an IA scheme is tightly coupled with a user's choice of a BS, making an optimal choice of a BS (so as to maximize the objective function) is not an easy problem.

5.4 An IA Scheme and Its Feasibility

In this section, we develop an IA scheme for the uplink communication in a cellular network. The IA scheme includes IA constraints at each user and BS, as well as how to construct the encoding/decoding vectors for each stream. In Section 5.4.1, we present such an IA scheme. In Section 5.4.2, we give a feasibility proof of this IA scheme at the physical layer.

5.4.1 An IA Scheme

Consider a cellular network shown in Fig. 5.3. Each of BSs and users has a single antenna. Denote $\mathcal{N}$ as the set of all users in the network and N as its cardinality (i.e., $N = |\mathcal{N}|$). Denote $\mathcal{M}$ as the set of all BSs in the network and M as its cardinality (i.e., $M = |\mathcal{M}|$).



(a) At user i (b) At BS j

Figure 5.4: An example that illustrates IA constraints at user i and BS $j \in \mathcal{I}_i^{\text{bs}}$.

Denote $\mathcal{T}_j^{\text{usr}}$ as the set of users who choose BS j as their service provider. Denote $\mathcal{I}_j^{\text{usr}}$ as the set of users that interfere with BS j, i.e., BS j is within the interference range of these users and BS j is not the service provider of these users. Denote $\mathcal{I}_i^{\text{bs}}$ as the set of BSs that are interfered with by user i, i.e., these BSs are within the interference range of user i but are not chosen by user i as its service provider.

Suppose that user i is interfering with BS j, i.e., $j \in \mathcal{I}_i^{\text{bs}}$. Denote $\mathcal{S}_i = \{s_i^k : 1 \le k \le \sigma_i\}$ as the set of streams from user i, where s_i^k is the k-th stream and σ_i is the number of streams in set $\mathcal{S}_i$. Then each stream in $\mathcal{S}_i$ is an interfering stream for BS j. At BS j, we hope that as many interfering streams from user i can be aligned to some predefined interference directions as possible.

Among the interfering streams in $\mathcal{S}_i$, denote $\mathcal{A}_{ij}$ as the subset of interfering streams that can be aligned to some predefined interference directions at BS j. Denote α_{ij} as the cardinality of $\mathcal{A}_{ij}$, i.e., $\alpha_{ij} = |\mathcal{A}_{ij}|$. Among the streams in $\mathcal{S}_i$, there may be a subset $\mathcal{B}_i$ of streams that are not aligned to any predefined interference direction at all BSs in $\mathcal{I}_i^{\text{bs}}$. Denote β_i as the cardinality of $\mathcal{B}_i$, i.e., $\beta_i = |\mathcal{B}_i|$. Thus we have

$$\mathcal{B}_i = \mathcal{S}_i \backslash (\cup_{j \in \mathcal{I}_i^{\text{bs}}} \mathcal{A}_{ij}).$$

Now we present an IA scheme, which includes three constraints and feasible encod-

ing/decoding vectors.

User Constraints. At user i (see Fig. 5.4(a) for example), there are σ_i outgoing streams, each of which is an interfering stream to all BSs in $\mathcal{I}_i^{\text{bs}}$. For each interfering stream $s_i^k \in \mathcal{S}_i$, we will construct a feasible encoding vector for this stream so that it is successfully aligned at most at *one* BS in $\mathcal{I}_i^{\text{bs}}$ (see Section 5.4.2). That is, our encoding vector is only required to guarantee the alignment of a stream at one BS. Based on this requirement, we define $\mathcal{A}_{ij_1} \cap \mathcal{A}_{ij_2} = \emptyset$ ($j_1, j_2 \in \mathcal{I}_i^{\text{bs}}$, $j_1 \neq j_2$). Therefore, we have following constraints at user i:

$$\beta_i + \sum_{j \in \mathcal{I}_i^{\text{bs}}} \alpha_{ij} = \sigma_i, \quad \text{for } i \in \mathcal{N}. \tag{5.1}$$

BS Constraints. At BS j (see Fig. 5.4(b) for example), we need to align the interfering streams in $\mathcal{A}_{ij}$ (for each $i \in \mathcal{I}_j^{\text{usr}}$) to some predefined interference directions. To do this, one must answer two questions: (i) what should be the set of predefined interference directions at BS j; (ii) how to align the interfering streams in $\mathcal{A}_{ij}$ to the set of predefined interference directions.

There may be many possible solutions to the above two questions. Here, we show one solution for which we can offer a feasibility proof at physical layer (see Section 5.4.2). In our solution, for the first question, we use $\cup_{i \in \mathcal{I}_j^{\text{usr}}} \mathcal{B}_i$ as the set of predefined interference directions at BS j. That is, each interfering stream in $\mathcal{A}_{ij}$ will be aligned to an interfering stream in $\cup_{i \in \mathcal{I}_j^{\text{usr}}} \mathcal{B}_i$. For the second question, we align each interfering streams in $\mathcal{A}_{ij}$ for each $i \in \mathcal{I}_j^{\text{usr}}$ to a unique interference stream in $\cup_{k \in \mathcal{I}_j^{\text{usr}}}^{k \neq i} \mathcal{B}_k$. That is, each interfering stream in $\mathcal{A}_{ij}$ is aligned uniquely in the interference subspace formed by the union of $\mathcal{B}_k$, $k \in \mathcal{I}_j^{\text{usr}}$ except its own $\mathcal{B}_i$. Here, "uniquely" refers that any two interfering streams in $\mathcal{A}_{ij}$ will not be aligned to the same interfering stream in $\cup_{k \in \mathcal{I}_j^{\text{usr}}}^{k \neq i} \mathcal{B}_k$. Based on our proposed solution to questions (i) and (ii), we have the following constraints at BS j:

$$\alpha_{ij} \leq \sum_{k \in \mathcal{I}_j^{\text{usr}}}^{k \neq i} \beta_k, \quad \text{for } i \in \mathcal{I}_j^{\text{usr}}, j \in \mathcal{M}. \tag{5.2}$$

Dimension Constraints. At BS j, the total number of its desired data streams is $\sum_{i \in \mathcal{I}_j^{\text{usr}}} \sigma_i$, while the number of its unaligned interfering streams is $\sum_{i \in \mathcal{I}_j^{\text{usr}}} (\sigma_i - \alpha_{ij})$. Since

the number of directions for desired data streams and unaligned interfering streams cannot exceed the number of available subcarriers, we have the following constraints at BS j:

$$\sum_{i\in\mathcal{T}_j^{\text{usr}}} \sigma_i + \sum_{i\in\mathcal{I}_j^{\text{usr}}} (\sigma_i - \alpha_{ij}) \leq K \quad \text{for } j \in \mathcal{M} \,. \tag{5.3}$$

In the following section, we show the feasibility of this IA scheme at the physical layer by constructing an encoding/decoding vector for each data stream.

5.4.2 Feasibility of the IA Scheme

Denote $\mathbf{H}_{ji}$ as the channel matrix between user i and BS j over K subcarriers. $\mathbf{H}_{ji}$ is a diagonal complex matrix with the k-th diagonal entry representing the channel coefficient of the k-th subcarrier. For each stream s_i^k, denote $\mathbf{u}_i^k \in \mathbb{C}^{K\times 1}$ as its encoding vector at user i and $\mathbf{v}_j^l \in \mathbb{C}^{K\times 1}$ as its decoding vector at its intended BS j. Denote π as an IA scheme that meets the constraints in (5.1), (5.2), and (5.3), with corresponding encoding vector $\mathbf{u}_i^k$ and decoding vector $\mathbf{v}_j^l$ for each stream s_i^k ($i \in \mathcal{N}$, $1 \leq k \leq \sigma_i$).

To recover data stream s_i^k at BS j (zero-forcing), decoding vector $\mathbf{v}_j^l$ should be able to filter out interfering streams from two types of users. The first type is the interfering users who do not choose BS j as their service provider, i.e., for $i' \in \mathcal{I}_j^{\text{usr}}$, $(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0$ holds for $1 \leq k' \leq \sigma_{i'}$. The second type is the users who choose BS j as their service provider, i.e., for $i' \in \mathcal{T}_j^{\text{usr}}$ and $(i', k') \neq (i, k)$, $(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0$ holds for $1 \leq k' \leq \sigma_{i'}$ More formally, we have the following definition:

Definition 2. *An IA scheme π is feasible at the physical layer if for $i' \in (\mathcal{T}_j^{\text{usr}} \cup \mathcal{I}_j^{\text{usr}})$ and $(i', k') \neq (i, k)$,*

$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji} \mathbf{u}_i^k = 1, \tag{5.4}$$

$$(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0, \tag{5.5}$$

hold for $1 \leq k' \leq \sigma_{i'}$.

The following theorem is the main result of this subsection:

Theorem 5. *There exists at least one set of encoding and decoding vectors such that IA scheme π is feasible at the physical layer.*

The rest of this section will be devoted to a proof of this theorem. Here is our road map. The proof is based on construction. First, we describe an encoding vector for each stream. Then we give two lemmas characterizing the dimensions of such encoding vectors. Based on these lemmas, we show that there always exists an decoding vector for each stream such that constraints (5.4) and (5.5) in Definition 2 are satisfied.

Encoding Vector Construction. Denote $\mathcal{E}^{S_i} = \{\mathbf{u}_i^k : 1 \leq k \leq \sigma_i\}$ as the set of encoding vectors for the set of streams $\mathcal{S}_i$ at user i. Among the encoding vectors in $\mathcal{E}^{S_i}$, denote $\mathcal{E}^{A_{ij}}$ as the subset of encoding vectors that correspond to the interfering streams in $\mathcal{A}_{ij}$; denote $\mathcal{E}^{B_i}$ as the subset of encoding vectors that correspond to the interfering streams in $\mathcal{B}_i$. Since we define a unique encoding vector for each stream, we have

$$
\begin{aligned}
&|\mathcal{E}^{A_{ij}}| = \alpha_{ij} \text{ for } j \in \mathcal{M}, i \in \mathcal{I}_j^{\text{usr}};\\
&|\mathcal{E}^{B_i}| = \beta_i \text{ for } i \in \mathcal{N};\\
&\mathcal{E}^{B_i} = \mathcal{E}^{S_i} \backslash (\cup_{j \in \mathcal{I}_i^{\text{bs}}} \mathcal{E}^{A_{ij}}) \text{ for } i \in \mathcal{N};\\
&\mathcal{E}^{A_{ij_1}} \cap \mathcal{E}^{A_{ij_2}} = \emptyset, \text{ for } i \in \mathcal{N}, j_1, j_2 \in \mathcal{I}_i^{\text{bs}}, j_1 \neq j_2.
\end{aligned}
$$

We define $\mathcal{E}^A = \cup_{i \in \mathcal{N}, j \in \mathcal{I}_i^{\text{bs}}} \mathcal{E}^{A_{ij}}$ and $\mathcal{E}^B = \cup_{i \in \mathcal{N}} \mathcal{E}^{B_i}$. Then we have $\cup_{i \in \mathcal{N}} \mathcal{E}^{S_i} = \mathcal{E}^A \cup \mathcal{E}^B$. We first construct the encoding vectors in $\mathcal{E}^B$ and then construct the encoding vectors in $\mathcal{E}^A$.

Denote $\{\mathbf{u}_{\text{b}}^k : 1 \leq k \leq K\}$ as a set of *linearly independent* complex vectors with dimension $K \times 1$ and *nonzero* entries. Then, for each $\mathbf{u}_i^k \in \mathcal{E}^B$, we define

$$\mathbf{u}_i^k := \mathbf{u}_{\text{b}}^k \,. \tag{5.6}$$

Now we construct the encoding vectors in $\mathcal{E}^A$. Recall that in IA scheme π, each interfering stream in $\mathcal{A}_{ij}$ is aligned to an interfering stream in $\cup_{k \in \mathcal{I}_j^{\text{usr}}}^{k \neq i} \mathcal{B}_k$. Therefore, for each $\mathbf{u}_i^k \in \mathcal{E}^A$, we define

$$\mathbf{u}_i^k := \mathbf{H}_{ji}^{-1} \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} \,. \tag{5.7}$$

where $\mathbf{u}_{i'}^{k'}$ is an encoding vector in $\mathcal{E}^B$ (i.e., $\mathbf{u}_{i'}^{k'} := \mathbf{u}_{\mathrm{b}}^{k'}$) and $i' \neq i$.

Encoding Vector Properties. Denote $\dim(\mathcal{E}^{S_i})$ as the dimension of the sub-space spanned by the vectors in set $\mathcal{E}^{S_i}$. Then we have the following lemma:

Lemma 9. *At each user $i \in \mathcal{N}$, the constructed encoding vectors $\mathcal{E}^{S_i}$ are linearly independent, i.e.,* $\dim(\mathcal{E}^{S_i}) = |\mathcal{E}^{S_i}|$.

Proof. Based on the definitions of $\mathcal{E}^{S_i}$, $\mathcal{E}^{B_i}$, and $\mathcal{E}^{A_{ij}}$, we have $\mathcal{E}^{S_i} = \mathcal{E}^{B_i} \cup (\cup_{j\in\mathcal{I}_i^{\mathrm{bs}}}\mathcal{E}^{A_{ij}})$. According to the encoding vector construction procedure, we know that the constructed encoding vectors in $\mathcal{E}^{B_i}$ are independent of any channel matrices (see (5.6)), while the constructed encoding vectors in $\mathcal{E}^{A_{ij}}$ are dependent on the channel matrices (see (5.7)). Given that the diagonal entries in the channel matrices are drawn from complex Gaussian distribution, we have

$$\dim(\mathcal{E}^{S_i}) = \dim(\mathcal{E}^{B_i} \cup (\cup_{j\in\mathcal{I}_i^{\mathrm{bs}}}\mathcal{E}^{A_{ij}})) = \dim(\mathcal{E}^{B_i}) + \dim(\cup_{j\in\mathcal{I}_i^{\mathrm{bs}}}\mathcal{E}^{A_{ij}}). \tag{5.8}$$

According to (5.7), we know that the encoding vectors in $\mathcal{E}^{A_{ij}}$ is determined by the channel matrix $\mathbf{H}_{ji}$. Since the channel matrices in $\{\mathbf{H}_{ji} : j \in \mathcal{I}_i^{\mathrm{bs}}\}$ are randomly independent of each other, we conclude that

$$\dim(\cup_{j\in\mathcal{I}_i^{\mathrm{bs}}}\mathcal{E}^{A_{ij}}) = \sum_{j\in\mathcal{I}_i^{\mathrm{bs}}} \dim(\mathcal{E}^{A_{ij}}). \tag{5.9}$$

To analyze $\dim(\mathcal{E}^{A_{ij}})$, we divide the encoding vectors in $\mathcal{E}^{A_{ij}}$ into groups based on the their corresponding value of i' in (5.7): $\{\mathcal{E}^{A_{ij}^{i'}} : i' \in \mathcal{I}_j^{\mathrm{usr}}, i' \neq i\}$. Thus we have $\mathcal{E}^{A_{ij}} = \cup_{i'\in\mathcal{I}_j^{\mathrm{usr}}}^{i'\neq i}\mathcal{E}^{A_{ij}^{i'}}$, where $\mathcal{E}^{A_{ij}^{i'}} := \mathbf{H}_{ji}^{-1}\mathbf{H}_{ji'}\mathcal{E}^{\tilde{B}_{i'}}$ with $\mathcal{E}^{\tilde{B}_{i'}} \subseteq \mathcal{E}^{B_{i'}}$. Based on the encoding vector construction procedure, we have

$$\dim(\mathcal{E}^{A_{ij}^{i'}}) \overset{(a)}{=} \dim(\mathcal{E}^{B_{i'}}) \overset{(b)}{=} |\mathcal{E}^{B_{i'}}| \overset{(c)}{=} |\mathcal{E}^{A_{ij}^{i'}}|, \tag{5.10}$$

where (a) holds due to full rank of channel matrices [36]; (b) holds due to the fact that the encoding vectors in $\mathcal{E}^{B_{i'}}$ are constructed by (5.6); (c) holds due to the fact that in our IA scheme, each interfering streams in $\mathcal{A}_{ij}$ is aligned to a unique interfering stream in $\mathcal{B}_{i'}$ with $i' \neq i$.

Based on the definitions and (5.10), we have

$$\dim(\mathcal{E}^{A_{ij}}) = \dim(\cup_{i' \in \mathcal{I}_j^{\text{usr}}}^{i' \neq i} \mathcal{E}^{A_{ij}^{i'}}) \overset{(a)}{=} \sum_{i' \in \mathcal{I}_j^{\text{usr}}}^{i' \neq i} \dim(\mathcal{E}^{A_{ij}^{i'}}) = \sum_{i' \in \mathcal{I}_j^{\text{usr}}}^{i' \neq i} |\mathcal{E}^{A_{ij}^{i'}}| = |\mathcal{E}^{A_{ij}}|, \tag{5.11}$$

where (a) holds due to the fact that the channel matrices $\{\mathbf{H}_{ji'} : i' \in \mathcal{I}_j^{\text{usr}}, i' \neq i\}$ are drawn complex Gaussian distribution and independent of each other.

Based on (5.8), (5.9), and (5.11), we conclude

$$\begin{aligned}\dim(\mathcal{E}^{S_i}) &= \dim(\mathcal{E}^{B_i}) + \dim(\cup_{j \in \mathcal{I}_i^{\text{bs}}} \mathcal{E}^{A_{ij}}) = |\mathcal{E}^{B_i}| + \sum_{j \in \mathcal{I}_i^{\text{bs}}} \dim(\mathcal{E}^{A_{ij}}) \\ &= |\mathcal{E}^{B_i}| + \sum_{j \in \mathcal{I}_i^{\text{bs}}} |\mathcal{E}^{A_{ij}}| = |\mathcal{E}^{S_i}|.\end{aligned}$$

This completes the proof. □

At BS j, denote $\mathcal{Q}_j^{\text{T}}$ as the set of its directions for its desired data streams and $\mathcal{Q}_j^{\text{I}}$ as the set of directions for its interfering streams. We have

$$\begin{aligned}\mathcal{Q}_j^{\text{T}} &= \cup_{i \in \mathcal{T}_j^{\text{usr}}} \{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{S_i}\}, \\ \mathcal{Q}_j^{\text{I}} &= \cup_{i \in \mathcal{I}_j^{\text{usr}}} \{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{S_i}\}.\end{aligned}$$

Then, we have the following lemma:

Lemma 10. *At each BS $j \in \mathcal{M}$, each of its desired data stream occupies an independent direction, i.e.,*

$$\dim(\mathcal{Q}_j^{\text{T}} \cup \mathcal{Q}_j^{\text{I}}) = \sum_{i \in \mathcal{T}_j^{\text{usr}}} \sigma_i + \dim(\mathcal{Q}_j^{\text{I}}), \text{ for } j \in \mathcal{M}. \tag{5.12}$$

Proof. At BS j (see e.g., Fig. 5.4(b)), denote $\mathcal{Q}_j^{\text{I,Ali}}$ as the set of aligned interfering stream directions; denote $\mathcal{Q}_j^{\text{I,Def}}$ as the set of predefined interfering stream directions; denote $\mathcal{Q}_j^{\text{I,Eff}}$ as the set of "effective" interfering stream directions. Then we have

$$\begin{aligned}\mathcal{Q}_j^{\text{I,Ali}} &= \cup_{i \in \mathcal{I}_j^{\text{usr}}} \{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{A_{ij}}\}, \\ \mathcal{Q}_j^{\text{I,Def}} &= \cup_{i \in \mathcal{I}_j^{\text{usr}}} \{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{B_i}\},\end{aligned}$$

$$\mathcal{Q}_j^{\mathrm{I,Eff}} = \cup_{i\in\mathcal{I}_j^{\mathrm{usr}}}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{S_i}\backslash\mathcal{E}^{A_{ij}}\}.$$

Since $\mathcal{E}^{B_i} \subseteq \mathcal{E}^{S_i}\backslash\mathcal{E}^{A_{ij}}$, we have $\mathcal{Q}_j^{\mathrm{I,Def}} \subseteq \mathcal{Q}_j^{\mathrm{I,Eff}}$. Based on the encoding vector construction procedure, we know that for each $\mathbf{H}_{ji}\mathbf{u}_i^k \in \mathcal{Q}_j^{\mathrm{I,Ali}}$, there exists a $\mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'} \in \mathcal{Q}_j^{\mathrm{I,Eff}}$ such that $\mathbf{H}_{ji}\mathbf{u}_i^k := \mathbf{H}_{ji'}\mathbf{u}_{i'}^{k'}$. Consequentially, we have $\mathrm{span}(\mathcal{Q}_j^{\mathrm{I,Ali}}) \subseteq \mathrm{span}(\mathcal{Q}_j^{\mathrm{I,Def}})$. Thus the following equation holds:

$$\mathrm{span}(\mathcal{Q}_j^{\mathrm{I}}) = \mathrm{span}(\mathcal{Q}_j^{\mathrm{I,Eff}} \cup \mathcal{Q}_j^{\mathrm{I,Ali}}) = \mathrm{span}(\mathcal{Q}_j^{\mathrm{I,Eff}}). \tag{5.13}$$

We now argue that the signal subspace $\mathcal{Q}_j^{\mathrm{T}}$ is linearly independent of the "effective" interference subspace $\mathcal{Q}_j^{\mathrm{I,Eff}}$ at BS j. We show it is true based on the following two points. First, based on the given constraint (5.3), we know that $|\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I,Eff}}| = \sum_{i\in\mathcal{T}_j^{\mathrm{usr}}}\sigma_i + \sum_{i\in\mathcal{I}_j^{\mathrm{usr}}}(\sigma_i - \alpha_{ij}) \leq K$. Thus, the number of directions in $\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I,Eff}}$ is bounded by the total available dimension (i.e., the number of subcarriers K). Second, the channel matrices $\{\mathbf{H}_{ji} : i \in \mathcal{T}_j^{\mathrm{usr}} \cup \mathcal{I}_j^{\mathrm{usr}}\}$ are frequency-selective and are randomly independent of each other. These properties of the channel matrices is attributive to the network environment. Based on the above two points, we can conclude that

$$\dim(\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I,Eff}}) = \dim(\mathcal{Q}_j^{\mathrm{T}}) + \dim(\mathcal{Q}_j^{\mathrm{I,Eff}}). \tag{5.14}$$

To characterize the dimension of the signal (desired data stream) subspace at BS j, we have

$$\begin{aligned}
\dim(\mathcal{Q}_j^{\mathrm{T}}) &= \dim(\cup_{i\in\mathcal{T}_j^{\mathrm{usr}}}\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{S_i}\}) \\
&\overset{(a)}{=} \sum_{i\in\mathcal{T}_j^{\mathrm{usr}}} \dim(\{\mathbf{H}_{ji}\mathbf{u}_i^k : \mathbf{u}_i^k \in \mathcal{E}^{S_i}\}) \\
&\overset{(b)}{=} \sum_{i\in\mathcal{T}_j^{\mathrm{usr}}} \dim(\mathcal{E}^{S_i}) \\
&\overset{(c)}{=} \sum_{i\in\mathcal{T}_j^{\mathrm{usr}}} \sigma_i,
\end{aligned} \tag{5.15}$$

where (a) holds due to the random independence of the channel matrices $\{\mathbf{H}_{ji} : i \in \mathcal{T}_j^{\mathrm{usr}}\}$ and $|\mathcal{Q}_j^{\mathrm{T}}| \leq K$; (b) holds due to the full rank of $\mathbf{H}_{ji}$ [36]; (c) holds due to Lemma 9 and $|\mathcal{E}^{S_i}| = \sigma_i$.

Therefore, we have

$$\dim(\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}}) \overset{(a)}{=} \dim(\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I,Eff}}) \overset{(b)}{=} \dim(\mathcal{Q}_j^{\mathrm{T}}) + \dim(\mathcal{Q}_j^{\mathrm{I,Eff}}) \overset{(c)}{=} \sum_{i \in \mathcal{T}_j^{\mathrm{usr}}} \sigma_i + \dim(\mathcal{Q}_j^{\mathrm{I}}), \quad (5.16)$$

where (a) and (c) hold due to (5.13); (b) holds due to (5.15).

Combining (5.16) and Lemma 1, we conclude that Theorem 5 holds. □

Decoding Vector. For the decoding vectors, we have the following proposition:

Proposition 1. *If the encoding vectors satisfy (5.12), then there exists a decoding vector for each stream such that constraints (5.4) and (5.5) are satisfied.*

Proof. We show that if the encoding vectors satisfy constraint (5.12), then there exist a set of decoding vectors that satisfy (5.4) and (5.5) in Definition 2. Specifically, we argue that if constraint (5.12) is satisfied, then the following linear system is consistent (i.e., the system has at least one feasible solution):

$$\begin{aligned}
&(\mathbf{v}_j^l)^T \mathbf{H}_{ji} \mathbf{u}_i^k = 1, \\
&(\mathbf{v}_j^l)^T \mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} = 0,\ 1 \le k' \le \sigma_{i'}, i' \in \mathcal{T}_j^{\mathrm{usr}} \cup \mathcal{I}_j^{\mathrm{usr}}, (i', k') \ne (i, k)
\end{aligned}$$

where $\mathbf{v}_j^l$ is variable vector while $\mathbf{H}$'s and $\mathbf{u}$'s are given.

Based on the definition of $\mathcal{Q}_j^{\mathrm{T}}$ and $\mathcal{Q}_j^{\mathrm{I}}$, we know

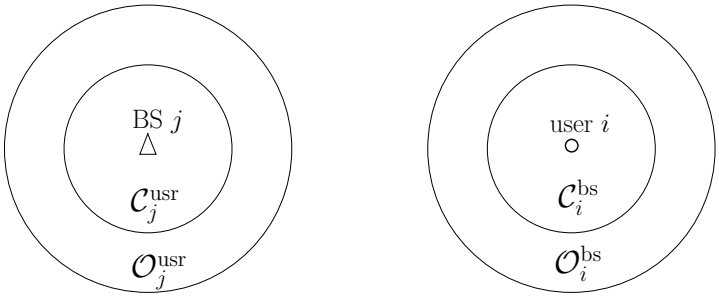
$$\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}} = \{\mathbf{H}_{ji'} \mathbf{u}_{i'}^{k'} : i' \in \mathcal{T}_j^{\mathrm{usr}} \cup \mathcal{I}_j^{\mathrm{usr}}, 1 \le k' \le \sigma_{i'}\}.$$

It is easy to see that $\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}}$ is the set of coefficient-vectors of this linear system. Moreover, this system has K free variables and at most K linearly independent equations. If we can show that vector $\mathbf{H}_{ji} \mathbf{u}_i^k$ is not a linear combination of other vectors in $\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}}$, then this system is consistent. Next, we argue this point by contradiction.

Suppose that $\mathbf{H}_{ji} \mathbf{u}_i^k$ is a linear combination of other vectors in $\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}}$. Since $\mathbf{H}_{ji} \mathbf{u}_i^k \in \mathcal{Q}_j^{\mathrm{T}}$, we have

$$\dim(\mathcal{Q}_j^{\mathrm{T}} \cup \mathcal{Q}_j^{\mathrm{I}}) < |\mathcal{Q}_j^{\mathrm{T}}| + \dim(\mathcal{Q}_j^{\mathrm{I}}) = \sum_{i \in \mathcal{I}_j^{\mathrm{usr}}} \sigma_i + \dim(\mathcal{Q}_j^{\mathrm{I}}).$$

This contradicts the given condition in (5.12). Therefore, we conclude that the linear system is consistent. □



(a) Sets for BS j. (b) Sets for user i

Figure 5.5: Sets illustration at BS j and user i.

This completes the proof of Theorem 5.

5.5 BS Selection and Impact on IA

As stated earlier, a user may be within the service area of multiple BSs and thus must choose one BS. For each user, how to choose a BS as its service provider is a part of our optimization problem.

As shown in Fig. 5.5(a), denote $\mathcal{C}_j^{\text{usr}}$ as the set of users within the service area of BS j; denote $\mathcal{O}_j^{\text{usr}}$ as the set of users that are outside the service area of BS j but can still interfere with BS j. As shown in Fig. 5.5(b), denote $\mathcal{C}_i^{\text{bs}}$ as the set of BSs that user i can choose as its service provider; denote $\mathcal{O}_i^{\text{bs}}$ as the set of BSs whose service areas do not cover user i but are still inside the interference range of user i.

BS Selection. Denote x_{ij} as a binary variable to indicate whether or not user i chooses BS $j \in \mathcal{C}_i^{\text{bs}}$ as its service provider, i.e., $x_{ij} = 1$ if user i chooses BS j and 0 otherwise. Since user i can choose at most one BS as its service provider, we have

$$\sum_{j \in \mathcal{C}_i^{\text{bs}}} x_{ij} = 1, \quad i \in \mathcal{N}. \tag{5.17}$$

Denote y_{ij} as the complementary binary variable of x_{ij}. That is, $y_{ij} = 1$ if user i does *not* choose BS $j \in \mathcal{C}_i^{\text{bs}}$ as its service provider and 0 otherwise. Then we have the following

constraints:

$$x_{ij} + y_{ij} = 1, \quad j \in \mathcal{C}_i^{\text{bs}}, i \in \mathcal{N}. \tag{5.18}$$

Impact on IA. We now show that the above BS selection variables can be incorporated into (5.1), (5.2), and (5.3) in our IA scheme.

- To incorporate BS selection variables in (5.1), we need to first clarify $\mathcal{I}_i^{\text{bs}}$, i.e., the set of BSs that are interfered with by user i. Based on the definitions of $\mathcal{O}_i^{\text{bs}}$, $\mathcal{C}_i^{\text{bs}}$, and y_{ij}, we have

  $$\mathcal{I}_i^{\text{bs}} = \mathcal{O}_i^{\text{bs}} \cup \{j : y_{ij} = 1, j \in \mathcal{C}_i^{\text{bs}}\}.$$

  Then, (5.1) can be re-written as:

  $$\beta_i + \sum_{j \in \mathcal{O}_i^{\text{bs}}} \alpha_{ij} + \sum_{j \in \mathcal{C}_i^{\text{bs}}} \alpha_{ij} \cdot y_{ij} = \sigma_i, \quad i \in \mathcal{N}. \tag{5.19}$$

- Likewise, for (5.2), we need to first clarify $\mathcal{I}_j^{\text{usr}}$, i.e., the set of users that are interfering with BS j. Based on the definitions of $\mathcal{O}_j^{\text{usr}}$, $\mathcal{C}_j^{\text{usr}}$, and y_{ij}, we have

  $$\mathcal{I}_j^{\text{usr}} = \mathcal{O}_j^{\text{usr}} \cup \{i : y_{ij} = 1, i \in \mathcal{C}_j^{\text{usr}}\}. \tag{5.20}$$

  Depending on whether user $i \in \mathcal{O}_j^{\text{usr}}$ or $i \in \mathcal{C}_j^{\text{usr}}$, (5.2) can be re-written as:

  $$\alpha_{ij} \leq \sum_{k \in \mathcal{O}_j^{\text{usr}}}^{k \neq i} \beta_k + \sum_{k \in \mathcal{C}_j^{\text{usr}}} \beta_k \cdot y_{kj}, \quad i \in \mathcal{O}_j^{\text{usr}}, j \in \mathcal{M}, \tag{5.21}$$

  $$\alpha_{ij} \cdot y_{ij} \leq \sum_{k \in \mathcal{O}_j^{\text{usr}}} \beta_k + \sum_{k \in \mathcal{C}_j^{\text{usr}}}^{k \neq i} \beta_k \cdot y_{kj}, \quad i \in \mathcal{C}_j^{\text{usr}}, j \in \mathcal{M}, \tag{5.22}$$

- Finally, for (5.3), we need to first clarify $\mathcal{T}_j^{\text{usr}}$, i.e., the set of users that choose BS j as their service provider. Based on the definitions of $\mathcal{C}_j^{\text{usr}}$ and x_{ij}, we have

  $$\mathcal{T}_j^{\text{usr}} = \{i : x_{ij} = 1, i \in \mathcal{C}_j^{\text{usr}}\}.$$

  Then, (5.3) can be re-written as:

  $$\sum_{i \in \mathcal{C}_j^{\text{usr}}} \sigma_i \cdot x_{ij} + \sum_{i \in \mathcal{I}_j^{\text{usr}}} (\sigma_i - \alpha_{ij}) \leq K \qquad j \in \mathcal{M}\ .$$

which is equivalent to

$$\sum_{i\in\mathcal{C}_j^{\text{usr}}} \sigma_i \cdot x_{ij} + \sum_{i\in\mathcal{C}_j^{\text{usr}}} (\sigma_i - \alpha_{ij}) \cdot y_{ij} + \sum_{i\in\mathcal{O}_j^{\text{usr}}} (\sigma_i - \alpha_{ij}) \leq K,\ j \in \mathcal{M}, \tag{5.23}$$

based on $\mathcal{I}_j^{\text{usr}}$ in (5.20).

5.6 User Throughput Maximization Problem

In this section, we employ the IA scheme in Section 5.4.1 to study a uplink user throughput maximization problem in a cellular network. For simplicity, we assume that fixed modulation and coding scheme (MCS) is used for each data stream and that each data stream corresponds to one unit data rate. The goal is to maximize the minimum rate among all the users. Denote $r_{\min}$ as the minimum rate among all users. Then we have:

$$\sigma_i \geq r_{\min}\ , \qquad i \in \mathcal{N}. \tag{5.24}$$

Based on the constraints in Section 5.5, the user throughput maximization problem can be formulated as follows:

OPT-IA$^{\text{raw}}$:	max	$r_{\min}$
	s.t.	BS selection: (5.17), (5.18);
		IA constraints: (5.19), (5.21), (5.22), (5.23);
		Minimum rate constraints: (5.24).

OPT-IA$^{\text{raw}}$ is a mixed integer nonlinear programming (MINLP). To eliminate the nonlinear terms in the constraints, we employ the Reformulation-Linearization Technique in [76].

To eliminate the nonlinear term $\alpha_{ij} \cdot y_{ij}$ in the constraints, we define $\lambda_{ij} = \alpha_{ij} \cdot y_{ij}$. This replacement requires to add the following two constraints:

$$0 \leq \lambda_{ij} \leq \alpha_{ij}, \quad j \in \mathcal{C}_i^{\text{bs}}, i \in \mathcal{N}, \tag{5.25}$$

$$\alpha_{ij} - (1 - y_{ij}) \cdot K \leq \lambda_{ij} \leq y_{ij} \cdot K, \quad j \in \mathcal{C}_i^{\text{bs}}, i \in \mathcal{N}. \tag{5.26}$$

Similarly, to eliminate the nonlinear term $\beta_i \cdot y_{ij}$ in the constraints, we define $\mu_{ij} = \beta_i \cdot y_{ij}$. This replacement requires to add the following two constraints:

$$0 \leq \mu_{ij} \leq \beta_i, \quad j \in \mathcal{C}_i^{\text{bs}}, i \in \mathcal{N}, \tag{5.27}$$

$$\beta_i - (1 - y_{ij}) \cdot K \leq \mu_{ij} \leq y_{ij} \cdot K, \quad j \in \mathcal{C}_i^{\text{bs}}, i \in \mathcal{N}. \tag{5.28}$$

By replacing $\lambda_{ij} = \alpha_{ij} \cdot y_{ij}$ and $\mu_{ij} = \beta_i \cdot y_{ij}$ in the IA constraints (5.19), (5.21), (5.22), (5.23), we have the following linear IA constraints:

$$\beta_i + \sum_{j \in \mathcal{O}_i^{\text{bs}}} \alpha_{ij} + \sum_{j \in \mathcal{C}_i^{\text{bs}}} \lambda_{ij} = \sigma_i, \quad i \in \mathcal{N}, \tag{5.29}$$

$$\alpha_{ij} \leq \sum_{k \in \mathcal{O}_j^{\text{usr}}}^{k \neq i} \beta_k + \sum_{k \in \mathcal{C}_j^{\text{usr}}} \mu_{kj}, \quad i \in \mathcal{O}_j^{\text{usr}}, j \in \mathcal{M}, \tag{5.30}$$

$$\lambda_{ij} \leq \sum_{k \in \mathcal{O}_j^{\text{usr}}} \beta_k + \sum_{k \in \mathcal{C}_j^{\text{usr}}}^{k \neq i} \mu_{kj}, \quad i \in \mathcal{C}_j^{\text{usr}}, j \in \mathcal{M}, \tag{5.31}$$

$$\sum_{i \in \mathcal{C}_j^{\text{usr}}} (\sigma_i - \lambda_{ij}) + \sum_{i \in \mathcal{O}_j^{\text{usr}}} (\sigma_i - \alpha_{ij}) \leq K, \quad j \in \mathcal{M}. \tag{5.32}$$

Then, OPT-IA$^{\text{raw}}$ is reformulated as follows:

OPT-IA:	Max	$r_{\min}$
	s.t.	BS selection: (5.17), (5.18);
		IA constraints: (5.25) – (5.32);
		Minimum rate constraints: (5.24);

where $\mathcal{N}$, $\mathcal{M}$, $\mathcal{C}_i^{\text{bs}}$, $\mathcal{O}_i^{\text{bs}}$, $\mathcal{C}_j^{\text{usr}}$, $\mathcal{O}_j^{\text{usr}}$, and K are known; x_{ij} and y_{ij} are binary variables; $r_{\min}$, σ_i, α_{ij}, β_i, λ_{ij}, and μ_{ij} are non-negative integer variables.

OPT-IA is a mixed integer linear programming (MILP). Although the theoretical worst-case complexity to a general MILP problem is exponential [19, 74], there exist highly efficient optimality/approximation algorithms (e.g., branch-and-bound with cutting planes [75]) and heuristics (e.g., sequential fixing algorithm [34, 32]). Another approach is to apply an off-the-shelf solver (CPLEX [104]), which can successfully handle a moderate-sized network. We will adopt this approach as it is sufficient to serve our purpose in the next section.

5.7 Performance Evaluation

In this section, we use a case study to illustrate how IA scheme works in a cellular network to maximize user throughput. We also compare the user throughput performance of our IA scheme to two other schemes: "no-IA" scheme and "crude-IA" scheme. In the no-IA scheme, a subset of subcarriers is allocated to each user for its data transmission such that at each BS, each data or interfering stream occupies a *unique* subcarrier. That is, there is a complete absence of overlapping of interfering streams on any subcarrier. We denote the user throughput maximization problem under the no-IA scheme as OPT-noIA. In the crude-IA scheme, a subset of subcarriers is allocated to each user for its data transmission such that at a BS, each of its desired data streams is on a unique subcarrier while the interfering streams are allowed to overlap. This problem is similar to ours except that each data stream in our IA scheme occupies *all* subcarriers and there is an optimization on the design of directions for intended data streams and interfering data streams. In light of this key difference, we denote the user throughput maximization problem under the crude-IA scheme as OPT-crudeIA. In what follows, we first formulate the two problems as performance benchmark for evaluating our IA scheme, and then offer our numerical results to demonstrate the throughput gain of IA.

5.7.1 Two Performance Benchmarks

Throughput Maximization under no-IA scheme. In the no-IA scheme, a subset of subcarriers is allocated to each user for its data transmission such that at each BS, each data or interfering stream occupies a *unique* subcarrier. That is, there is a complete absence of overlapping of interfering streams on any subcarrier. Denote $\mathcal{K}$ as the set of subcarriers in the network. Denote w_{ik} as a binary variable to indicate whether the k-th subcarrier is used by user i. Specifically, $w_{ik} = 1$ if the k-th subcarrier is used for data transmission at user i and $w_{ik} = 0$ otherwise. Thus, the number of outgoing streams from user i can be expressed

as

$$\sigma_i = \sum_{k \in \mathcal{K}} w_{ik}, \quad i \in \mathcal{N}. \tag{5.33}$$

At BS $j \in \mathcal{M}$, a subcarrier $k \in \mathcal{K}$ can be used by only one user within its transmission range and interference range. Otherwise, it will cause interference collision or overlapping. Thus, we have the following constraints.

$$\sum_{i \in \mathcal{T}_j \cup \mathcal{I}_j} w_{ik} \leq 1, \quad k \in \mathcal{K}, j \in \mathcal{M}. \tag{5.34}$$

Therefore, we can formulate the same problem in Section 5.6 as follows.

OPT-noIA:	max	$r_{\min}$
	s.t.	(5.24), (5.33), (5.34),

where w_{ik} and σ_i are variables, and $\mathcal{C}_j^{\text{usr}}$, $\mathcal{O}_j^{\text{usr}}$, $\mathcal{N}$, $\mathcal{M}$, $\mathcal{K}$ are known *a priori* based on the network topology and setting.

Throughput Maximization under Crude-IA Scheme. We formulate the same network throughput problem under the crude-IA scheme. In the crude-IA scheme, a subset of subcarriers is allocated to each user for its data transmission such that at each BS, each of its desired data streams is on a unique subcarrier while the interfering streams are allowed to overlap.

Denote $\mathcal{K}$ as the set of subcarriers in the network. Denote w_{ik} as a binary variable to indicate whether the k-th subcarrier is used at user i. Specifically, $w_{ik} = 1$ if the k-th subcarrier is used for data transmission at user i and $w_{ik} = 0$ otherwise. Thus, the number of outgoing streams from user i can be expressed as

$$\sigma_i = \sum_{k \in \mathcal{K}} w_{ik}, \quad i \in \mathcal{N}. \tag{5.35}$$

Consider a BS j and its serving users (i.e. users in $\mathcal{T}_j^{\text{usr}}$). To avoid transmission conflict, at most one of the users in $\mathcal{T}_j^{\text{usr}}$ can use the k-th subcarrier for data stream transmission. Thus we have $\sum_{i \in \mathcal{T}_j^{\text{usr}}} w_{ik} \leq 1$. If none of the users in $\mathcal{T}_j^{\text{usr}}$ uses the k-th subcarrier for data stream transmission, then this subcarrier can accommodate any amount of interference (i.e.,

$\frac{1}{N}\sum_{i\in\mathcal{I}_j^{\rm usr}} w_{ik} \le 1$). Combining these two cases, the interference avoidance scheme can be modeled by the following constraint.

$$\sum_{i\in\mathcal{T}_j^{\rm usr}} w_{ik} + \frac{1}{N}\sum_{i\in\mathcal{I}_j^{\rm usr}} w_{ik} \le 1, \quad j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.36}$$

Recall that $\mathcal{C}_j^{\rm usr}$ is the set of users within the transmission range of BS j and $\mathcal{O}_j^{\rm usr}$ is the set of users within the interference range of BS j. A user may be within the transmission range of multiple BSs and we use x_{ij} to indicate which BS is serving for it. Thus we have $\mathcal{T}_j^{\rm usr} = \{i : i\in\mathcal{C}_j^{\rm usr}, x_{ij}=1\}$ and $\mathcal{I}_j^{\rm usr} = \mathcal{O}_j^{\rm usr}\cup\{i : i\in\mathcal{C}_j^{\rm usr}, x_{ij}=0\}$. Then the interference avoidance constraints in the network can be expressed as

$$\sum_{i\in\mathcal{C}_j^{\rm usr}} x_{ij}w_{ik} + \frac{1}{N}\sum_{i\in\mathcal{C}_j^{\rm usr}}(1-x_{ij})w_{ik} + \frac{1}{N}\sum_{i\in\mathcal{O}_j^{\rm usr}} w_{ik} \le 1, \qquad j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.37}$$

To make the problem more tractable, we linearize the nonlinear term $x_{ij}w_{ik}$ in the interference avoidance constraint (5.37) by RLT. Specifically, we let

$$q_{ijk} = x_{ij}w_{ik}, \quad j\in\mathcal{M}, i\in\mathcal{C}_j^{\rm usr}, k\in\mathcal{K}. \tag{5.38}$$

Given that both x_{ij} and w_{ik} are binary variables, it it easy to verify that constraint (5.38) is equivalent to the combination of the following three linear constraints.

$$q_{ijk} \le x_{ij}, \quad i\in\mathcal{C}_j^{\rm usr}, j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.39}$$

$$q_{ijk} \le w_{ik}, \quad i\in\mathcal{C}_j^{\rm usr}, j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.40}$$

$$q_{ijk} \ge x_{ij} + w_{ik} - 1, \quad i\in\mathcal{C}_j^{\rm usr}, j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.41}$$

Replacing $x_{ij}w_{ik}$ by q_{ijk} in interference avoidance constraint (5.37), we have

$$\frac{N-1}{N}\sum_{i\in\mathcal{C}_j^{\rm usr}} q_{ijk} + \frac{1}{N}\sum_{i\in\mathcal{C}_j^{\rm usr}\cup\mathcal{O}_j^{\rm usr}} w_{ik} \le 1, \qquad j\in\mathcal{M}, k\in\mathcal{K}. \tag{5.42}$$

Therefore, we can formulate the same problem in Section 5.6 as follows.

OPT-crudeIA:	max	$r_{\min}$
	s.t.	(5.17), (5.24), (5.42), (5.39), (5.40), (5.41), (5.35),

where x_{ij}, w_{ik}, σ_i, q_{ijk}, and $r_{\min}$ are variables; while $\mathcal{C}_j^{\rm usr}$, $\mathcal{O}_j^{\rm usr}$, N, $\mathcal{N}$, $\mathcal{M}$, $\mathcal{K}$ are known *a priori* based on the network topology and setting.

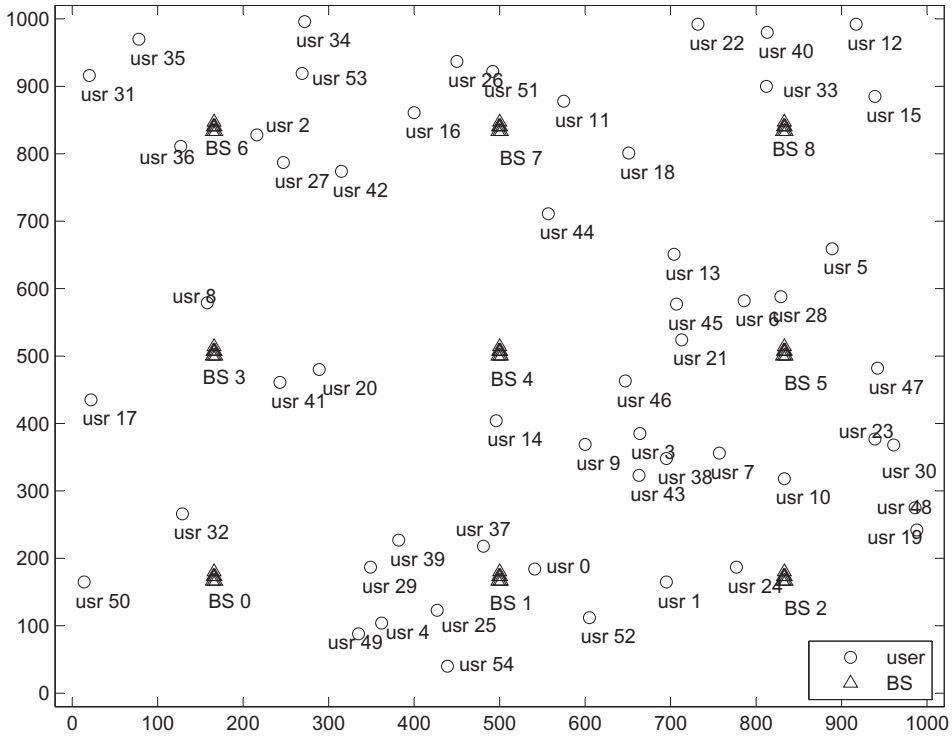


Figure 5.6: Network topology in the case study.

5.7.2 Simulation Setting

For ease of exposition, we normalize all units for distance, time, bandwidth, and data rate with appropriate dimensions. We consider a cellular network with 9 BSs and 100 users within a 1000×1000 area (see Fig. 5.6 for example). We divide the whole area into 9 equal-size grids and deploy the BSs at the center of the grids. The 100 users are randomly distributed in the area with a uniform probability. A user can be in "active" or "inactive" state, with equal probability. When "active", a user has a persistence traffic for transmission; when "inactive" a user is not served by any BS. For any comparison study (or "instance"), the state of a user is the same under all three schemes. We assume that the transmission range and interference range of a user are 240 and 480, respectively. The total number of subcarriers in the network is 256.

Before we present the extensive results over 100 network instances, we first show the results for one network instance with 55 active users in Fig. 5.6 (the 45 inactive users are not

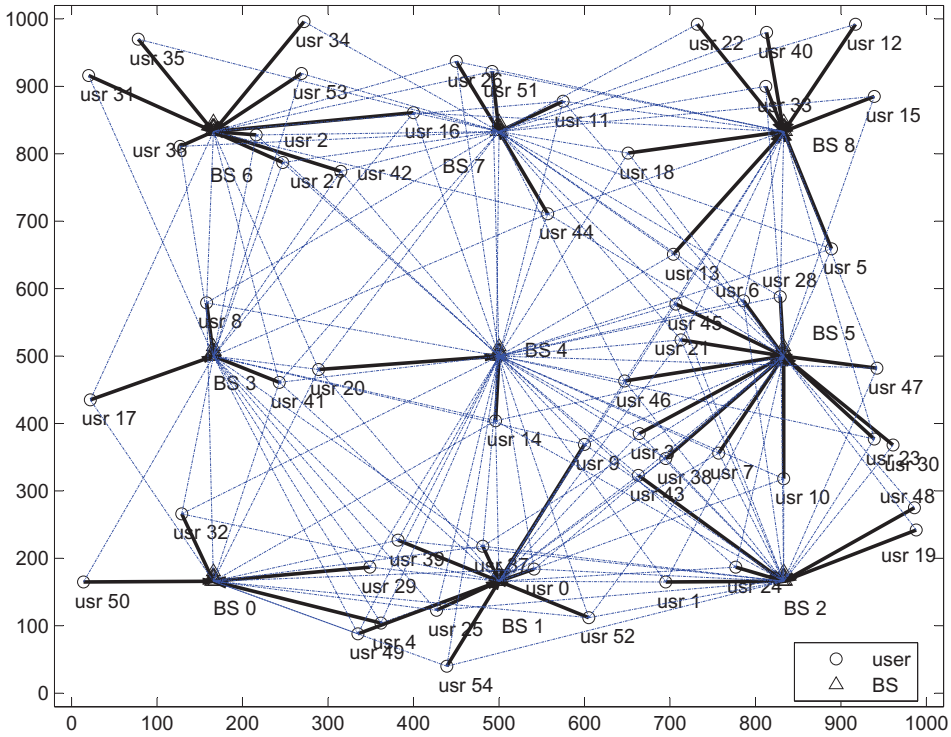


Figure 5.7: BS selection and interference of each user.

shown). By solving the OPT-IA problem for this network instance, we obtain the optimal objective value of 13. We then solve the OPT-noIA problem for this network instance, we obtain the optimal objective value of 6. This indicates that our IA scheme can increase the user throughput by 117% when compared to the no-IA scheme. We also solve the OPT-crudeIA problem for this network instance, we obtain the optimal objective value of 9. This indicates that our IA scheme can increase the user throughput by 44% when compared to the crude-IA scheme.

We now give some details in the solution to the OPT-IA problem. Fig. 5.7 shows the BS selection by each user and interference by the users on each BS. In this figure, a solid arrow line represents an established link from a user to a BS and a dashed line represents an interference. Table 5.2 summarizes the IA behavior at each BS. In this table, the first column lists the BSs in the network; the second column lists the number of users that choose this BS as their service provider; the third column lists the number of desired data streams at this

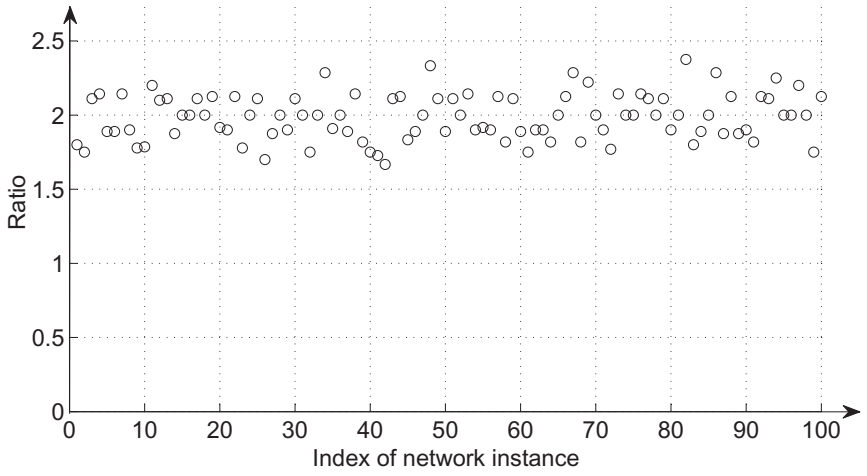
Table 5.2: IA behavior at each BS in the case study.

BS j	# of users	# of data streams	dimension of interfering streams	# of interfering streams	Interference overlapping ratio
BS 0	4	52	204	156	0.76
BS 1	8	104	152	208	1.37
BS 2	5	65	191	260	1.36
BS 3	3	39	217	247	1.14
BS 4	2	26	230	494	2.15
BS 5	12	156	100	221	2.21
BS 6	9	117	139	104	0.75
BS 7	4	52	204	325	1.59
BS 8	8	104	152	156	1.03

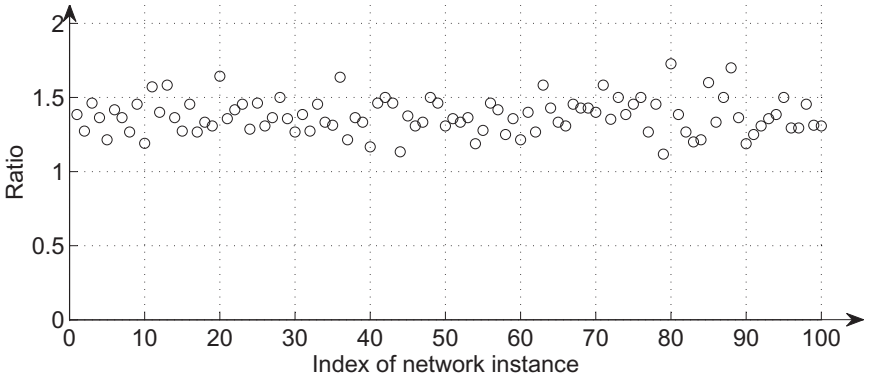
BS, where each user has 13 data streams to its BS; the fourth column lists the dimension of the subspace for the interfering streams at this BS, which is 256 minus the number in the third column; the fifth column lists the number of undesired interfering streams (from neighboring interfering users) at this BS; the sixth column lists the interference overlapping ratio, which is the ratio of the fifth column to the fourth column. In the sixth column, a value greater than 1 indicates the existence of interference overlapping. The large the ratio is, the more IA has been achieved at the corresponding BS.

5.7.3 A Case Study

Now let's take a look at the row for BS 5 in Table 5.2 as an example. As shown in Fig. 5.6, BS 5 is used as service provider by 12 users. Since each user has 13 outgoing data streams, the number of desired data streams at BS 5 is 156. Thus, the dimension of the subspace for the interfering streams is upper bounded by 100 ($= 256 - 156$). As shown in Fig. 5.6, BS 5 is being interfered by 17 users and thus has 221 ($= 17 \times 13$) interfering streams. Therefore, the interference overlapping ratio at BS 5 is $221/100 = 2.21$ (as shown in the table).



(a) Throughput gain of OPT-IA over OPT-noIA



(b) Throughput gain of OPT-IA over OPT-crudeIA

Figure 5.8: Throughput gain of OPT-IA over OPT-noIA and OPT-crudeIA.

5.7.4 Complete Results

We now present complete comparison results over 100 network instances, where each network instance represents a unique active/inactive behavior among the users. Figure 5.8(a) presents the throughput gain of OPT-IA over OPT-noIA and Figure 5.8(b) presents the throughput gain of OPT-IA over OPT-crudeIA. For both figures, x-axis is the index of network instance and y-axis is the throughput gain of OPT-IA (i.e., the ratio of the optimal objective value of OPT-IA to that of OPT-noIA or OPT-crudeIA). Simulation results from the 100 network instances show that our IA scheme has an average 98% throughput improvement over the

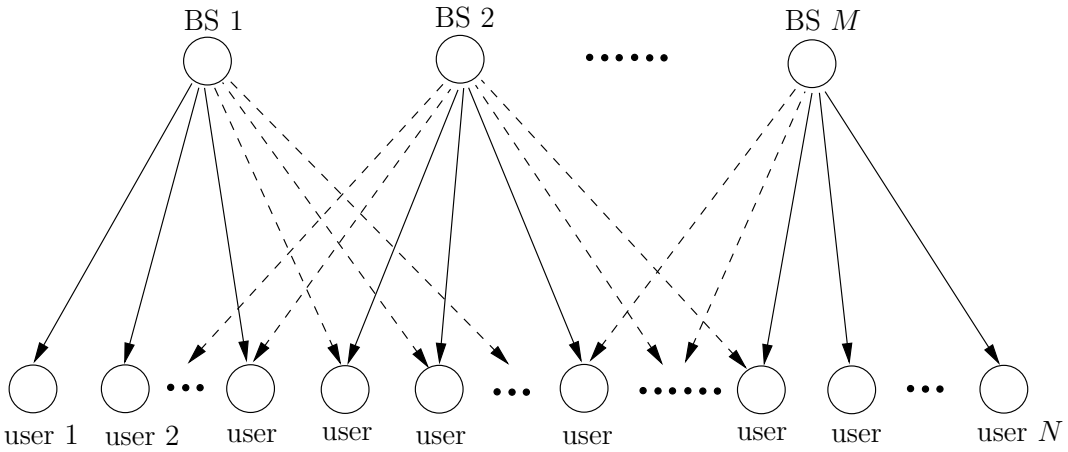


Figure 5.9: Downlink transmission in a cellular network.

no-IA scheme and an average 39% throughput improvement over the crude-IA scheme.

5.8 Downlink User Throughput Maximization

In Sections 5.2 to 5.7, we studied a user throughput maximization problem for the uplink of a cellular network. Now we consider the downlink case. We will show that the downlink user throughput maximization problem can be solved in the exactly same way as the uplink problem, with the same objective value and user throughput.

Consider the downlink channel in Fig. 5.9. We assume that each BS serves the same set of users as in the solution for the uplink problem. So there is no BS selection problem for the downlink case. Suppose that BS j is the service provider of user i. Denote $\hat{\mathcal{S}}_i = \{\hat{s}_i^k : 1 \leq k \leq \hat{\sigma}_i\}$ as the set of streams from BS j to user i, where $\hat{s}_i^k$ is the k-th stream and $\hat{\sigma}_i$ is the number of streams in set $\hat{\mathcal{S}}_i$. For each stream $\hat{s}_i^k$, denote $\hat{\mathbf{u}}_j^l$ as its encoding vector at BS j and $\hat{\mathbf{v}}_i^k$ as its decoding vector at user i. Denote $\hat{\pi}$ as a downlink IA scheme with corresponding encoding vector $\hat{\mathbf{u}}_j^l$ and decoding vector $\hat{\mathbf{v}}_i^k$ for each stream $\hat{s}_i^k$.

Consider an uplink IA scheme π and a downlink IA scheme $\hat{\pi}$, with $\hat{\sigma}_i = \sigma_i$ for $i \in \mathcal{N}$. For each stream $\hat{s}_i^k$ in downlink IA scheme $\hat{\pi}$, we construct its encoding and decoding vectors by letting $\hat{\mathbf{u}}_j^l = \mathbf{v}_j^l$ and $\hat{\mathbf{v}}_i^k = \mathbf{u}_i^k$.[2] Then we have the following theorem:

[2]Recall that $\mathbf{u}_i^k$ and $\mathbf{v}_j^l$ are the encoding and decoding vectors for stream s_i^k in uplink IA scheme π.

Theorem 6. *If IA scheme π is feasible for the uplink, then IA scheme $\hat{\pi}$ is feasible for the downlink, and vice versa.*

Proof. Suppose that IA scheme π is feasible for the uplink, then we know that for each stream s_i^k ($i \in \mathcal{N}, 1 \leq k \leq \sigma_i$), there exist an encoding vector $\mathbf{u}_i^k$ and a decoding vector $\mathbf{v}_j^l$ that satisfy (5.4) and (5.5). Furthermore, the set of encoding vectors $\{\mathbf{u}_i^k : i \in \mathcal{N}, 1 \leq k \leq \sigma_i\}$ can be obtained by (5.6) and (5.7) in Section 5.4.2, while the set of decoding vectors $\{\mathbf{v}_j^l : j \in \mathcal{M}, 1 \leq l \leq \sum_{i \in \mathcal{T}_j^{\text{usr}}} \sigma_i\}$ can be obtained by solving the linear system in the proof of Proposition 1.

Now we consider IA scheme $\hat{\pi}$ with $\hat{\sigma}_i = \sigma_i$. For each stream $\hat{s}_i^k$, its encoding vector $\hat{\mathbf{u}}_j^l$ is constructed by $\hat{\mathbf{u}}_j^l = \mathbf{v}_j^l$ and its decoding vector $\hat{\mathbf{v}}_i^k$ is constructed by $\hat{\mathbf{v}}_i^k = \mathbf{u}_i^k$. Next, we will show that $\hat{\pi}$ is feasible for the downlink by arguing that each steam $\hat{s}_i^k$ in $\hat{\pi}$ can be transported free of interference.

We first check the transfer function of stream $\hat{s}_i^k$ as follows:

$$(\hat{\mathbf{v}}_i^k)^T \mathbf{H}_{ji} \hat{\mathbf{u}}_j^l \overset{(a)}{=} (\mathbf{u}_i^k)^T \mathbf{H}_{ji} \mathbf{v}_j^l \overset{(b)}{=} [(\mathbf{v}_j^l)^T \mathbf{H}_{ji} \mathbf{u}_i^k]^T \overset{(c)}{=} 1, \tag{5.43}$$

where (a) holds due to $\hat{\mathbf{u}}_j^l = \mathbf{v}_j^l$ and $\hat{\mathbf{v}}_i^k = \mathbf{u}_i^k$; (b) holds due to the fact that $\mathbf{H}_{ji}$ is a diagonal matrix, i.e., $(\mathbf{H}_{ji})^T = \mathbf{H}_{ji}$; and (c) holds due to (5.4).

We then check whether the interference can be completely cancelled. For stream $\hat{s}_i^k$, it suffers from interference from the streams that correspond to encoding vectors $\mathbf{v}_{j'}^{l'}$ with $j' \in \mathcal{I}_i^{\text{bs}} \cup \{j\}$, $1 \leq l' \leq \sum_{i' \in \mathcal{T}_{j'}} \sigma_{i'}$, and $(j', l') \neq (j, l)$. Based on (5.5), we have

$$(\hat{\mathbf{v}}_i^k)^T \mathbf{H}_{ji} \hat{\mathbf{u}}_{j'}^{l'} = (\mathbf{u}_i^k)^T \mathbf{H}_{ji} \mathbf{v}_{j'}^{l'} = [(\mathbf{v}_{j'}^{l'})^T \mathbf{H}_{ji} \mathbf{u}_i^k]^T = 0, \tag{5.44}$$

for $j' \in \mathcal{I}_i^{\text{bs}} \cup \{j\}$, $1 \leq l' \leq \sum_{i' \in \mathcal{T}_{j'}} \sigma_{i'}$, and $(j', l') \neq (j, l)$.

Results in (5.43) and (5.44) ensure that each stream $\hat{s}_i^k$ ($i \in \mathcal{N}, 1 \leq k \leq \hat{\sigma}_i$) can be transported free of interference in the downlink. Therefore, we conclude that IA scheme $\hat{\pi}$ is feasible for the downlink.

The converse part can be argued by following the same token and thus we omit it. □

Based on Theorem 6, we have the following observations on π and $\hat{\pi}$:

- For user $i \in \mathcal{N}$, if it can send σ_i data streams to its BS j in the uplink problem, then it can receive σ_i data streams from BS j in the downlink problem, and vice versa.
- For the downlink problem, the encoding and decoding vectors in $\hat{\pi}$ are the same as the corresponding decoding and encoding vectors in π, respectively. That is, stream $\hat{s}_i^k$'s encoding vector is stream s_i^k's decoding vector while stream $\hat{s}_i^k$'s decoding vector is stream s_i^k's encoding vector.

Since the downlink user throughput maximization problem shares the exactly same formulation as the uplink user throughput maximization problem, the optimal objective value and user throughput from both problems are the same.

5.9 Related Work

The concept of IA was coined in a seminar paper by Jafar and Shamai for the two-user X channel [39]. Since its emergence, the idea has gained tremendous momentum in research pursued by industry as well as the academic within a variety of communities. We survey the results of IA in the literature along three research lines: From information theory perspective, from experimentation perspective, and from networking perspective.

In information theory, the results for IA have been developed for a variety of channels and networks in increasingly sophisticated forms. In [9], Cadambe and Jafar showed that by using IA, the aggregate DoF of the K-user interference channel can linearly increase with the number of users K, indicating that each user can obtain "half of cake". In [84] and [85], Suh et al. proposed an IA scheme, namely subspace interference alignment, for both uplink and downlink of cellular networks. They further showed that their proposed IA scheme approaches to interference-free DoF as the number of users in each cell increases. The advances of IA can also be found for other networks and channels, such as the K-user MIMO interference channel [26], the X network with arbitrary number of users [10], the MIMO Y channel [48], ergodic capacity in fading channel [41, 65], and the complex interference channel [11]. A tutorial on IA from information theory perspective is [40].

On another line of research, various efforts have been made to validate the feasibility of IA

in practical networks via experimentation. In [15], El Ayach et al. did an experimental study of IA in MIMO-OFDM interference channels and showed that IA achieves the theoretical throughput gains. In [23], Gollakotta et al. demonstrated that the combination of IA and IC increases the average throughput by 1.5× on the downlink and 2× on the uplink in a 2×2 MIMO WLAN. In [53], Lin et al. proposed a distributed random access protocol (called 802.11n$^+$) based on IA and demonstrated that the system can double the average network throughput in a small network with three pairs of nodes.

In contrast with its prosperity in information theory, the research of IA in networking community is still in its infancy. In [51], Li et al. attempted to explore IA in a multi-hop MIMO networks. However, the key concept of IA was not incorporated into their problem formulation and thus was absent in the final solution. In [37], the authors used IA in their paper title although they only considered transmitter-side zero-forcing technique. In [100], Zeng et al. developed an IA model for multi-hop MIMO networks to facilitate the networking research on IA. However, the proposed IA model is based on the assumption of MIMO channel and thus not applicable to the cellular networks with single antenna.

5.10 Chapter Summary

This chapter advances the state-of-the-art on IA for cellular networks. We developed an IA scheme for cellular networks by removing the limiting assumptions associate with the existing literature. Specifically, our IA scheme allows heterogeneous data streams from each user, finite number of subcarriers, different number of users for each BS, and asymmetric interference pattern between user and BS. We prove the feasibility of our IA scheme by constructing encoding and decoding vectors for each data stream. Based on the proposed IA scheme, we studied an uplink user throughput maximization problem and demonstrated its throughput improvement over some other schemes. For the downlink problem, we showed the IA scheme for the uplink could be carried over to the downlink by reserving the role of user and BS. Further, the downlink user throughput maximization problem has the same formulation as the uplink problem and can be solved in the same way.

Chapter 6

Temporal Interference Alignment in Underwater Acoustic Networks

6.1 Introduction

The growing need for oceanographic data collection, remote sensing, and tactical communications has led to a surge of research efforts in the area of underwater acoustic (UWA) networks [2]. A fundamental issue in UWA networks is the large propagation delays as a result of slow speed of sound traveling in water (1500 m/s) [81]. For instance, it takes about 2/3 seconds for sound to travel 1 km in water. As expected, such large propagation delay phenomenon poses a fundamental limitation on the performance of UWA networks.

In wireless networking community, the prevailing approach is to design efficient protocols and algorithms that take into account the characteristics of large propagation delays associated with UWA networks [28, 30, 47, 55, 61, 67, 69]. In other words, all these approaches accept the fundamental limitation of large propagation delays and try to develop the best possible protocols and algorithms that live with such limitation. As expected, the throughput under such an approach deteriorates rapidly when the propagation delays increase. In this chapter, instead of considering large propagation delays as an adversary, we exploit propagation delays as an advantage for throughput improvement in UWA networks. A technique

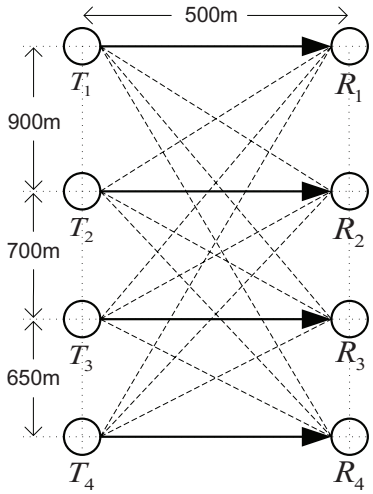
Table 6.1: Propagation delays normalized with respect to a symbol duration.

Node	R_1	R_2	R_3	R_4
T_1	3.9	8.0	13.1	18.0
T_2	8.0	3.9	6.7	11.2
T_3	13.1	6.7	3.9	6.4
T_4	18.0	11.2	6.4	3.9

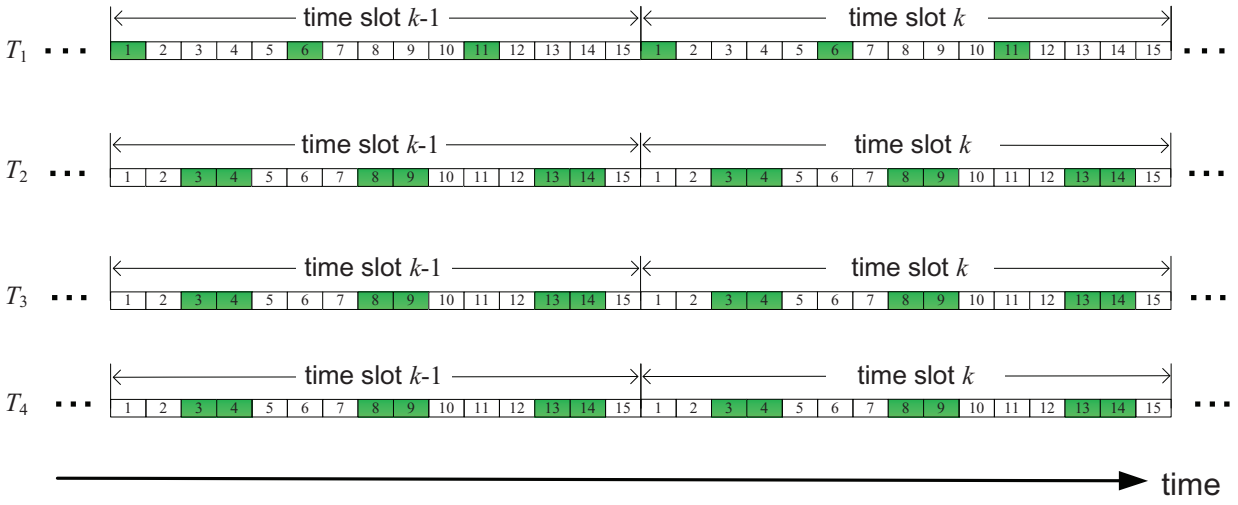
that shows such an approach is plausible is time-based IA, or propagation delay (PD)-based IA (PD-IA). PD-IA aims to achieve overlapping of interference shadows by exploiting the propagation delays of signals from their transmitters to receivers. Specifically, it exploits large propagation delays as an enabling physical phenomenon that allows to project interference from different transmitters to the same time intervals, leaving more time intervals available for data transmission. Note that PD-IA is uniquely suitable for UWA networks as propagation delays here are on a *larger* time scale than the symbol duration. PD-IA is not suitable for terrestrial wireless networks as their propagation delays are on a much *smaller* time scale.

6.1.1 PD-IA: A Motivating Example

To see how PD-IA works in UWA networks with large propagation delays, consider a 4-link network shown in Fig. 6.1(a), where the solid lines represent intended transmissions and dashed lines represent interference. The propagation delays between any transmit and receive nodes can be computed based on the distances between the nodes and the speed of sound in water (1500 m/s). Suppose that the data transmission is done in time slots, with each time slot carrying 15 OFDM symbols and each OFDM symbol having 85.5 ms duration [50]. Then the normalized propagation delays (with respect to the OFDM symbol duration time) between the transmitters and receivers can be computed, as shown in Table 6.1. Suppose that we schedule OFDM symbol payload at each transmitter as in Fig. 6.1(b), where the shadowed intervals represent payload while empty intervals represent idle time. Then at

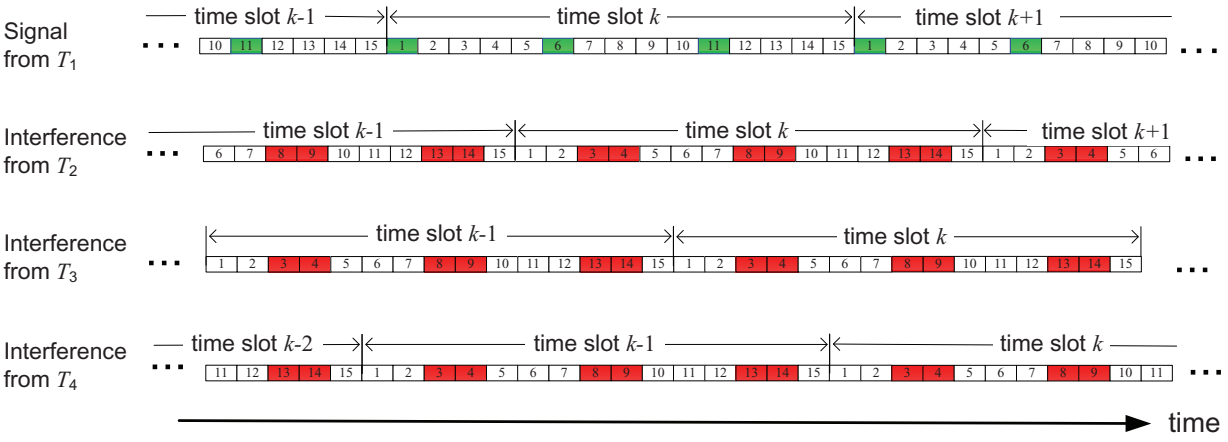


(a) Four mutually interfering transmissions



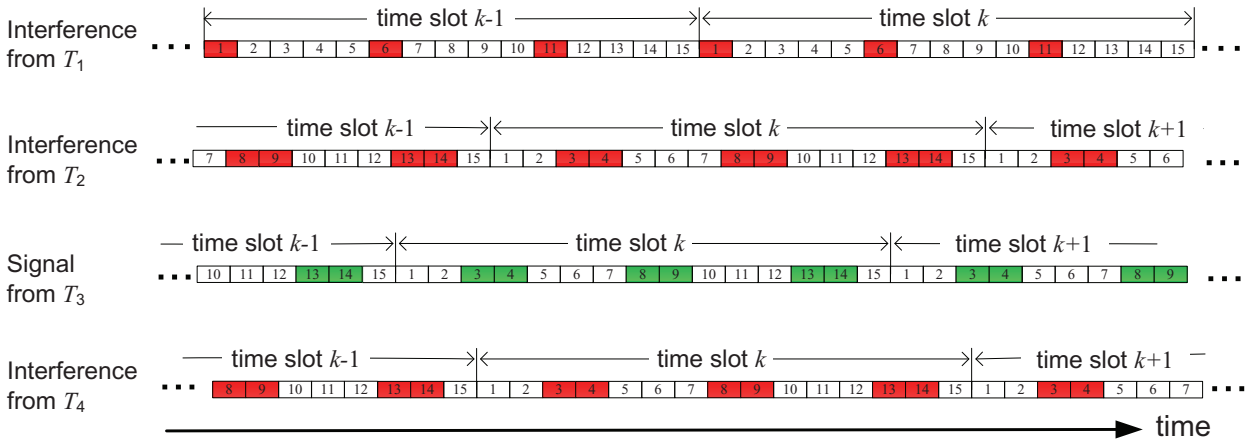
(b) A schedule of payloads at each transmitter

Figure 6.1: An example of PD-IA in an UWA network.

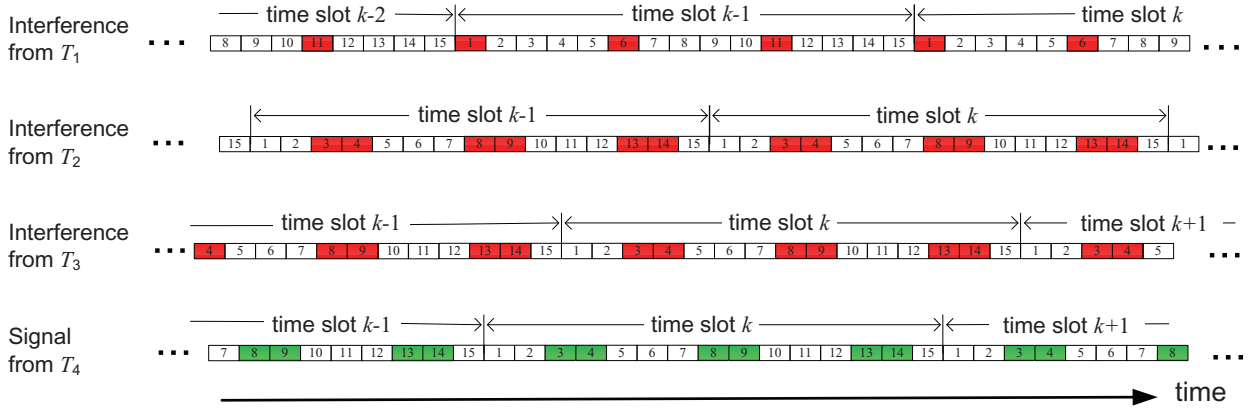


(a) Received signal and interference at receiver R_1

(b) Received signal and interference at receiver R_2



(c) Received signal and interference at receiver R_3



(d) Received signal and interference at receiver R_4

Figure 6.2: Received signal and interference at each receiver.

each receiver, it receives its desired symbol stream, plus three interfering symbol streams. Based on their respective propagation delays in Table 6.1, the received symbol streams at each receiver are shown in Fig. 6.2. We can see that the desired symbol stream at each receiver is completely separated (no overlap) from the other three interfering streams, thanks to the payload scheduling in Fig. 6.1(b) in conjunction with the underlying propagation delays between the nodes. For example, in Fig. 6.2(a), the desired symbols at receiver R_1 (from T_1) are completely free of interference from the other three interfering streams (from T_2, T_3, and T_4). On the other hand, there is overlap among the symbols from the three interfering streams (T_2, T_3, and T_4). Likewise, similar separation and alignment of desired and interfering symbols occur at receivers R_2, R_3, and R_4, as shown in Fig. 6.2(b), (c) and (d), respectively.

Quantitatively, in Fig. 6.2, we have a total of 21 payload symbols that are successfully transmitted over 15 symbol intervals. That is, this scheduling of payload symbols allows 6 more symbols to be transported over 15 symbol intervals, making an increase of 40% for spectral efficiency.[1] As this example shows, the essence of PD-IA is the design of a scheduling algorithm to exploit the specific propagation delays between the transmitters and the receivers so that at each receiver, the interfering symbols overlap as much as possible while the desired symbols are free of interference.

6.1.2 Goals of This Chapter

Although the idea of PD-IA has been studied by some researchers, the current results are either based on information theory (IT) perspective [8, 27] or limited to physical-layer/single-hop scenario [6, 13]. It remains unclear how PD-IA can be fully exploited in a complex *multi-hop* UWA network. The goal of this chapter is to explore PD-IA in UWA networks so that the benefits of PD-IA can be reaped at the network level. Specifically, we are interested in how to take advantage of PD-IA to improve throughput in a multi-hop UWA network

[1] If there were no propagation delays, at most 15 payload symbols can be transported in this network since all links are in the same collision domain.

with large propagation delays.

6.1.3 Main Contributions

We propose a TDMA-based frame structure for scheduling and data transmission in a multi-hop UWA network. Under this frame structure, we develop an analytical model for PD-IA in each time slot. Our model consists of a set of constraints such that at each receiver: (i) its *desired* payload symbols are received free of interference, and (ii) the *interfering* payload symbols are allowed to overlap.

Based on this model, we study a throughput maximization problem in a multi-hop UWA network with a set of sessions. Specifically, we develop a distributed PD-IA scheduling algorithm, called Shark-IA, to maximize the minimum rate among a set of sessions. In essence, Shark-IA is an iterative greedy algorithm that attempts to increase the minimum rate among all active links in each iteration. During each iteration, it chooses a symbol interval for payload so that the interference generated by this new payload symbol is maximally overlapped at its non-intended receivers.

To evaluate the performance of Shark-IA algorithm, we first compare it to an idealized benchmark algorithm with perfect scheduling and zero propagation delays (similar to the comparison example in Fig. 6.1). Our simulation results show that Shark-IA can significantly outperform this idealized benchmark algorithm. Further, we find that the performance gain widens as traffic volume in the network increases. We also compare Shark-IA algorithm to a centralized solution with perfect PD-IA scheduling. Our simulation results show that Shark-IA can achieve more than 80% optimal throughput performance by the centralized solution when the number of sessions is small. When the number of sessions becomes large, the centralized solution is no longer computable (even on the supercomputer in our institution), while Shark-IA can yield a competitive feasible solution very quickly.

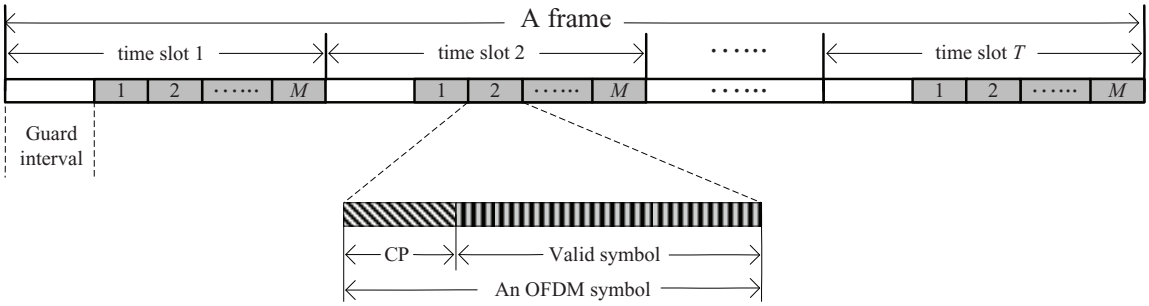


Figure 6.3: A frame structure.

6.1.4 Chapter Organization

The remainder of this chapter is organized as follows. In Section 6.2, we develop a PD-IA model for UWA networks. In Section 6.3, we develop a distributed scheduling algorithm to explore PD-IA in a multi-hop network. In Section 6.4, we present the simulation results. Section 6.5 presents related work on PD-IA and Section 6.6 concludes this chapter. Table 6.2 lists the notation that we will use in this chapter.

6.2 A PD-IA Model

In this section, we develop a basic model to study PD-IA. Such a model is important for network level research of PD-IA in a multi-hop UWA network.

6.2.1 A Frame Structure

We consider a TDMA frame structure in Fig. 6.3 for scheduling and data transmission in a network environment [81]. Each node repeats the same frame structure over time. As shown in Fig. 6.3, a frame is divided into T time slots, each of which consists of a guard interval and M OFDM symbols.

A guard interval is employed at the head of each time slot to eliminate the "tail effect" of the previous time slots from unintended transmitters. That is, the use of guard interval in a time slot allows independent PD-IA scheduling of time slots at all nodes in the network.

Table 6.2: Notation for temporal IA in UWA networks.

Network setting	
$\mathcal{F}$	The set of sessions in the network
F	The number of sessions in the network
$\mathcal{L}$	The set of links traversed by the sessions
L	The number of links in the network
$\mathcal{L}_i^{\text{in}}$	The set of incoming links at node i
$\mathcal{L}_i^{\text{out}}$	The set of outgoing links at node i
$\mathcal{N}$	The set of nodes in the network
N	The number of nodes in the network
T	The number of time slots in a frame
$\text{Tx}(l)$	The transmitter of link l
$\text{Rx}(l)$	The receiver of link l
$\text{src}(f)$	The source node of session f
$\text{dst}(f)$	The destination node of session f
PD-IA modeling	
M	The number of OFDM symbols in a time slot
$\mathcal{P}_l$	The set of links whose receiver is within the interference range of $\text{Tx}(l)$
$\mathcal{Q}_l$	The set of links whose transmitter is within the interference range of $\text{Rx}(l)$
τ	The time duration of an OFDM symbol (e.g., 85.5 ms)
δ_{lk}	The symbol offset of signals from $\text{Tx}(l)$ and $\text{Tx}(k)$ at $\text{Rx}(l)$
d_{lk}	The distance between $\text{Rx}(l)$ and $\text{Tx}(k)$
$f_{lk}^{\text{L}}(m)$	The index of the symbol on link $k \in \mathcal{P}_l$ that cause collision at the m-th symbol on link l
$f_{lk}^{\text{R}}(m)$	The index of the symbol on link $k \in \mathcal{P}_l$ that cause collision at the m-th symbol on link l

Table 6.2: Continued.

$g_{lk}^{\mathrm{L}}(m)$	The index of the symbol on link $k \in \mathcal{Q}_l$ that are collided by the m-th symbol on link l
$g_{lk}^{\mathrm{R}}(m)$	The index of the symbol on link $k \in \mathcal{Q}_l$ that are collided by the m-th symbol on link l
$z_l(t,m)$	The activity of the m-th symbol in time slot t on link l
Scheduling algorithm design	
$\mathcal{M}$	A set of eligible symbol intervals in time slot t on link l that are eligible for payload
$\mathcal{R}$	A set of symbol intervals on link l that can be potentially used for payload
b_l	The potential interference burden of link l
c_l	The average data rate of link l in a time frame
$p_l(t,m)$	The amount of interference overlapping shadows created by symbol (t,m) on link l
$q_l(t,m)$	The amount of payload adjustment for symbol interval (t,m) on link l
$s(i,t)$	The status of node i in time slot t
$y_l(t,m)$	The amount of interference overlapping shadows at symbol (t,m) on link l

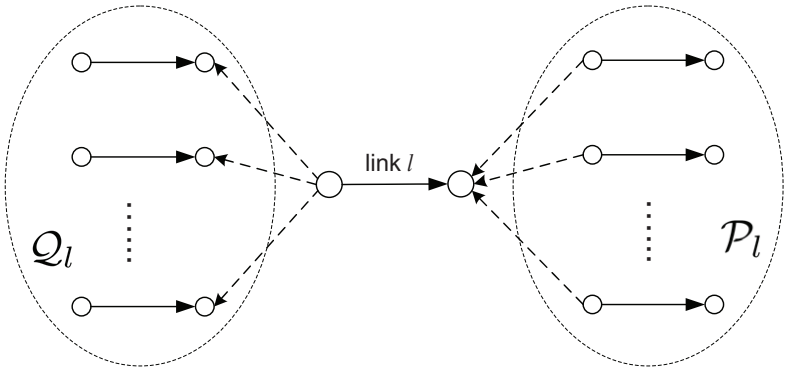


Figure 6.4: An example that illustrates sets $\mathcal{P}_l$ and $\mathcal{Q}_l$ for link l.

To serve this purpose, the duration of guard interval should be greater than the maximum propagation delay between any two interfering nodes.

Each OFDM symbol consists of two parts: cyclic prefix (CP) and valid symbol. The part of valid symbol can be used to carry payload packet and the part of CP is used to eliminate the "multipath effect" of channel. We note that our PD-IA scheduling does not require the existence of line-of-sight channel between any two nodes as [27], since the OFDM modulation can effectively eliminate the inter-symbol interference (ISI) caused by the multipath channel. For example, the OFDM modulation in [50] has a CP of 20 ms duration. Since the delay caused by the multipath channel in UWA networks is typically less than 11 ms [83], this OFDM modulation can completely eliminate the ISI caused by multipath channel.

In this frame structure, half-duplex of a node's transceiver is on time slot level. That is, a node would not change its status (transmitting, receiving, or idling) within a time slot, and can only change its status for a different time slot. In each time slot, the smallest granularity of our PD-IA scheduling is the OFDM symbol. Specifically, for each OFDM symbol, our PD-IA scheduling is to determine whether or not it is used to carry payload packet.

6.2.2 Constraints for OFDM Symbol Payload

We study the constraints for PD-IA scheduling in each time slot in a frame. Specifically, we wish to describe the constraints for each OFDM symbol in a time slot that can be scheduled

for payload while not being interfered with by other links.

Denote $\mathcal{L}$ as the set of links that are traversed by the sessions in the network. Denote $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$ as the transmit and receive nodes of link $l \in \mathcal{L}$, respectively. Referring to Fig. 6.4, denote $\mathcal{P}_l$ as the set of links whose transmitters are interfering with $\mathrm{Rx}(l)$. Similarly, denote $\mathcal{Q}_l$ as the set of links whose receivers are being interfered with by $\mathrm{Tx}(l)$. Now consider $\mathrm{Rx}(l)$. $\mathrm{Rx}(l)$ receives both of the desired signal from $\mathrm{Tx}(l)$ and the interfering signals from $\mathrm{Tx}(k)$, $k \in \mathcal{P}_l$. Suppose that all transmit nodes in the network are synchronized. Then the received signals from intended and unintended transmitters, after taking into consideration their respective propagation delays, will exhibit a time shift with respect to their time slots, as shown in Fig. 6.5(a). Denote d_{lk} as the Cartesian distance between $\mathrm{Rx}(l)$ and $\mathrm{Tx}(k)$. Denote δ_{lk} as the time offset (in number of OFDM symbols) in a time slot between the desired signal and undesired signal (interference). Then we have

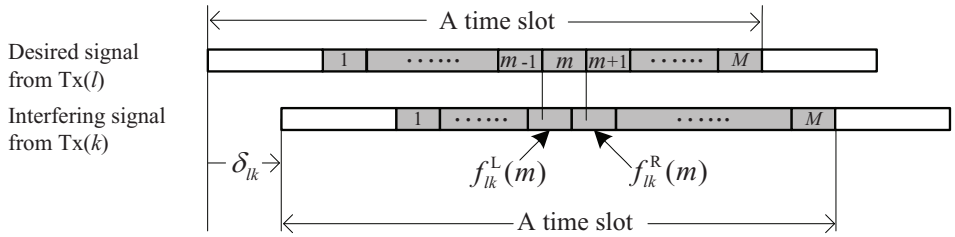
$$\delta_{lk} = \frac{d_{lk} - d_{ll}}{c\tau},$$

where c is the speed of sound in water and τ is the time duration of an OFDM symbol (e.g., $\tau = 85.5$ ms in [50]). Note that a negative value of δ_{lk} indicates that the interfering transmit node $\mathrm{Tx}(k)$ is closer to $\mathrm{Rx}(l)$ than the intended transmit node $\mathrm{Tx}(l)$.

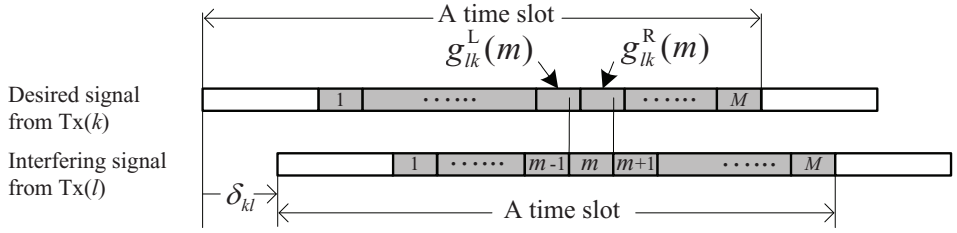
Referring to Fig. 6.3, denote m as the position of a symbol in a time slot, with $1 \leq m \leq M$. Denote $z_l(t, m)$ as the indicator of a symbol payload at position m in time slot t for link l. Specifically, $z_l(t, m) = 1$ if the symbol at position m in time slot t is a payload for link l and $z_l(t, m) = 0$ otherwise. For ease of exposition, denote 0 as the position of the guard interval in a time slot. Since a guard interval is filled with null symbols, we have $z_l(t, 0) \equiv 0$ for $l \in \mathcal{L}$ and $1 \leq t \leq T$.

To explore the constraints for symbol payload, we consider the case for $\mathrm{Rx}(l)$ and $\mathcal{P}_l$ and the case for $\mathrm{Tx}(l)$ and $\mathcal{Q}_l$ separately, as shown in Fig. 6.5(a) and (b), respectively.

Constraints for $\mathrm{Rx}(l)$ and $\mathcal{P}_l$: As illustrated in Fig. 6.5(a), for the m-th symbol in a time slot at $\mathrm{Rx}(l)$, it may be interfered with by two consecutive symbols from $\mathrm{Tx}(k)$, $k \in \mathcal{P}_l$. Denote the positions of these two interfering symbols from $\mathrm{Tx}(k)$ as $f_{lk}^{\mathrm{L}}(m)$ and $f_{lk}^{\mathrm{R}}(m)$,



(a) At Rx(l), signal from Tx(l) and interference from Tx(k), $k \in \mathcal{P}_l$.



(b) At Rx(k), signal from Tx(k) and interference from Tx(l), $k \in \mathcal{Q}_l$.

Figure 6.5: An example that illustrates the constraints.

respectively.[2] Then we have

$$f_{lk}^{\mathrm{L}}(m) = \begin{cases} m - \lfloor \delta_{lk} \rfloor & \text{if } 1 + \lfloor \delta_{lk} \rfloor \le m \le M + \lfloor \delta_{lk} \rfloor, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{lk}^{\mathrm{R}}(m) = \begin{cases} m - \lceil \delta_{lk} \rceil & \text{if } 1 + \lceil \delta_{lk} \rceil \le m \le M + \lceil \delta_{lk} \rceil, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, in time slot t, if the m-th position on link l carries a symbol payload, then the $f_{lk}^{\mathrm{L}}(m)$-th and $f_{lk}^{\mathrm{R}}(m)$-th positions on interfering link k cannot carry a symbol payload. Mathematically, this can be characterized by:

$$z_l(t,m) + \frac{1}{2}\Big[z_k\Big(t, f_{lk}^{\mathrm{L}}(m)\Big) + z_k\Big(t, f_{lk}^{\mathrm{R}}(m)\Big)\Big] \le 1, \quad \text{for } k \in \mathcal{P}_l, l \in \mathcal{L}, 1 \le m \le M, 1 \le t \le T. \tag{6.1}$$

Constraints for Tx(l) **and** $\mathcal{Q}_l$**:** As illustrated in Fig. 6.5(b), at Rx(k), the m-th symbol in a time slot from Tx(l) may be interfering with two consecutive symbols from Tx(k). Denote

[2] In some cases, $f_{lk}^{\mathrm{L}}(m)$ and $f_{lk}^{\mathrm{R}}(m)$ may be equal. Our scheduling algorithm can handle these cases.

$g_{lk}^{\mathrm{L}}(m)$ and $g_{lk}^{\mathrm{R}}(m)$ as the positions of these two desired symbols at $\mathrm{Rx}(k)$ that are being interfered with by the m-th symbol from $\mathrm{Tx}(l)$, respectively.[3] Then we have

$$g_{lk}^{\mathrm{L}}(m) = \begin{cases} m + \lfloor \delta_{kl} \rfloor & 1 - \lfloor \delta_{kl} \rfloor \leq m \leq M - \lfloor \delta_{kl} \rfloor, \\ 0 & \text{otherwise.} \end{cases}$$

$$g_{lk}^{\mathrm{R}}(m) = \begin{cases} m + \lceil \delta_{kl} \rceil & 1 - \lceil \delta_{kl} \rceil \leq m \leq M - \lceil \delta_{kl} \rceil, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, in time slot t, if the m-th position from interfering link l carries a symbol payload, then the $g_{lk}^{\mathrm{L}}(m)$-th and $g_{lk}^{\mathrm{R}}(m)$-th positions on link k cannot carry a symbol payload. Mathematically, this can be characterized by:

$$z_l(t,m) + \frac{1}{2}\Big[z_k\Big(t, g_{lk}^{\mathrm{L}}(m)\Big) + z_k\Big(t, g_{lk}^{\mathrm{R}}(m)\Big)\Big] \leq 1, \qquad k \in \mathcal{Q}_l, l \in \mathcal{L}, 1 \leq m \leq M, 1 \leq t \leq T. \tag{6.2}$$

6.3 Shark-IA: A Distributed PD-IA Scheduling Algorithm

Consider a multi-hop UWA network with a set of unicast sessions in the network (see Fig. 6.10 for example). The route from the source node of each session to its destination node is given *a priori,* which can be computed by some distributed routing protocol (e.g., the AODV algorithm [70]). We develop a distributed payload scheduling algorithm based on the proposed PD-IA model, with the objective of maximizing the minimum rate among the sessions. To do so, we first state our assumptions and give an overview of the algorithm. Then we explain the details of each key module in the algorithm. Finally, we discuss its complexity.

6.3.1 Assumptions

We have the following assumptions in the design of our distributed scheduling algorithm.

[3] Similarly, in some cases, $g_{lk}^{\mathrm{L}}(m)$ and $g_{lk}^{\mathrm{R}}(m)$ may be equal. Our scheduling algorithm can handle these cases.

- Each session has a persistence and latency-tolerant traffic at its source. This assumption helps us to explore the full potential of PD-IA and simplify the discussion of the algorithm.
- The nodes in the network are well synchronized. Some distributed synchronization protocols (see, e.g., [88]) can achieve several microsecond synchronization error within 10 seconds in the UWA environment. Since the duration of OFDM symbol is at the level of 100 ms (e.g., 85.5 ms [50]), our scheduling algorithm is robust to the synchronization error.
- Every node knows the location information of its neighboring nodes. Based on the location information, a node can compute the value of propagation delays between itself and its neighboring nodes. Since the existing distributed localization schemes can achieve the accuracy of 1 m for a 3 km $\times$ 4 km area [12], the propagation delay error caused by the inaccurate location information is less than 1 ms. Our scheduling algorithm is also robust to the location information error because it does not require perfect alignment of the OFDM symbols.
- Each node in the network can exchange scheduling information with those nodes inside its interference range. This is a mild assumption since there are many ways to achieve it in a distributed environment. Given that this is not our contribution, we skip its discussion to conserve space.

Since the goal of this chapter is to outline a distributed algorithm to show the benefits of PD-IA in a multi-hop network, issues associated with protocol design (e.g., message format, control packet delay and overhead, recovery from lost packet) are beyond the scope of this chapter.

6.3.2 Algorithm Overview

In essence, the proposed distributed algorithm is a greedy algorithm that attempts to increase the minimum rate among all the links traversed by the sessions iteratively. This is equivalent to increasing the rate on each link traversed by each session iteratively (a link traversed by

multiple sessions will be considered for multiple times). For a given link, the algorithm attempts to increase its rate by finding a symbol interval that can be used for a payload. Among the eligible symbol intervals, the final choice is determined by IA, in the sense that we wish to have the interference from this symbol to overlap with as many interfering symbols from other links as possible. If we cannot find any such eligible symbol interval in a frame on link l, then we attempt to make some adjustment on the current payload structure in sets $\mathcal{P}_l$ and $\mathcal{Q}_l$ so that some symbol interval can be used for a new payload on link l.

As shown in Fig. 6.6, there are three main modules in the algorithm: *link ordering, payload-IA,* and *payload adjustment*. We briefly describe them as follows:

- **Link Ordering.** The goal of this module is to order all the links traversed by the sessions into a list (a link traversed by multiple sessions will be on the list for multiple times) and consider the links on the list sequentially and cyclically. To exploit the most opportunity for interference overlapping from the beginning, we order the links in the list based on non-increasing values of interference burden.
- **Payload-IA.** The goal of this module is to increase the rate of the selected link by one payload symbol without any payload adjustment on other links. Recall that the time granularity for half-duplex at a node is the time slot. So we find a time slot (starting from the first time slot in a frame) on link l that satisfies half-duplex constraint at both Tx(l) and Rx(l). Further, we identify a set of eligible symbol intervals (i.e., the unused symbol intervals that meet the PD-IA constraints) for rate increment in that time slot. When there exist multiple eligible symbol intervals, we choose the one that can produce the most interference overlapping shadows on the neighboring links.
- **Payload Adjustment.** If the payload-IA module fails to find any eligible symbol interval in all time slots, then we attempt to make some necessary adjustment on the current payload structure in $\mathcal{P}_l$ and $\mathcal{Q}_l$ so that some symbol interval on link l can be assigned for a new payload. Specifically, we identify a set $\mathcal{R}$ of symbol intervals (over all time slots in a frame) on link l that meet half-duplex constraints at Tx(l) and Rx(l) but fail to meet the PD-IA constraints. Then we iteratively choose a symbol interval in

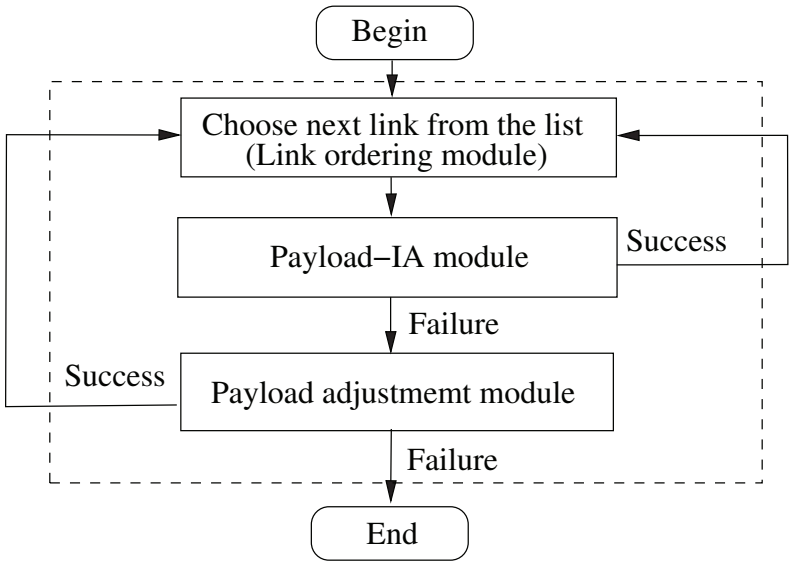


Figure 6.6: A flow chart of our proposed scheduling algorithm.

$\mathcal{R}$ (starting from the one that requires the minimum adjustment) and make necessary adjustment in the current payload structure in $\mathcal{P}_l$ and $\mathcal{Q}_l$, with the goal of turning this symbol interval into an eligible interval. The module terminates once such an interval is found (and thus a symbol payload can be assigned to this interval) or none of the symbol intervals in $\mathcal{R}$ works out.

6.3.3 Data Structure

For the implementation of this distributed algorithm, the state information that we maintain at each node $i \in \mathcal{N}$ is as follows:

- *Frame structure.* Since each node repeats the same transmission/reception/idling behavior at the frame level, it needs to maintain a frame structure as shown in Fig. 6.3.
- *Incoming/outgoing links.* Each node maintains the set $\mathcal{L}_i^{\text{in}}$ of its incoming links traversed by the set of unicast sessions in the network. Also, each node maintains the set $\mathcal{L}_i^{\text{out}}$ of its outgoing links traversed by the set of unicast sessions in the network.
- *Node half-duplex status.* Since the time granularity for half-duplex at a node is done at time slot level, each node i needs to maintain its half-duplex status in each time slot

for each link in $\mathcal{L}_i^{\text{in}} \cup \mathcal{L}_i^{\text{out}}$. Denote $s_l(i,t)$ as the half-duplex status ("IDLE", "TX", "RX") of node i in time slot t for link $l \in \mathcal{L}_i^{\text{in}} \cup \mathcal{L}_i^{\text{out}}$. $s_l(i,t)$ = "TX" ("RX") means that node i is used as the transmitter (receiver) of link l in time slot t.

- *Current payload scheduling.* Each node i needs to maintain the payload scheduling status at symbol level (with or without payload) in each time slot for its associated links. Recall that we have denoted $z_l(t,m)$ for payload status of the m-th symbol in time slot t for link $l \in \mathcal{L}_i^{\text{in}} \cup \mathcal{L}_i^{\text{out}}$.
- *Interfering links.* If node i's half-duplex status is "TX" for link l, then it needs to maintain the set of links $\mathcal{Q}_l$ that is being interfered with by link l. Similarly, if node i's half-duplex status is "RX" for link l, then it needs to maintain the set of links $\mathcal{P}_l$ that are interfering with link l.
- *Node location.* Each node needs to maintain the location information of all the nodes within its interference range. Based on the location information, each node can compute the propagation delay between itself and its neighboring nodes.

The state information at each node i is initialized as follows: $s_l(i,t)$ = "IDLE" for $i \in \mathcal{N}$, $l \in \mathcal{L}_i^{\text{in}} \cup \mathcal{L}_i^{\text{out}}$, and $1 \le t \le T$; $z_l(t,m) = 0$ for $l \in \mathcal{L}$, $1 \le t \le T$, and $1 \le m \le M$.

6.3.4 Link Ordering Module

Our proposed distributed algorithm is a greedy algorithm that attempts to increase the minimum rate among all the links traversed by the sessions iteratively. This is equivalent to increasing the rate of each link traversed by each session iteratively (a link traversed by multiple sessions will be considered for multiple times). A straightforward approach is to order all the links traversed by the sessions into a list (a link traversed by multiple sessions will be on the list for multiple times) and consider the links in the list sequentially. Here, we find that the ordering of the links in this list plays an important role in the performance of the algorithm. In our algorithm, we propose to order the links in the list based on their "interference burden", which is defined as follows.

Definition 3. *The interference burden of a link $l \in \mathcal{L}$, denoted as b_l, is defined as the number*

of links in $\mathcal{P}_l$ and $\mathcal{Q}_l$, i.e., $b_l = |\mathcal{P}_l| + |\mathcal{Q}_l|$.

By ordering the links in the list based on non-increasing values of interference burden, we are exploiting the most opportunity for interference overlapping from the beginning. The details of how to schedule a symbol payload for a link by exploiting PD-IA will be explained in Sections 6.3.5 and 6.3.6. Note that each iteration considers one link in the list. After all the links in the list are considered sequentially, the algorithm returns to the first link in the list in a cyclic manner until the algorithm terminates.

For distributed implementation, we assume that there is a dedicated control channel for scheduling. The status of start or completion of a particular iteration is shared among the nodes via this control channel. To find the order of a link in the link ordering list in a distributed network, we adopt the distributed ranking algorithm by Zaks [98]. Zaks' algorithm was proposed to solve the problem of ranking the nodes in a network with a given initial value in non-decreasing order. To adopt this node ordering algorithm for our link ordering problem, we can have the receiver of link $l \in \mathcal{L}$ maintain the link's interference burden b_l, and then execute the distributed ranking algorithm by treating $1/b_l$ as its initial value (as we are interested in a non-increasing order of links).

6.3.5 Payload-IA Module

The goal of this module is to increase the rate of the current link by one payload symbol without any change of payload on other links. To do so, we consider the time slots in a frame in a sequential order, starting with the first time slot. If the rate increment attempt fails in a time slot, we try the next time slot and so forth, until a rate increment is successful or it fails in all T time slots. A flow chart of payload-IA module is given in Fig. 6.7.

Choosing a symbol interval in time slot t. Suppose that the current iteration is on link l. Denote i and j as the transmit and receive nodes of link l, i.e., $i = \mathrm{Tx}(l)$ and $j = \mathrm{Rx}(l)$.

At transmit node i, we first check its half-duplex status in time slot t for link l. To consider time slot t for rate increment, its half-duplex status must be "IDLE" or "TX" for

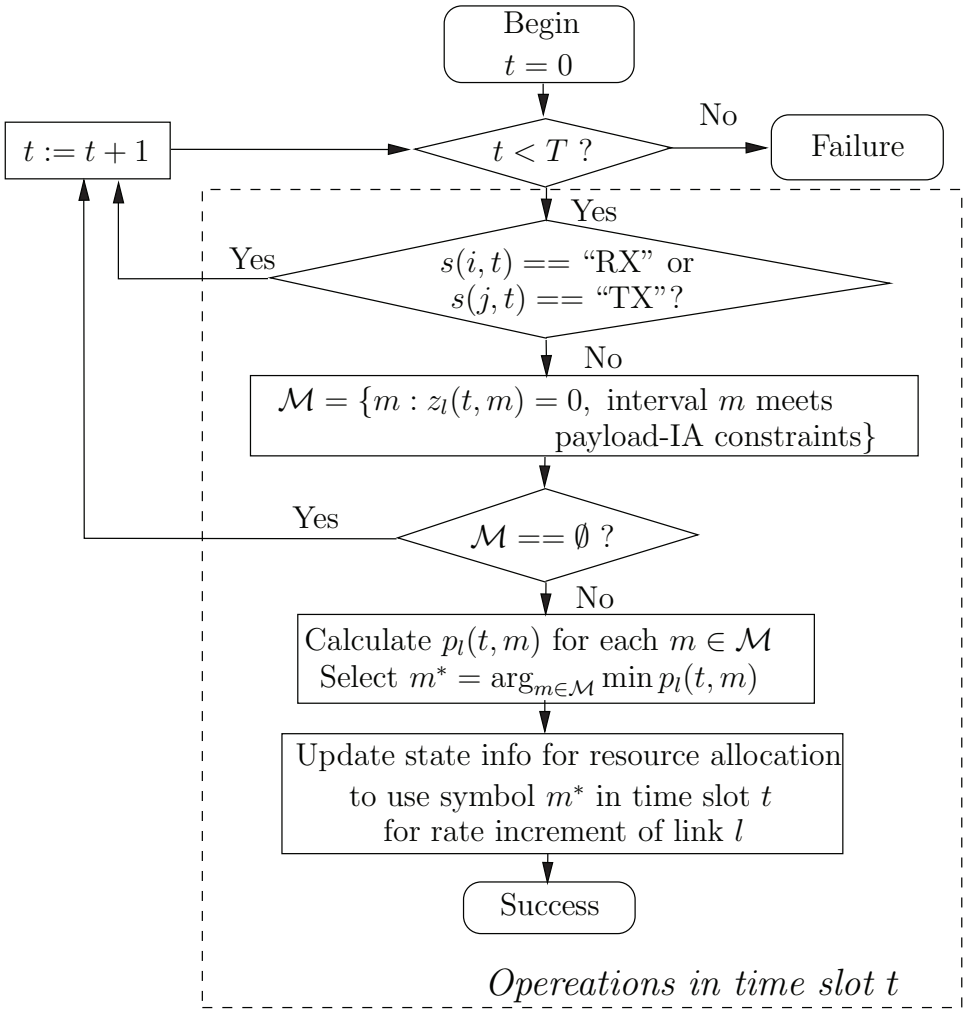


Figure 6.7: A flow chart of payload-IA module.

link l. Otherwise, this time slot cannot be used for rate increment for link l. Likewise, at receive node j, to consider the same time slot t for rate increment, its half-duplex status must be "IDLE" or "RX" for link l. Otherwise, this time slot still cannot be used for rate increment for link l.

If both transmit node i and receive node j meet the half-duplex requirement, then we move on to find a set of eligible symbol intervals for payload in this time slot. At transmit node i, we identify a set of unused symbol intervals in this time slot that meet the PD-IA

constraint (6.2), which we denote as $\mathcal{M}_i$, i.e.,

$$\mathcal{M}_i = \Big\{ m : z_l(t,m) = 0, z_k(t, g^{\mathrm{L}}_{lk}(m)) = 0 \text{ and } \qquad z_k(t, g^{\mathrm{R}}_{lk}(m)) = 0 \text{ for } k \in \mathcal{Q}_l \Big\}.$$

Likewise, at receive node j, we identify a set of unused symbol intervals in this time slot that meet the PD-IA constraint (6.1), which we denote as $\mathcal{M}_j$, i.e.,

$$\mathcal{M}_j = \Big\{ m : z_l(t,m) = 0, z_k(t, f^{\mathrm{L}}_{lk}(m)) = 0 \text{ and } \qquad z_k(t, f^{\mathrm{R}}_{lk}(m)) = 0 \text{ for } k \in \mathcal{P}_l \Big\}.$$

An unused symbol interval is eligible for payload on link l only if this interval is in both $\mathcal{M}_i$ and $\mathcal{M}_j$. Denote $\mathcal{M}$ as the set of such eligible intervals on link l, i.e., $\mathcal{M} = \mathcal{M}_i \cap \mathcal{M}_j$. Although all symbol intervals in $\mathcal{M}$ are eligible for payload, which symbol interval to choose from $\mathcal{M}$ for payload in this iteration is important. Our approach is to choose the one that can create the most interference overlapping shadows on the other links, since this will exploit IA to the fullest extent.

We now consider a link $k \in \mathcal{Q}_l$ and its receive node $\mathrm{Rx}(k)$ as shown in Fig. 6.8. For $\mathrm{Rx}(k)$, denote $y_k(t,n)$ as the amount of interference overlapping shadows on its n-th symbol interval in time slot t on link k, which we define as follows:

$$y_k(t,n) = \sum_{h \in \mathcal{P}_k} \Big[z_h\Big(t, f^{\mathrm{L}}_{kh}(n)\Big) + z_h\Big(t, f^{\mathrm{R}}_{kh}(n)\Big) \Big], \qquad k \in \mathcal{Q}_l, 1 \le t \le T, 1 \le n \le M.$$

As shown in Fig. 6.4 and Fig. 6.5(b), symbol interval m from node i (i.e., $\mathrm{Tx}(l)$) is overlapping with the $g^{\mathrm{L}}_{lk}(m)$-th and $g^{\mathrm{R}}_{lk}(m)$-th symbol intervals at $\mathrm{Rx}(k)$, $k \in \mathcal{Q}_l$. If the $g^{\mathrm{L}}_{lk}(m)$-th and $g^{\mathrm{R}}_{lk}(m)$-th symbol intervals at $\mathrm{Rx}(k)$ are already interfered with by other links (i.e., $y_k(t, g^{\mathrm{L}}_{lk}(m)) \ge 1$ and $y_k(t, g^{\mathrm{R}}_{lk}(m)) \ge 1$), then the setting of symbol interval $m \in \mathcal{M}$ for a payload will only align new interference on these already interfered intervals rather than adding interference on some uninterfered intervals. Therefore, we choose a symbol interval that would cast the maximum interference overlapping shadows on the links in $\mathcal{Q}_l$.

Denote $p_l(t,m)$ as the amount of interference overlapping shadows casted by symbol interval $m \in \mathcal{M}$ on the links in $\mathcal{Q}_l$. Then we have

$$p_l(t,m) = \sum_{k \in \mathcal{Q}_l} \Big[1^+\Big(y_k\big(t, g^{\mathrm{L}}_{lk}(m)\big)\Big) + 1^+\Big(y_k\big(t, g^{\mathrm{R}}_{lk}(m)\big)\Big) \Big], \tag{6.3}$$

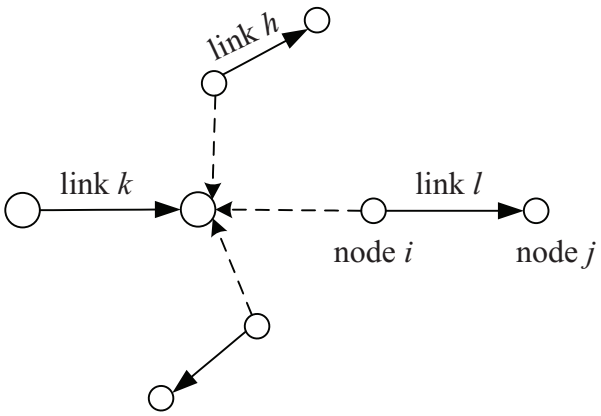


Figure 6.8: The interference overlapping shadows on link k.

where $1^+(x)$ is an indicator function (i.e., $1^+(x) = 1$ if $x \geq 1$ and $1^+(x) = 0$ otherwise).

Denote m^* as the symbol interval that leads to the maximum value of $p_l(t, m)$, $m \in \mathcal{M}$, i.e.,[4]

$$m^* = \arg_{m \in \mathcal{M}} \max p_l(t, m). \tag{6.4}$$

Then the m^*-th symbol interval in time slot t will be chosen as new symbol payload for rate increment.

Update state information. After choosing m^*-th symbol interval for payload, we update the state information for link l as follows:

- At transmit node i, if $s_l(i, t) =$ "IDLE", then set $s_l(i, t) =$ "TX". Set $z_l(t, m^*) = 1$.
- At receive node j, if $s_l(j, t) =$ "IDLE", then set $s_l(j, t) =$ "RX". Set $z_l(t, m^*) = 1$.

It is easy see that the payload-IA module is amenable to local implementation as all operations of this module (identifying the symbol interval set $\mathcal{M}$, selecting a symbol interval from $\mathcal{M}$ for payload, and updating state information) are performed at nodes i and j and their neighboring nodes.

[4]A tie can be handled by any tie-breaking rule, e.g., choosing the symbol interval with the minimum position.

6.3.6 Payload Adjustment Module

As described in Fig. 6.6, if the payload-IA module fails to increase the rate of the current link l, the payload adjustment module will be invoked. The goal is to increase link l's rate by one symbol payload through adjusting payload structures on links in $\mathcal{P}_l$ and $\mathcal{Q}_l$. In this module, for current link l, we first identify a set $\mathcal{R}$ of symbol intervals over all time slots in a frame on Tx(l) and Rx(l) that meet half-duplex constraints but fail to meet the PD-IA constraints. Then we consider a symbol interval in $\mathcal{R}$ iteratively (starting from the one that requires minimum adjustment) and attempt to make some payload adjustments on links in $\mathcal{P}_l$ and $\mathcal{Q}_l$, with the goal of turning the current symbol interval into an eligible interval. The module terminates once we turn an interval in $\mathcal{R}$ into an eligible interval or none of the symbol intervals in $\mathcal{R}$ works out. For the eligible interval, we set it to a payload and update the state information at nodes Tx(l) and Rx(l). A flow chart of the payload adjustment module is given in Fig. 6.9.

Finding a set of intervals for payload adjustment. Again we denote i and j as the transmit and receive nodes of the current link l, i.e., $i = \text{Tx}(l)$ and $j = \text{Rx}(l)$. For ease of explanation, denote (t, m) as the m-th symbol interval in time slot t.

At transmit node i, we first identify a set of unused symbol intervals over the entire frame that meets half-duplex requirement (as explained in the payload-IA module), which we denote as $\mathcal{R}_i$, i.e.,

$$\mathcal{R}_i = \{(t, m) : z_l(t, m) = 0, s_l(i, t) = \text{“TX” or “IDLE”}\}.$$

Likewise, at receive node j, we identify a set of unused symbol intervals over the entire frame that meets the half-duplex requirement, which we denote as $\mathcal{R}_j$, i.e.,

$$\mathcal{R}_j = \{(t, m) : z_l(t, m) = 0, s_l(j, t) = \text{“RX” or “IDLE”}\}.$$

We consider an unused symbol interval (t, m) and attempt to turn it into an eligible one only if it meets the half-duplex requirement at both transmit node i and receive node j. Denote $\mathcal{R}$ as the set of such unused symbol intervals in a whole frame. Then we have

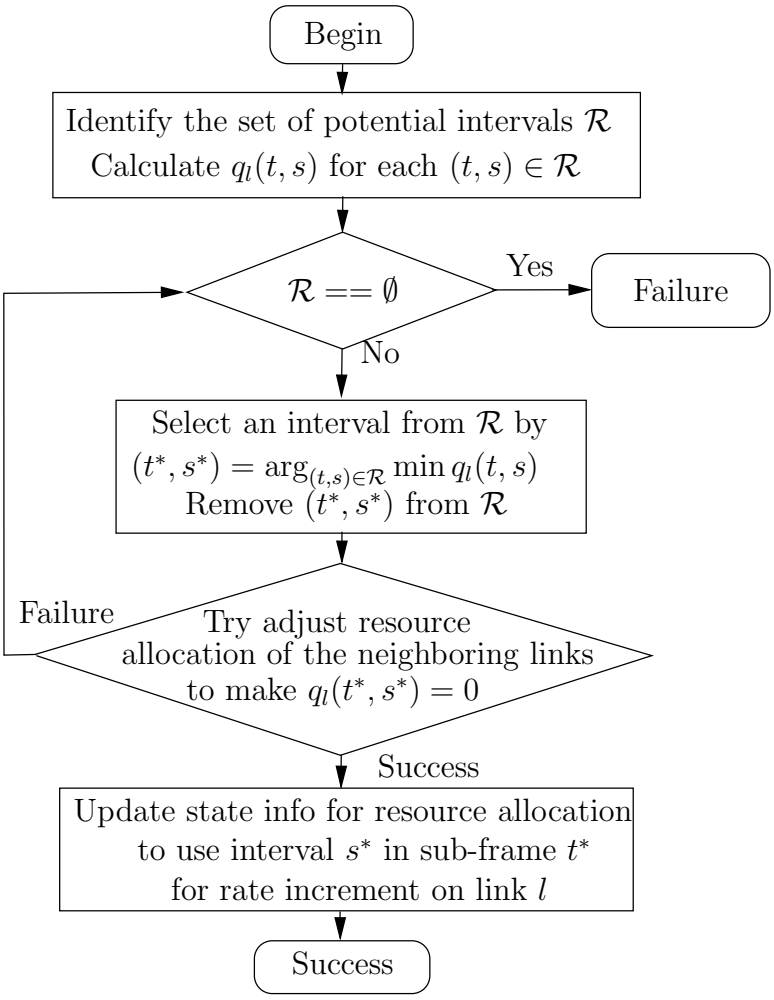


Figure 6.9: A flow chart of payload adjustment module.

$\mathcal{R} = \mathcal{R}_i \cap \mathcal{R}_j$. Based on the procedure of payload-IA module, we know that the (only) reason why a symbol interval in $\mathcal{R}$ is not eligible for payload is that it fails to meet the PD-IA constraints. To increase the rate of link l by one symbol payload, we attempt to consider a symbol interval in $\mathcal{R}$ one at a time and see if it can be turned into an eligible one by adjusting the current payload structures in sets $\mathcal{P}_l$ and $\mathcal{Q}_l$. Naturally, among the symbol intervals in $\mathcal{R}$, we start with the one that requires the minimum adjustment and so forth.

Suppose that the current unused symbol interval under consideration is (t, m). We now check how much payload adjustment on the links in $\mathcal{P}_l$ and $\mathcal{Q}_l$ is needed if we want to turn it to an eligible interval for payload. For transmit node i, its symbol interval (t, m) is

overlapping with symbol intervals $(t, g^{\mathrm{L}}_{lk}(m))$ and $(t, g^{\mathrm{R}}_{lk}(m))$ at $\mathrm{Rx}(k)$, $k \in \mathcal{Q}_l$. Should we set (t, m) to a payload at transmit node i, we need to move symbol payloads in intervals $(t, g^{\mathrm{L}}_{lk}(m))$ and $(t, g^{\mathrm{R}}_{lk}(m))$, if they indeed carry payload, to other unused symbol intervals for each link $k \in \mathcal{Q}_l$. Denote q^{tx}_l as the amount of required payload adjustment for setting a payload in interval $(t, m) \in \mathcal{R}$ on link l at $\mathrm{Tx}(l)$ (i.e., transmit node i), which we define as:

$$q^{\mathrm{tx}}_l(t, m) = \sum_{k \in \mathcal{Q}_l} \Big[z_k\Big(t, g^{\mathrm{L}}_{lk}(m)\Big) + z_k\Big(t, g^{\mathrm{R}}_{lk}(m)\Big)\Big].$$

Likewise, at receive node j, symbol interval (t, m) is overlapping with symbol intervals $(t, f^{\mathrm{L}}_{lk}(m))$ and $(t, f^{\mathrm{R}}_{lk}(m))$ from $\mathrm{Tx}(k)$, $k \in \mathcal{P}_l$. Should we set a payload in (t, m) at receive node j, we need to move symbol payload in intervals $(t, f^{\mathrm{L}}_{lk}(m))$ and $(t, f^{\mathrm{R}}_{lk}(m))$, if they indeed carry payload, to other unused symbol intervals for each link $k \in \mathcal{P}_l$. Denote $q^{\mathrm{rx}}_l(t, m)$ as the amount of required payload adjustment for setting a payload in interval $(t, m) \in \mathcal{R}$ on link l at $\mathrm{Rx}(l)$ (i.e., receive node j), which we define as:

$$q^{\mathrm{rx}}_l(t, m) = \sum_{k \in \mathcal{P}_l} \Big[z_k\Big(t, f^{\mathrm{L}}_{lk}(m)\Big) + z_k\Big(t, f^{\mathrm{R}}_{lk}(m)\Big)\Big].$$

Denote $q_l(t, m)$ as the total amount of required payload adjustment for setting a payload in interval $(t, m) \in \mathcal{R}$ on link l at both $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$. Then $q_l(t, m) = q^{\mathrm{tx}}_l(t, m) + q^{\mathrm{rx}}_l(t, m)$. Among all symbol intervals in $\mathcal{R}$, we choose a symbol interval (t, m) that has the smallest value of $q_l(t, m)$. Denote this symbol interval as (t^*, m^*). We have

$$(t^*, m^*) = \arg_{(t,m)\in\mathcal{R}} \min q_l(t, m).$$

Payload adjustment on links in $\mathcal{P}_l \cup \mathcal{Q}_l$. For interval (t^*, m^*), we try to make necessary payload adjustment on the links in $\mathcal{P}_l \cup \mathcal{Q}_l$ with an attempt to turn this interval to an eligible one. Since link k may be in both $\mathcal{P}_l$ and $\mathcal{Q}_l$, we explain the operations for payload adjustment on each link k by three cases: $k \in \mathcal{Q}_l \backslash \mathcal{P}_l$, $k \in \mathcal{P}_l \backslash \mathcal{Q}_l$, and $k \in \mathcal{P}_l \cap \mathcal{Q}_l$.

Case I: Consider link $k \in \mathcal{Q}_l \backslash \mathcal{P}_l$. At $\mathrm{Rx}(k)$, its symbol intervals $(t^*, g^{\mathrm{L}}_{lk}(m^*))$ and $(t^*, g^{\mathrm{R}}_{lk}(m^*))$ are overlapping with symbol interval (t^*, m^*) from $\mathrm{Tx}(l)$. Thus, there are at most two intervals on link $k \in \mathcal{Q}_l \backslash \mathcal{P}_l$ that need adjustment, i.e.,

$$\mathcal{S}^{\mathrm{tx}}_k = \{(t^*, n) : z_k(t^*, n) = 1, n \in \{g^{\mathrm{L}}_{lk}(m^*), g^{\mathrm{R}}_{lk}(m^*)\}\}.$$

If $\mathcal{S}_k^{\text{tx}} = \emptyset$, then no adjustment is needed. Otherwise, we perform the payload-IA module in Section 6.3.5 at nodes Tx(k) and Rx(k). If the payload-IA module successfully increases a payload symbol for link k, then we release a payload symbol on link k by setting: $z_k(t^*, n) = 0$ for $(t^*, n) \in \mathcal{S}_k^{\text{tx}}$ at both Tx(k) and Rx(k). We repeat the above operation until both payload symbols in $\mathcal{S}_k^{\text{tx}}$ are released.

Case II: Consider a link $k \in \mathcal{P}_l \backslash \mathcal{Q}_l$. At Rx($l$), its symbol interval (t^*, m^*) are overlapping with symbol intervals $(t^*, f_{lk}^{\text{L}}(m^*))$ and $(t^*, f_{lk}^{\text{R}}(m^*))$ from Tx(k). Thus, there are at most two intervals on link $k \in \mathcal{P}_l \backslash \mathcal{Q}_l$ that need adjustment, i.e.,

$$\mathcal{S}_k^{\text{rx}} = \{(t^*, n) : z_k(t^*, n) = 1, n \in \{f_{lk}^{\text{L}}(m^*), f_{lk}^{\text{R}}(m^*)\}\}.$$

Again, if $\mathcal{S}_k^{\text{rx}} = \emptyset$, then no adjustment is needed. Otherwise, we follow the same approach in Case I for the payload symbol relocation at nodes Tx(k) and Rx(k).

Case III: Consider link $k \in \mathcal{P}_l \cap \mathcal{Q}_l$. At Rx($k$), its symbol intervals $(t^*, g_{lk}^{\text{L}}(m^*))$ and $(t^*, g_{lk}^{\text{R}}(m^*))$ are overlapping with symbol interval (t^*, m^*) from Tx(l); at Rx(l), its symbol interval (t^*, m^*) are overlapping with symbol intervals $(t^*, f_{lk}^{\text{L}}(m^*))$ and $(t^*, f_{lk}^{\text{R}}(m^*))$ from Tx(k). Thus, there are at most four intervals on link $k \in \mathcal{P}_l \cap \mathcal{Q}_l$ that need adjustment, i.e.,

$$\mathcal{S}_k^{\text{tx,rx}} = \{(t^*, n) : z_k(t^*, n) = 1, n \in \{g_{lk}^{\text{L}}(m^*), g_{lk}^{\text{R}}(m^*), f_{lk}^{\text{L}}(m^*), f_{lk}^{\text{R}}(m^*)\}\}.$$

Again, if $\mathcal{S}_k^{\text{tx,rx}} = \emptyset$, then no adjustment is needed. Otherwise, we follow the same approach in Case I for the payload symbol relocation at nodes Tx(k) and Rx(k).

In the three cases, if any link $k \in \mathcal{P}_l \cup \mathcal{Q}_l$ fails to relocate its payload symbols, we remove symbol interval (t^*, m^*) from set $\mathcal{R}$ and consider the next symbol interval in $\mathcal{R}$, until the link rate is successfully increased or all symbol intervals in $\mathcal{R}$ are removed.

Update state information. If all the links in $\mathcal{P}_l \cup \mathcal{Q}_l$ adjust their payload structure successfully for symbol interval (t^*, m^*) on link l, then we update the state information for link l as follows:

- At transmit node i, if $s_l(i, t^*) =$ "IDLE", then set $s_l(i, t^*) =$ "TX". Set $z_l(t^*, m^*) = 1$.
- At receive node j, if $s_l(j, t^*) =$ "IDLE", then set $s_l(j, t^*) =$ "RX". Set $z_l(t^*, m^*) = 1$.

It is easy see that this module is amenable to local implementation as all operations of this module (identifying the potential payload symbols in $\mathcal{R}$, payload structure adjustment, and updating state information) are restricted on the selected link and its neighboring links.

6.3.7 Complexity and Overhead Analysis

Computational Complexity Analysis. We analyze the complexity for the scheduling algorithm at each node. (i) As we explained in Section 6.3.4, link ordering module can be done in a distributed fashion by employing the distributed ranking algorithm in [98], which has $O(N^3)$ complexity at each node. (ii) Consider the transmit/receive node of the *current* link l. In an iteration, its complexity associated with payload-IA and payload adjustment modules is $O(T^2M^2)$. Since the number of iterations on link l is bounded by $O(TM)$ and there are at most L links, we know that the total complexity on the transmit/receive node of link l is $O(LT^3M^3)$. (iii) The same transmit/receive node of link l are also involved in the payload-IA and payload adjustment modules when the iteration is working on a link $k \in \mathcal{P}_l \cup \mathcal{Q}_l$. Such complexity on the transmit/receive node of link l is $O(LT^3M^3)$. In summary, the total complexity at a (transmit/receive) node is $O(LT^3M^3 + N^3)$. Note that this complexity analysis is performed for the worst case by assuming no interference when calculating the number of iterations. In practical networks, there exists much interference in the network. As a result, the number of iterations in the algorithm will be decreased significantly. Based on current computational capability at a UWA node, we do not expect complexity issue with the proposed algorithm for practical purposes.

Overhead Analysis. As we showed in Section 6.3.4–6.3.6, the Shark-IA algorithm relies on the message exchange between neighboring nodes to achieve convergence. Since it is hard to precisely quantify overhead in a large network, we develop an upper bound for the total volume of message exchanges that may induced by the Shark-IA algorithm.

For link ordering module, message exchanges are required to sort the links in the network. As we explained in Section 6.3.4, link ordering module sorts the links by employing the distributed ranking algorithm in [98]. As per [98], link ordering module requires $O(N^2)$

message exchanges in the worst case. For payload-IA and payload adjustment modules, they are iterative modules and each of them has at most $O(TML)$ iterations. To characterize the total volume of their message exchanges, we first analyze the volume of their respective message exchanges in each iteration. For payload-IA module, nodes i and j need to check T time slots and the volume of message exchanges in each time slot is $O(1)$. So payload-IA module requires $O(T)$ message exchanges in an iteration. For payload-adjustment module, nodes i and j need to check T time slots and the volume of message exchanges in each time slot is $O(N)$. So payload-adjustment module requires $O(TN)$ message exchanges in an iteration. Since there are at most $O(TML)$ iterations, the total volume of message exchanges required by payload-IA and payload adjustment modules is $O(T^2MNL)$.

By adding the message exchanges induced by the three modules together, the Shark-IA algorithm requires $O(T^2MNL+N^2)$ message exchanges in the worst case. But in a practical network, the algorithm will converge much faster than $O(T^2MNL)$ due to the interference and half-duplex constraints and thus will incur much less message overhead.

6.4 Performance Evaluation

To evaluate the performance of Shark-IA algorithm, we first compare it to an idealized benchmark algorithm with perfect scheduling and zero propagation delays. We formulate it to an optimization problem and denote it as OPT-noIA. We also compare Shark-IA algorithm to a centralized algorithm with perfect PD-IA scheduling. We formulate it to another optimization problem and denote it as OPT-IA. In what follows, we first formulate the two performance benchmarks (OPT-noIA and OPT-IA), and then offer numerical results for the performance evaluation of our Shark-IA algorithm.

6.4.1 Two Performance Benchmarks

Perfect Scheduling with Zero Propagation Delays. In the literature, research efforts have been spent on the design of underwater MAC protocols with the aim of alleviating

the ill effect of large propagation delay [28, 47, 69]. However, the philosophy of these MAC protocols is to fight the propagation delay (rather than leveraging the propagation delay) and thus their throughput performance is bounded by that under delay-free model. Therefore, to evaluate the throughput performance of our distributed PD-IA scheduling algorithm, we compare it to the throughput performance of the same problem but without using PD-IA under the delay-free model. We denote it as OPT-noIA, which is formulated as follows.

Half-Duplex Constraints. In our scheduling, a node cannot be a transmitter and a receiver in the same time slot. Denote $\alpha_i(t)$ as the indicator of a transmitter for node i in time slot t. That is, $\alpha_i(t) = 1$ if node i is a transmitter in time slot t and $\alpha_i(t) = 0$ otherwise. Similarly, denote $\beta_i(t)$ as the indicator of a receiver for node i in time slot t, That is, $\beta_i(t) = 1$ if node i is a receiver in time slot t and $\beta_i(t) = 0$ otherwise. Then we have

$$\alpha_i(t) + \beta_i(t) \leq 1, \quad 1 \leq i \leq N, 1 \leq t \leq T. \tag{6.5}$$

Link Activity Constraints. Consider a node in a given time slot. If it is a transmitter, then its outgoing links can be active but its incoming links are forced to be inactive; if it is a receiver, then its incoming links can be active but its outgoing links are forced to be inactive; otherwise (being idle), all of its outgoing/incoming links are forced to be inactive. Denote $\mathcal{L}_i^{\text{in}}$ as the set of incoming links to node i and $\mathcal{L}_i^{\text{out}}$ as the set of outgoing links from node i. Denote $x_l(t)$ as the activity of link l in time slot t, i.e., $x_l(t) = 1$ if link l is scheduled to be active in time slot t and $x_l(t) = 0$ otherwise. Then we have

$$\frac{1}{L} \sum_{l \in \mathcal{L}_i^{\text{out}}} x_l(t) \leq \alpha_i(t), \qquad 1 \leq i \leq N, 1 \leq t \leq T. \tag{6.6}$$

$$\frac{1}{L} \sum_{l \in \mathcal{L}_i^{\text{in}}} x_l(t) \leq \beta_i(t), \qquad 1 \leq i \leq N, 1 \leq t \leq T. \tag{6.7}$$

Consider a link in a given time slot. If it is active, then the symbols on this link can be used for payload; otherwise, all of the symbols on this link cannot be used for payload. Thus, we have

$$z_l(t, m) \leq x_l(t), \qquad 1 \leq l \leq L, 1 \leq t \leq T, 1 \leq m \leq M. \tag{6.8}$$

Symbol Activity Constraints. We model the physical-layer behavior of the (OFDM) symbols on each link in each time slot under the delay-free model. Consider symbol m on link l in a given time slot. If it is used for payload, then symbol m on link $k \in \mathcal{Q}_l$ cannot be used for payload due to the interference conflict; otherwise, there is no constraint on the activity of this symbol. Thus we have

$$z_l(t,m) + \frac{1}{L}\sum_{k\in\mathcal{Q}_l} z_k(t,m) \le 1, \qquad l\in\mathcal{L}, 1 \le m \le M, 1 \le t \le T. \tag{6.9}$$

Link Capacity Constraints. For simplicity, we also normalize the time duration of a frame to one unit. We assume that fixed modulation and coding scheme (MCS) is used for data transmission at payload symbols and each payload symbol carries one data unit. Denote c_l as the data rate of link l in a frame. Then we have

$$c_l = \sum_{t=1}^{T}\sum_{m=1}^{M} z_l(t,m) \qquad l \in \mathcal{L}, 1 \le t \le T. \tag{6.10}$$

For each link l, its aggregate data rate attributed to the sessions cannot exceed its achievable data rate. Thus, we have

$$\sum_{f=1}^{F} r_l(f) \le c_l. \quad 1 \le l \le L. \tag{6.11}$$

Flow Balance Constraints. Denote $r(f)$ as the data rate of session f. For the traffic flow from a source to its destination, we denote $r_l(f)$ as the amount of data rate on link l that is attributed to session f. At each node, flow conservation must be observed.

At a source node, we have

$$\sum_{l\in\mathcal{L}_i^{\text{out}}} r_l(f) = r(f), \qquad i = \text{src}(f), 1 \le f \le F. \tag{6.12}$$

At an intermediate relay node, we have

$$\sum_{l\in\mathcal{L}_i^{\text{in}}} r_l(f) = \sum_{l\in\mathcal{L}_i^{\text{out}}} r_l(f), \qquad 1 \le i \le N, i \ne \text{src}(f), i \ne \text{dst}(f), 1 \le f \le F. \tag{6.13}$$

At a destination node, we have

$$\sum_{l\in\mathcal{L}_i^{\text{in}}} r_l(f) = r(f), \qquad i = \text{dst}(f), 1 \le f \le F. \tag{6.14}$$

It can be easily verified that if (6.12) and (6.13) are satisfied, then (6.14) is also satisfied. Therefore, it is sufficient to include (6.12) and (6.13) in the problem formulation.

Throughput Constraints. Denote $r_{\min}$ as the throughput rate of the bottleneck session. Then we have

$$r_{\min} \leq r(f), \quad 1 \leq f \leq F. \tag{6.15}$$

By combining the delay-free model with the cross-layer constraints, the throughput optimization problem in a centralized network without using PD-IA technique can be formulated as follows:

OPT-noIA:	max	$r_{\min}$
	s.t.	Half-duplex constraints: (6.5);
		Link activity constraints: (6.6)–(6.8);
		Symbol activity constraints: (6.9);
		Link capacity constraints: (6.10)–(6.11);
		Flow balance constraints: (6.12)–(6.13);
		Throughput constraints: (6.15).

This formulation is in the form of a mixed integer linear programming (MILP), which is NP-hard in general. However, we find that the OPT-noIA problem for a mid-size network (e.g., a network with 50 nodes) can be solved by CPLEX [104]. We use the optimal objective value of OPT-noIA as the benchmark when evaluating the throughput performance of our distributed PD-IA scheduling algorithm.

Perfect PD-IA Scheduling. The throughput optimization problem in a centralized network with PD-IA, denoted as OPT-IA, can be formulated following the similar procedure as in OPT-noIA. The only difference between OPT-IA and OPT-noIA is the symbol activity constraints. In OPT-IA, the payload scheduling over the OFDM symbols must conform to the PD-IA constraints developed in Section 6.2.2. By replacing (6.9) with PD-IA constraints (6.1)–(6.2), the throughput optimization problem in a centralized network using PD-IA technique can be formulated as follows:

OPT-IA:	max	$r_{\min}$
	s.t.	Half-duplex constraints: (6.5);
		Link activity constraints: (6.6)–(6.8);
		PD-IA constraints: (6.1)–(6.2);
		Link capacity constraints: (6.10)–(6.11);
		Flow balance constraints: (6.12)–(6.13);
		Throughput constraints: (6.15).

This formulation is also in the form of MILP, which is NP-hard in general. Although OPT-IA shares most of constraints with OPT-noIA, it turns out that OPT-IA is much more difficult to solve than OPT-noIA. Our simulation shows that for a 50-node network instance, OPT-IA cannot be solved by cplex solver in reasonable amount of time (24 hours) when the session size is greater than three.

6.4.2 Simulation Setting

We consider networks with 50 nodes being randomly deployed in a 5 km by 5 km area. Among the nodes in the network, there is a set of active sessions with their source and destination nodes being randomly selected among all the nodes. The route from the source node of a session to its destination node is found by the AODV algorithm [70].

We assume that all the nodes have the same transmission range 1 km.[5] At a receiving node, we assume that the interference is negligible if the power of the interference is less than -20 dB of the power of its desired signal. Therefore, we set the interference range of a node to 4 km based on the relationship between path loss and distance in underwater acoustic environment [54, Figure 6]. A frame has $T = 10$ time slots, each of which is comprised of a guard interval and $M = 50$ OFDM symbols (see Fig. 6.3). For each OFDM symbol, we use the same parameters as the "VHF08 EXPERIMENT" in [50], i.e., an OFDM symbol is of 85.5 ms time duration (with a CP of 20 ms time duration). For simplicity, we normalize

[5]Data transmission over 1 km is short/medium range communication in underwater acoustic sensor networks [2].

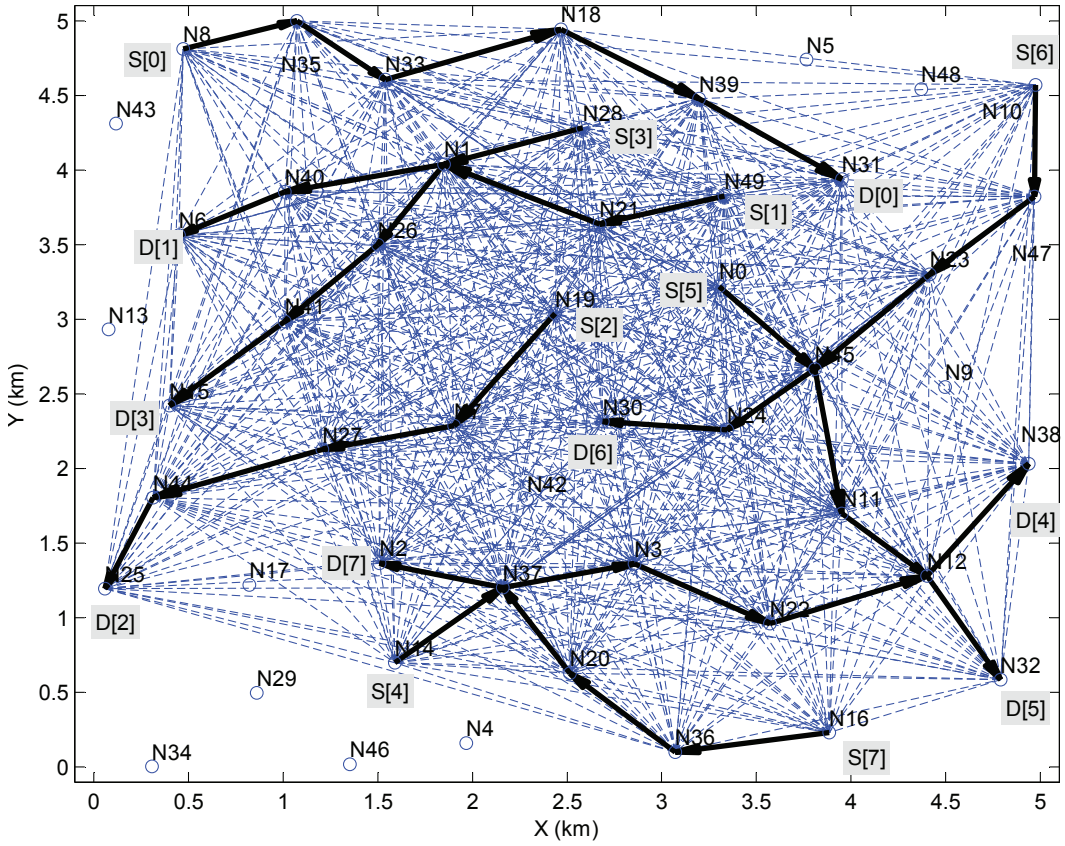


Figure 6.10: The topology and routing for a network instance.

the time duration of a frame to one unit. We assume that fixed MCS is used for data transmission at payload (OFDM) symbols and each payload (OFDM) symbol carries one data unit.

6.4.3 A Case Study

Before presenting our extensive simulation results, we first show results for one network instance in Fig. 6.10. In this figure, a solid arrow line represents a link while a dashed line represents a potential interference. There are 8 sessions in this network, as shown in Fig. 6.10. We apply our proposed distributed scheduling algorithm to this network instance. The obtained objective value is 25. We then solve OPT-noIA problem by CPLEX and have an optimal objective value of 18. This implies that our PD-IA distributed scheduling

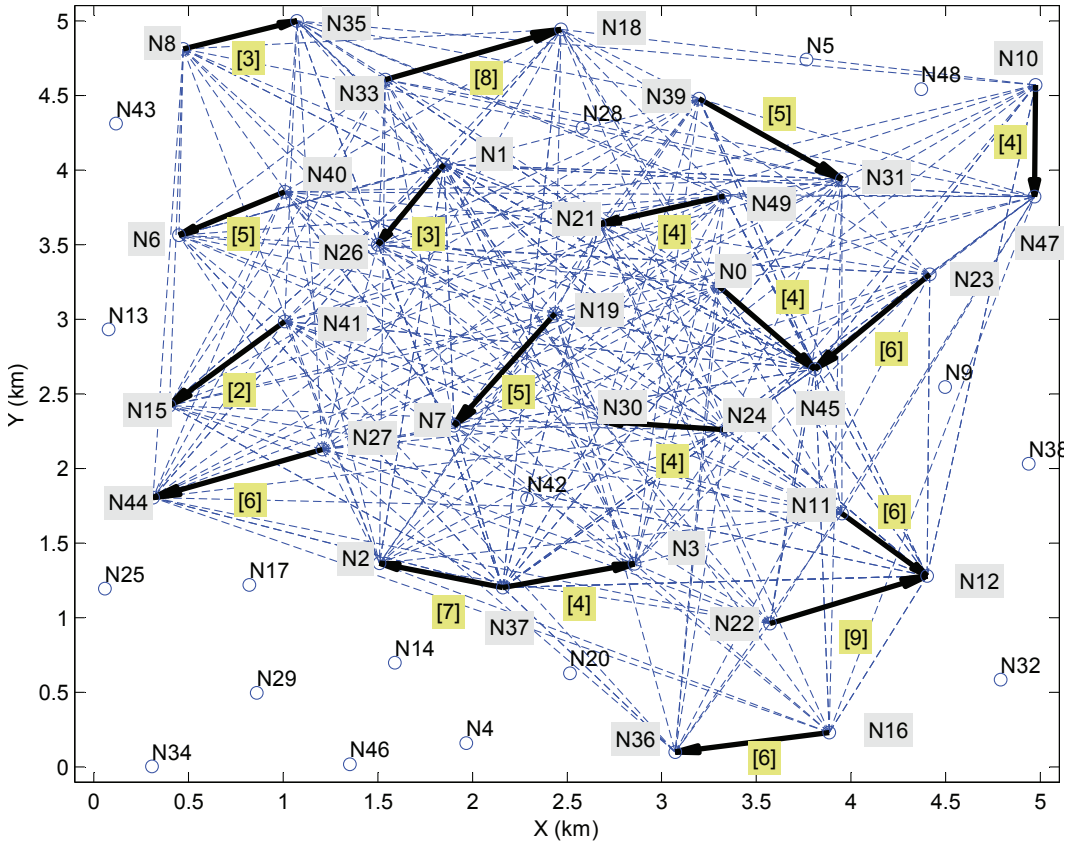


Figure 6.11: The scheduling pattern in the first time slot.

algorithm can increase the network throughput by 38.9% when compared to PD-noIA.

We now show some details in the solutions. Fig. 6.11 shows the scheduling results in the first time slot, with the number of payload symbols shown in square brackets next to the active links. Table 6.3 summarizes the payload scheduling information in this time slot. In the table, the first column lists the receive node of each active link; the second column lists the number of payload symbols; the third column lists the number of its unused symbol intervals, which is 50 minus the number in the second column; the fourth column lists the number of interfering symbols (from neighboring interfering transmit nodes); and the fifth column lists the interference overlapping ratio, which is the ratio of the fourth column to the third column. In the fifth column, a value greater than 1 indicates the existence of interference overlapping. The larger the ratio is, the more PD-IA has been exploited by the

Table 6.3: The scheduling results in the first time slot.

Active receiver	# of payload symbols	# of unused symbols	# of interfering symbols	Interference overlapping ratio
$\mathrm{Rx}(N_{19} \rightarrow N_7)$	5	45	86	1.91
$\mathrm{Rx}(N_8 \rightarrow N_{35})$	3	47	67	1.42
$\mathrm{Rx}(N_{33} \rightarrow N_{18})$	8	42	68	1.62
$\mathrm{Rx}(N_{39} \rightarrow N_{31})$	5	45	86	1.91
$\mathrm{Rx}(N_{49} \rightarrow N_{21})$	4	46	87	1.89
$\mathrm{Rx}(N_{40} \rightarrow N_6)$	5	45	67	1.49
$\mathrm{Rx}(N_{27} \rightarrow N_{44})$	6	44	75	1.70
$\mathrm{Rx}(N_1 \rightarrow N_{26})$	3	47	82	1.74
$\mathrm{Rx}(N_{41} \rightarrow N_{15})$	2	48	73	1.52
$\mathrm{Rx}(N_{37} \rightarrow N_3)$	4	46	84	1.83
$\mathrm{Rx}(N_{22} \rightarrow N_{12})$	9	41	66	1.61
$\mathrm{Rx}(N_0 \rightarrow N_{45})$	4	46	87	1.89
$\mathrm{Rx}(N_{11} \rightarrow N_{12})$	6	44	69	1.57
$\mathrm{Rx}(N_{10} \rightarrow N_{47})$	4	46	76	1.65
$\mathrm{Rx}(N_{23} \rightarrow N_{45})$	6	44	85	1.93
$\mathrm{Rx}(N_{24} \rightarrow N_{30})$	4	46	87	1.89
$\mathrm{Rx}(N_{16} \rightarrow N_{36})$	6	44	57	1.30
$\mathrm{Rx}(N_{37} \rightarrow N_2)$	7	43	80	1.86

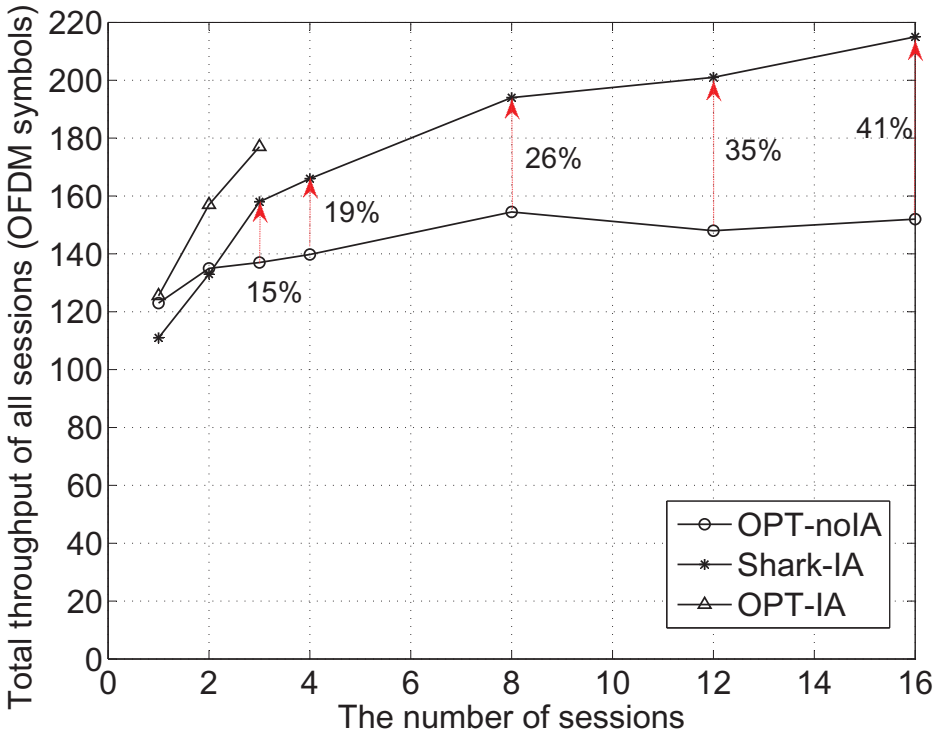


Figure 6.12: Comparison of Shark-IA, OPT-noIA, and OPT-IA.

algorithm.

Now let's take a look at the first row (for receive node $\mathrm{Rx}(N_{19}, N_7)$) in Table 6.3 as an example. As shown in Fig. 6.11, receive node N_7 is within the interference range of all transmit nodes in the network. The number of interfering payload symbols at receive node N_6 is 86. Since 5 symbol intervals have been used for receiving payloads from intended transmit node N_{19}, there are only 45 $(= 50 - 5)$ unused symbol intervals in which these interfering symbols may fall. That is, the interference overlapping ratio is $86/45 \approx 1.91$ at receiver N_7.

6.4.4 Complete Results

We consider 7 cases with the above network setting: 1 session, 2 sessions, 3 sessions, 4 sessions, 8 sessions, 12 sessions, and 16 sessions. For each case, we study 100 network instances to obtain their average throughput. Figure 6.12 exhibits our simulation results,

with x-axis being the number of sessions and y-axis being the total throughput of all sessions (i.e., average objective value $\times$ the number of sessions). Note that when the number of sessions is greater than 3, the OPT-IA cannot be solved in acceptable amount of time (24 hours per network instance on Blueridge supercomputer at VT).

The simulation results yield the following conclusions: First, Shark-IA significantly outperforms the OPT-noIA when the number of sessions is greater than two. Second, the throughput gain of Shark-IA over OPT-noIA increases as the traffic in the network becomes more intensive. Third, when the number of sessions is small (≤ 3), Shark-IA can achieve more than 80% optimal throughput of the centralized solution (by OPT-IA). Finally, for the network with more than 3 sessions, the optimal centralized solution (OPT-IA) cannot be obtained in reasonable amount of time, while Shark-IA can yield a competitive solution very quickly.

6.5 Related Work

In UWA networks, the phenomenon of large propagation delays is a fundamental issue that adversely affects the network throughput performance. Efforts have been spent to alleviate this ill effect in various ways. In [61], Molins *et al.* proposed a MAC protocol to address the large propagation delay and low efficiency problems in UWA networks. The protocol was based on slotted floor acquisition multiple access (S-FAMA) which combined both carrier sensing and a conversion between the sender and the sinker prior to data transmission. In [69], Peleato and Stojanovic developed a distance-aware channel access protocol based on RTS/CTS shaking. By taking into account propagation delays, their developed protocol was free of collision and thus outperformed CS-ALOHA and slotted FAMA. In [47], Kredo et al. proposed a staggered TDMA-based scheduling algorithm (viz. STUMP) for UWA networks, with the objective of increasing channel utilization by taking into account propagation delays. They showed that their scheduling algorithm was superior to the conventional TDMA-based scheduling in terms of throughput. In [28], Guo et al. proposed a RTS/CTS-based protocol for the networks with large propagation delays. With the information of propagation delays,

the protocol enabled simultaneous data transmissions among the nodes while avoiding collision. In [67], Noh et al. proposed the DOTS protocol, in which the node uses neighbors' propagation delay map and the expected transmission schedules to increase the chances of concurrent transmissions while reducing the likelihood of collisions. In [30], Han et al. proposed a multi-session floor acquisition multiple access (M-FAMA) algorithm based on the RTS/CTS/ACK shaking mechanism. This protocol enabled the senders to initiate multiple concurrent sessions while avoiding collisions by calculating their neighbors' transmission schedules and propagation delays. Although all these efforts [28, 30, 47, 61, 67, 69] considered propagation delays in the design of their algorithms, none of them offered a systematic and discipled approach to fully exploit propagation delays for throughput maximization.

PD-IA is an effective technique that leverages propagation delays to our advantage for throughput maximization. The concept of IA was coined by Jafar and Shamai for the two-user X channel [39] and its huge potential benefits were substantiated in a seminar paper [9]. As a specific form of IA, PD-IA has been studied in [8, 13, 27]. In [27], Grokop et al. studied the PD-IA from the information theoretic perspective. They showed that by using PD-IA, the spectral efficiency of the K-user interference channel can linearly grow with K. However, their result is based on an unpractical assumption of the bandwidth scaling K in order $O(K^{2K^2})$. A similar result was obtained by Cadambe and Jafar [8], based on the assumption of infinite bandwidth. In [13], Chitre et al. explored the probability to improve the network throughput by exploiting propagation delays. They developed a scheduling algorithm to harvest the benefits of propagation delays. But their results were limited to single-collision domain and single-hop scenario. Although all these efforts exploited propagation delays to improve the network throughput, their results are limited in physical-layer and single-hop scenario. In this chapter, we advance PD-IA a further step by studying the PD-IA in multi-hop networks.

6.6 Chapter Summary

In this chapter, we exploited large propagation delays in UWA networks as an advantage instead of adversary. We developed a PD-IA model that specifies a set of constraints to ensure feasibility of PD-IA at the physical layer. Based on this model, we studied a network throughput optimization problem with the goal of maximizing the minimum rate among a set of sessions. We developed a distributed PD-IA scheduling algorithm (Shark-IA) that iteratively increases payloads in a time frame so that at each receiver, (i) the payload symbols from its intended transmitter can be received free of interference, while (ii) the interfering payload symbols from its unintended transmitters can maximally overlap. We validated the performance of Shark-IA and found significant throughput gains when compared to the case with perfect scheduling and zero propagation delays. More importantly, we found that the throughput gain increases with the traffic intensity in the network, which shows that higher traffic intensity in the network can actually help increase the opportunity to achieve PD-IA.

Chapter 7

Interference Neutralization in Multi-hop Networks

7.1 Introduction

Traditional paradigm for interference management in wireless networks is avoidance, where interference is avoided by orthogonalizing channel access, either in time, frequency, space, or code. Recent advances in physical (PHY) layer technologies allow a shift in paradigm toward interference exploitation, which allows multiple concurrent independent transmissions in an interference domain. This is made possible through the use of some powerful PHY layer techniques (e.g., interference cancellation (IC), interference alignment (IA)) at a node's transceiver. This chapter studies *interference neutralization* (IN), which is a fairly new technique in the interference exploitation family. IN refers to a joint design of signals at the transmitters, so that these transmit signals nullify themselves in the air at their unintended receivers while remaining resolvable at their intended receivers. To achieve the interference nullification in the air, IN requires that multiple transmitters have the same data that is under transmission, making it uniquely suitable for multi-hop wireless networks. Note that IN differs from transmitter-side IC, which projects interference to a perpendicular direction of the desired signal at the unintended receivers (rather than nullifying interference in the

air). IN is also different from IA. Although both of them require a joint signal design at the transmitters, the signal design of IN is to nullify interference in the air, while the signal design of IA is to align interference to the same direction.

Consider the three transmit nodes and six receive nodes as shown in Fig. 7.1. Each node only has a single antenna. Suppose that the three transmit nodes $\{T_1, T_2, T_3\}$ have the same data information for transmission, which we denote as x. Also suppose that $\{R_1, R_2, R_3, R_4\}$ are the intended receivers of x in the next hop and $\{R_5, R_6\}$ are the unintended receivers. Further, we require that the interference on $\{R_5, R_6\}$ from $\{T_1, T_2, T_3\}$ be nullified.

Denote h_{ji} as the channel coefficient between receive node j and transmit node i, which is a complex number. Denote u_i as the precoding coefficient at transmit node i, which is a complex number as well. Denote y_j as the received signal (desired signal or interference) at receive node j. Then we have

$$y_j = \Big(h_{j1}u_1 + h_{j2}u_2 + h_{j3}u_3\Big)x,$$

for each of the six receivers R_j, $j = 1, 2, \cdots, 6$. We now show that through carefully design of the precoding coefficients at the three transmitters $\{T_1, T_2, T_3\}$, the interference at the two unintended receive nodes $\{R_5, R_6\}$ can be neutralized while the desired signal at the four intended receivers $\{R_1, R_2, R_3, R_4\}$ can be decoded successfully. To do so, we need to show that there exist precoding coefficients (u_1, u_2, and u_3) that satisfy the following four constraints:

$$R_1: \quad |h_{11}u_1 + h_{12}u_2 + h_{13}u_3| > 0, \tag{7.1a}$$

$$R_2: \quad |h_{21}u_1 + h_{22}u_2 + h_{23}u_3| > 0, \tag{7.1b}$$

$$R_3: \quad |h_{31}u_1 + h_{32}u_2 + h_{33}u_3| > 0, \tag{7.1c}$$

$$R_4: \quad |h_{41}u_1 + h_{42}u_2 + h_{43}u_3| > 0, \tag{7.1d}$$

$$R_5: \quad h_{51}u_1 + h_{52}u_2 + h_{53}u_3 = 0, \tag{7.1e}$$

$$R_6: \quad h_{61}u_1 + h_{62}u_2 + h_{63}u_3 = 0. \tag{7.1f}$$

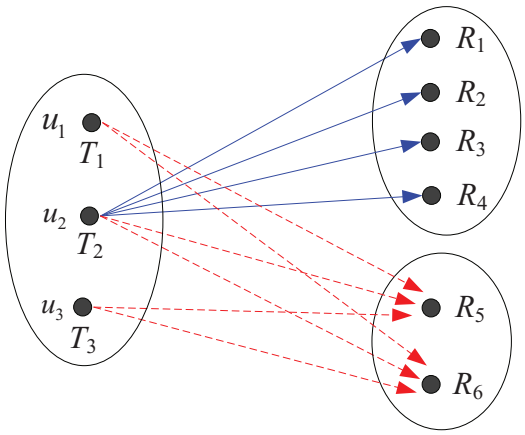


Figure 7.1: An example that illustrates IN capabilities and its application.

Given that the channel coefficients h_{ji} ($j = 1, 2, \cdots, 6$ and $i = 1, 2, 3$) are independent of each other, it is not difficult to find a set of precoding coefficients u_1, u_2, u_3 to meet the six linear constraints. In fact, it can be shown that as long as (i) u_1, u_2, or u_3 is nonzero [96], and (ii) the set of coefficients u_1, u_2, u_3 satisfies (7.1e)–(7.1f), then the same set of u_1, u_2, u_3 will satisfy (7.1a)–(7.1d) almost surely. Therefore, to find a feasible solution to (7.1), one only need to focus on (7.1e)–(7.1f). For tow equations with three variables, there exist non-unique feasible solutions. For example, we can set $u_1 = 1$ and solve u_2 and u_3 in (7.1e)–(7.1f). We have

$$\begin{aligned} u_1 &= 1, \\ u_2 &= (h_{61}h_{53} - h_{63}h_{51})/(h_{63}h_{52} - h_{62}h_{53}), \\ u_3 &= (h_{61}h_{52} - h_{62}h_{51})/(h_{62}h_{53} - h_{52}h_{63}). \end{aligned}$$

It is easy to verify that this set of precoding coefficients satisfy all six constraints in (7.1). As a result, the interference from T_1, T_2, and T_3 are effectively neutralize at R_5 and R_6 while the same signal is received successfully at R_1, R_2, R_3 and R_4. In general, as we shall shown in Section 7.1, for K transmitters, we can neutralize their interference to $(K-1)$ unintended receivers while there is no limit on the number of intended receivers.

Conceptually, IN resembles distributed MIMO in the sense that IN exploits precoding efficients at the transmitters to achieve interference nullification at a select subset of receivers.

This is true. But given that historically, the goal of distributed MIMO was mainly to maximize the SINR of users in a cellular network (instead of nullifying interference at a select subset of receivers), we choose to use a separate terminology – IN in this chapter to contrast its objective from that of distributed MIMO.

The next question is: why is IN useful? We answer this question by applying IN to a multi-hop wireless network. Consider the multi-hop network in Fig. 7.2 as an example. There are three sessions in the network: $S_1 \rightarrow D_1$, $S_2 \rightarrow D_2$, and $S_3 \rightarrow D_3$, respective, each going through multi-hop transmissions inside the network. Inside the network, due to broadcast advantage of wireless channel, there will be multiple nodes overhearing the same message from one of the source nodes and each of them can help relay the message to its next-hop nodes. In this regard, consider the scenario in Fig. 7.2, where the set of nodes in $\mathcal{T}$ can send information to the set of nodes in $\mathcal{R}$ in one hop. Within the set of nodes in $\mathcal{T}$, there are three subsets of nodes, differentiated by the legends ★, ■, and ▲, each of which has message from S_1, S_2, and S_3, respectively. By using IN, the three subsets of nodes in $\mathcal{T}$ can send their respective messages to their intended receivers in $\mathcal{R}$ (marked with the three different legends) *simultaneously, in one time slot (instead of three)*. Inside $\mathcal{R}$, three subsets of nodes will decode their own desired incoming messages successfully since the undesired messages can be neutralized by using appropriate precoding coefficients at the transmit nodes in $\mathcal{T}$.

The above example illustrates the potential benefits of IN, i.e., allowing simultaneous transmissions of different messages among different group of nodes by nullifying different interferences. To date, all prior works on IN have been limited to simple and specialized network configurations, such as $2\times2\times2$ network in [25] and ZZ network in [58]. It is not clear IN can be systematically exploited in a general network setting, such as a multi-hop network shown in Fig. 7.2. This knowledge gap underlines both the technical depth of this problem and the critical need to bridge this gap. The goal of this chapter is to make a concrete step toward advancing IN technique in a general multi-hop network. The main contributions of this chapter are summarized as follows:

- We establish a reference model for IN. Under the reference model, we derive the maxi-

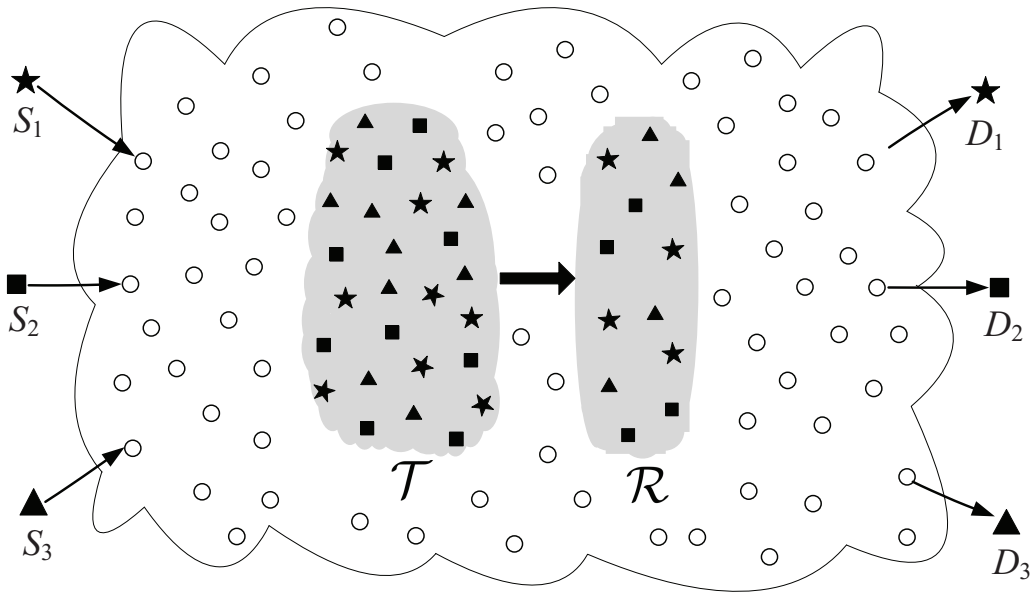


Figure 7.2: An example that illustrates IN in a multi-hop network.

mum number of unintended receivers whose interference can be effectively neutralized. By applying the IN reference model to a set of links, we derive a set of feasibility constraints for a subset of links that can be active simultaneously.

- We show how IN can enable simultaneous activation of transmissions link in an ad hoc network by taking advantage of broadcast nature of a wireless channel and the availability of idle nodes. We introduce a concept called *neut* to represent those idle nodes that can be exploited for IN.
- Based on the notion of neut, we study IN in a general multi-hop network. We show the core component of the problem is to select an optimal set of neuts along each hop of a session's path. Subsequently, we develop the necessary mathematical models (constraints) that characterize neut selection, IN, and scheduling. Collectively, these constraints allow us to determine which subsets of links can be active simultaneously without the need of getting involved into the complex IN signal design at the PHY layer.
- As an application, we apply our multi-hop IN model to study a throughput maximiza-

tion problem. For performance evaluation, we compare the throughput of the network with IN against that without IN. Simulation results show that the use of IN can generally increase throughput. We find that the throughput gain is most profound when the network is dense and the number of idle nodes that can be selected as neuts is abundant.

The remainder of this chapter is organized as follows. In Section 7.2, we review related work. In Section 7.3, we establish a reference model for IN. Based on this reference model, we characterize the feasibility constraints for a set of active links. In Section 7.4, we develop a mathematical model for IN in a multi-hop network. In Section 7.5, we apply our IN model to study a throughput maximization problem. Section 7.6 presents performance evaluation results. Section 7.7 concludes this chapter.

7.2 Related Work

The IN terminology was recently coined by Mohajer et al. in [58, 59, 60] when studying two-hop relay networks. However, a similar idea has been around for many years under different names such as multiuser zero-forcing, distributed zero-forcing, and orthogonalize-and-forward (see, e.g., [3, 72]). Due to the requirement that multiple transmitters should have the same data information from the source, most research on IN has been focused on two-hop (relay-aided) networks. In [58], Mohajer et al. proposed an IN scheme for a special two-hop relay network (called ZZ network) and showed that their IN scheme can convey the maximum amount of information under deterministic channel models. Later, they applied IN to the same network, but under Gaussian channel model, to study the approximate network capacity in [59, 60]. In [31], Ho and Jorswieck studied IN in the instantaneous interference relay channel and derived an achievable rate region for this channel when IN was employed at relay nodes. In [72], Rankov and Wittneben studied a $K \times N \times K$ relay interference network and showed that it required at least $K(K-1)+1$ relay nodes to achieve IN at destination nodes.

Another research thread in IN has been focused on the aligned interference neutralization

Table 7.1: Notation for IN in multi-hop networks.

Symbol	Definition
N	The number of nodes in the network
T	The number of time slots in a time frame
$\mathcal{N}$	The set of nodes in the network
$\mathcal{N}_{\mathrm{s}}$	The set of nodes involved in the routing paths
$\mathcal{N}_{\mathrm{idle}}$	The set of nodes not involved in the routing paths
$\mathcal{L}_{\mathrm{s}}$	The set of links along the routing paths
$\mathcal{S}$	The set of sessions in the network
$\mathcal{S}_l$	The set of sessions that traverse link l
$\mathcal{T}_l$	The set of transmit nodes of link l
$\mathcal{R}_l$	The set of receive nodes of link l
$\mathcal{I}_q$	The set of idle nodes that may be node q's neuts
$\mathcal{A}_q$	The set of idle nodes that may serve as neut for node q, plus node q itself
$\mathcal{B}_q$	The set of node q's neuts, plus node q itself
$\mathrm{Tx}(l)$	The transmitter of link l
$\mathrm{Rx}(l)$	The receiver of link l
$\mathcal{M}_i$	The set of path nodes for whom node i can be a neut
h_{ji}	The channel coefficient between node j and node i
u_i	The precoding coefficient for signal transmission at node i
$\lambda_{i\to q}$	A binary indicator whether node i serves as node q's neut
$\gamma_{j\to q}(t)$	An optimization variable for reformulation
$r(s)$	The data rate of session f
$\alpha_l(t)$	A binary indicator whether link l is active in time slot t
$\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$	A binary indicator whether $\mathcal{B}_{\mathrm{Tx}(l)}$ neutralize their interferences to node j
$\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$	A binary indicator whether node j is interfered with by $\mathcal{B}_{\mathrm{Tx}(l)}$
$r_{\min}$	The minimum rate of the sessions in the network

(AIN) in two-hop interference MIMO networks. In [25], Gou et al. showed that the use of AIN with two-symbol extension allowed the $2 \times 2 \times 2$ interference channel to achieve the min-cut outer bound value of 2 DoFs. A similar AIN scheme was employed by Lee and Wang in [49], where they showed that AIN can achieve significant DoF gain in a two-user network with instantaneous relay. In [91], Vaze and Varanasi studied the DoF region of the $2 \times 2 \times 2$ relay network and showed that an AIN-based scheme can achieve the min-cut DoF outer-bound, no matter how many antennas the nodes have.

It is worth pointing out that there exists a major difference between IN and CoMP (or distributed MIMO) in cellular networks [73], although both of them requires multiple transmitters having the same data information for transmission. For CoMP, the purpose of signal design at the transmitters is to improve the SINR of the cell-edge users by turning the interference to the useful signals [73]. In contrast, for IN in multi-hop networks, the purpose of signal design at the transmitters is to nullify their interference at their unintended receivers. In addition, CoMP was designed for one-hop cellular networks while IN works best for dense ad hoc networks, as we shall show in our performance evaluation.

To date, the study of IN is mainly limited to the information theory (IT) community and, to the best of our knowledge, there is no result of IN in the networking community. The goal of this chapter is to make a concrete step towards advancing IN from the networking perspective.

7.3 Feasibility Constraints for Interfering Links in a Single Hop

IN is a transmitter-side interference management technique. It requires a set of transmitters with the same message to transmit to another set of receivers. By setting the precoding coefficients at the set of transmitters, the transmitted signals can nullify themselves in the air at another set of unintended receivers while remaining decodable at their intended set of receivers. In this section, we study the feasibility conditions (or constraints) that IN will

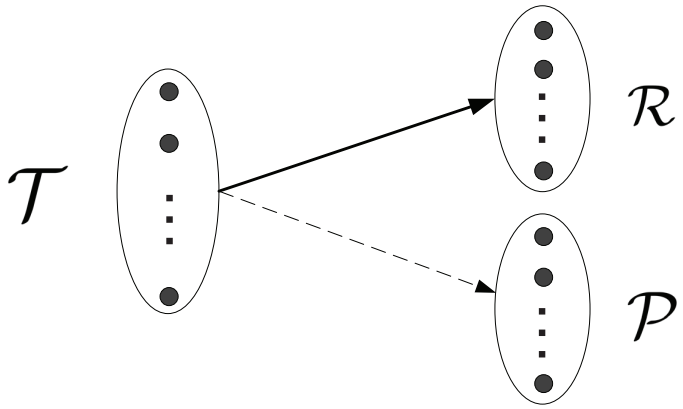


Figure 7.3: A reference model for IN.

work in a single hop transmission. Table 7.1 lists the notation that we use in this chapter.

A Reference Model and Basic Result. Figure 7.3 shows the reference model (or building block) for IN in our study. Denote $\mathcal{T}$ as the set of transmitters and $\mathcal{R}$ as the set of intended receivers. Each receiver in $\mathcal{R}$ is in the transmission range of the transmitters in $\mathcal{T}$. Denote $\mathcal{P}$ as the set of receivers that are interfered with by all the transmitters in $\mathcal{T}$. That is, each node $j \in \mathcal{P}$ is within the interference range of every node $i \in \mathcal{T}$. A solid arrow line represents "aggregate" transmission link from the transmitters in $\mathcal{T}$ to the receivers in $\mathcal{R}$. Such an aggregate link refers to concurrent transmissions from nodes in $\mathcal{T}$ to those in $\mathcal{R}$. Likewise, a dashed arrow line represents "aggregate" interference link from transmitters in $\mathcal{T}$ to unintended receivers in $\mathcal{P}$. Again, such an aggregate link refers to concurrent interference from nodes in $\mathcal{T}$ to those in $\mathcal{P}$. When there is no ambiguity, we drop the wording "aggregate" when we refer to such links in IN.

For the transmitters in $\mathcal{T}$, they may not be able to neutralize their interference to all the receivers in $\mathcal{P}$. So we ask the following question: How many receivers in $\mathcal{P}$ can have their interference be neutralized by the transmitters in $\mathcal{T}$?

Define $\mathcal{Q}$ as a subset of $\mathcal{P}$ (i.e., $\mathcal{Q} \subseteq \mathcal{P}$). To have the transmitters in $\mathcal{T}$ neutralize their interference for each receiver in $\mathcal{Q}$ while keeping their signals decodable at each receiver in

$\mathcal{R}$, we need to ensure that the following linear constraints have a feasible solution:

$$|\sum_{i\in\mathcal{T}} h_{ji}u_i| > 0, \quad j \in \mathcal{R}; \tag{7.2}$$

$$\sum_{i\in\mathcal{T}} h_{ji}u_i = 0, \quad j \in \mathcal{Q}; \tag{7.3}$$

where h_{ji} is assumed to be a constant and u_i is a variable.

Since the channel coefficients are independent among themselves, it was shown by Yetis et al. [96] that a *nonzero* solution to (7.3) also satisfies (7.2) almost surely (with probability of 1). Therefore, we only need to make sure that there exists a *nonzero* solution to (7.3). For the linear equations in (7.3), there are $|\mathcal{Q}|$ constraints and $|\mathcal{T}|$ variables. According to [57, Ch. 2], a sufficient condition for the existence of nonzero solution to (7.3) is that it has more variables than constraints, i.e., $|\mathcal{Q}| \leq |\mathcal{T}| - 1$. Therefore, (7.2) and (7.3) always have a feasible solution if $|\mathcal{Q}| \leq |\mathcal{T}| - 1$, which means that the transmitters in $\mathcal{T}$ can neutralize their interference to $|\mathcal{T}| - 1$ receivers.

To find a set of precoding coefficients that satisfy (7.3), we can use Gauss–Jordan elimination algorithm. After performing row reduction, we can obtain the reduced row echelon form as follows:

$$\begin{aligned} u_1 + \tilde{h}_{12}u_2 + \tilde{h}_{13}u_3 + \cdots + \tilde{h}_{1(T-1)}u_{(T-1)} + \tilde{h}_{1T}u_T &= 0, \\ u_2 + \tilde{h}_{13}u_3 + \cdots + \tilde{h}_{2(T-1)}u_{(T-1)} + \tilde{h}_{2T}u_T &= 0, \\ \cdots & \\ u_{(T-1)} + \tilde{h}_{QT}u_T &= 0, \end{aligned} \tag{7.4}$$

where $T = |\mathcal{T}|$ and $Q = |\mathcal{Q}|$.

In the reduced row echelon form (7.4), if a variable is a leading variable for an equation, then it is a basic variable; otherwise, it is a free variable [57, Ch. 2]. Denote $\mathcal{G}$ as the set of basic variables and $\mathcal{E}$ as the set of free variables. Then, a solution to (7.3) can be computed

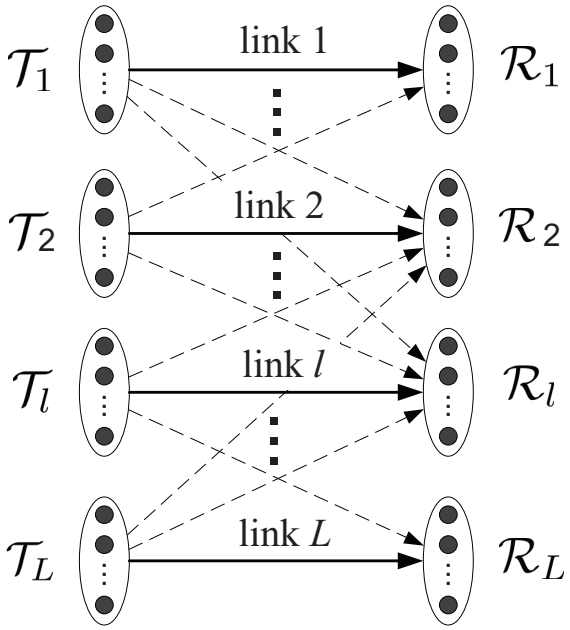


Figure 7.4: A set of aggregate transmission links.

as follows:

$$u_i = \begin{cases} 1, & \text{for } u_i \in \mathcal{E}, \\ -\sum_{k=i+1}^{T} \tilde{h}_{ik} u_k, & \text{for } u_i \in \mathcal{G}. \end{cases} \tag{7.5}$$

Given that $|\mathcal{Q}| \leq |\mathcal{T}| - 1$, there exists at least one free variable in (7.4), i.e., $\mathcal{E} \neq \emptyset$. Since the free variable is set to 1 in (7.5), the solution is a nonzero and thus satisfies (7.2) almost surely. The following lemma summarizes our results:

Lemma 11. *Through joint design of precoding coefficients, the transmitters in $\mathcal{T}$ can neutralize their interference to any subset of receivers $\mathcal{Q}$ in $\mathcal{P}$ with $|\mathcal{Q}| \leq |\mathcal{T}| - 1$.*

Feasibility Constraints for IN in a Single Hop. We now apply the results in Lemma 11 to a general one-hop transmission scenario as shown in Fig. 7.4. In this figure, there is a set $\mathcal{L}$ of concurrent links, where each link $l \in \mathcal{L}$ denotes aggreate transmissions from the transmitters in $\mathcal{T}_l$ to the receivers in $\mathcal{R}_l$, as we discussed previously. Suppose that each transmitter in $\mathcal{T}_l$ has the same interference range. Note that among the receivers in $\mathcal{R}_k$, not

all nodes may be interfered with by the transmitters in $\mathcal{T}_l$ (due to interference range). Denote $\mathcal{P}_l$ as the subset of receivers in $\cup_{k\in\mathcal{L}}^{k\neq l}\mathcal{R}_k$ that are interfered with by at least one transmitter in $\mathcal{T}_l$. Then the subset of receivers in $\mathcal{R}_k$ that are interfered with by the transmitters in $\mathcal{T}_l$ is $\mathcal{P}_l \cap \mathcal{R}_k$.

Denote α_l as a binary variable to indicate whether link $l \in \mathcal{L}$ is active, i.e., $\alpha_l = 1$ if link l is active (i.e., the transmitters in $\mathcal{T}_l$ are transmitting to the receivers in $\mathcal{R}_l$) and 0 otherwise. Obviously, not all links in $\mathcal{L}$ may be allowed to be active at the same time. So the question we ask is: for a given subset of links, can we determine if they can be active at the same time? In the following, we present feasibility constraints that can be used to make this determination. Further, this feasibility constraint can be used to plot the entire feasibility region for $(\alpha_1, \alpha_2, \cdots, \alpha_L)$.

To explore the feasibility constraints, we consider the following two cases for link $l \in \mathcal{L}$:

- *Link l is an active link (i.e., $\alpha_l = 1$).* In this case, the transmitters in $\mathcal{T}_l$ must neutralize their interference to all receivers in $\mathcal{P}_l$ that are receiving signals from their intended transmitters. To do so, the number of transmitters in $\mathcal{T}_l$ should be at least 1 more than the number of such active receivers in $\mathcal{P}_l$ (based on Lemma 11). We now count the number of such active receivers in $\mathcal{P}_l$. For link $k \in \mathcal{L}$, the subset of its intended receivers in $\mathcal{R}_k$ that is also within $\mathcal{P}_l$ is $\mathcal{P}_l \cap \mathcal{R}_k$. So the total number of active receivers in $\mathcal{P}_l$ is $\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k$. Based on Lemma 11, we have $\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k \leq |\mathcal{T}_l| - 1$.
- *Link l is not an active link (i.e., $\alpha_l = 0$).* In this case, the transmit nodes in $\mathcal{T}_l$ do not generate interference. So there is no restriction on the number of the active receivers with their interference ranges.

By defining N as the total number of nodes in the network, it is easy to verify that the above two cases can be combined as follows:

$$\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k \leq |\mathcal{T}_l| - 1 + (1-\alpha_l)N, \qquad l \in \mathcal{L}. \tag{7.6}$$

Specifically, when $\alpha_l = 1$, (7.6) is simplified to $\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k \leq |\mathcal{T}_l| - 1$, which is the requirement in the first case. When $\alpha_l = 0$, (7.6) is simplified to $\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k \leq N$,

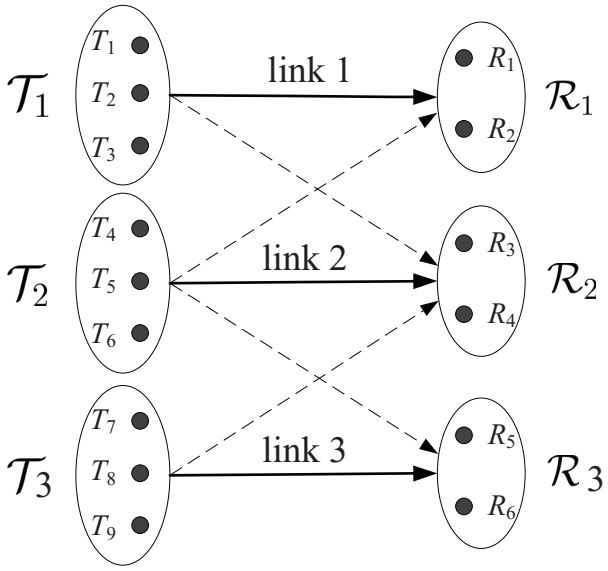


Figure 7.5: An example that explains IN constraints.

which is a dumb constraint as required the second case.

Another restriction of neutralization in this network is that the transmitters in $\mathcal{T}_l$ can only neutralize their interferences to a node that are interfered with by all transmitters in $\mathcal{T}_l$. To model this restriction, we define a binary constant for links l and k in Fig. 7.4 as follows:

$$W_{l,k} = \begin{cases} 0 & \text{There exists a node } j \in \mathcal{R}_k \text{ that is interfered} \\ & \text{with by a subset of transmitters in } \mathcal{T}_l \\ 1 & \text{otherwise.} \end{cases}$$

Based on the definition of $W_{l,k}$, we have the following two cases:

- $W_{l,k} = 0$: In this case, there are two facts: (i) node $j \in \mathcal{R}_k$ is interfered with by at least one transmitter in $\mathcal{T}_l$; (ii) node $j \in \mathcal{R}_k$ is not interfered with by all transmitters in $\mathcal{T}_l$. As neutralized is only done for a node that are interfered by all transmitters in $\mathcal{T}_l$, the interference on node j cannot be neutralized by $\mathcal{T}_l$. As a result, link k and l cannot be active at the same time, i.e., $\alpha_l + \alpha_k \leq 1$.
- $W_{l,k} = 1$: In this case, there is no additional restriction on α_l and α_k.

It is easy to verify that the above two cases can be equivalently combined to the following

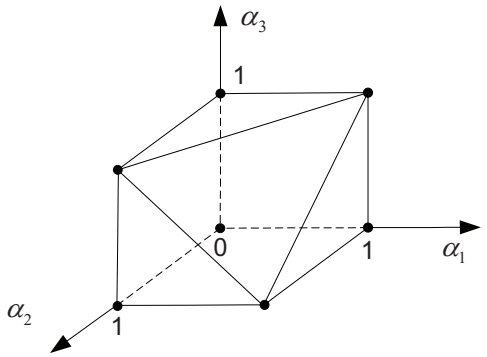


Figure 7.6: Feasible region of the network in Fig. 7.5.

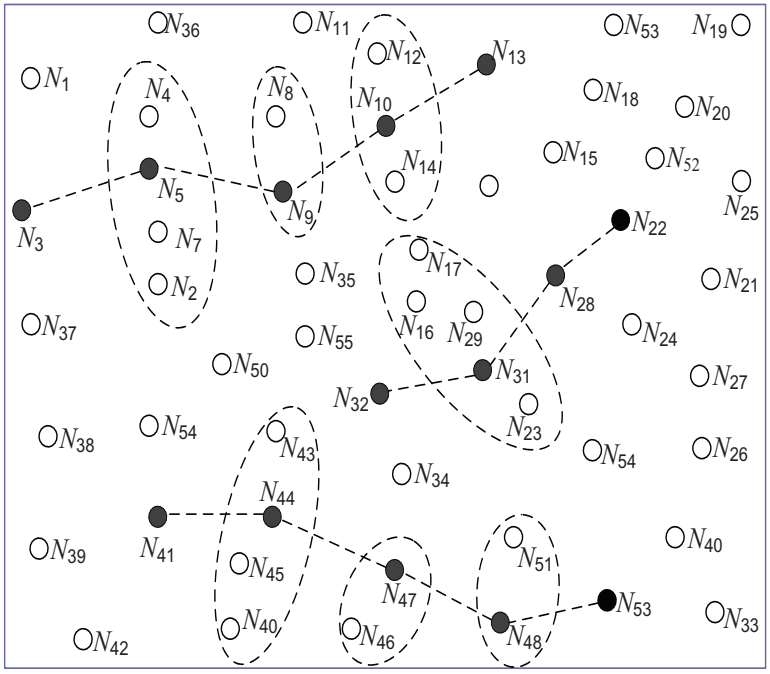
constraint:

$$\alpha_l + \alpha_k \leq 1 + W_{l,j} \ , \qquad l, k \in \mathcal{L}, l \neq k. \tag{7.7}$$

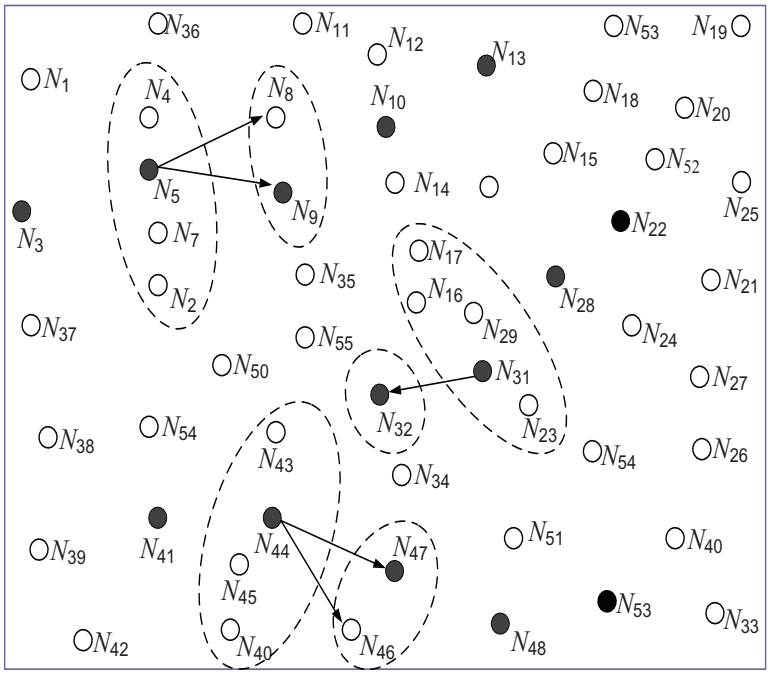
Therefore, (7.6) and (7.7) form the feasibility constraints for $(\alpha_1, \alpha_2, \cdots, \alpha_L)$. By enumerating all possible setting of $(\alpha_1, \alpha_2, \cdots, \alpha_L)$, we can use (7.6) to plot the feasible region.

An Example. Consider Fig. 7.5 where each link has three transmitters and two receivers, $\mathcal{P}_1 = \{R_3, R_4\}$, $\mathcal{P}_2 = \{R_1, R_2, R_5, R_6\}$, and $\mathcal{P}_3 = \{R_3, R_4\}$. We want to determine if $(\alpha_1, \alpha_2, \alpha_3) = (1, 1, 0)$ is feasible for this network, i.e., whether links 1 and 2 can be active at the same time. Since $W_{1,2} = W_{1,3} = W_{2,3} = 1$ in this network, it is easy to verify that $(1, 1, 0)$ satisfies (7.7). We now check whether or not the three links satisfies (7.6). For link 1 (i.e., $l = 1$), we have $\sum_{k\in\mathcal{L}}^{k\neq l} |\mathcal{P}_l \cap \mathcal{R}_k| \cdot \alpha_k = |\mathcal{P}_1 \cap \mathcal{R}_2| \cdot \alpha_2 + |\mathcal{P}_1 \cap \mathcal{R}_3| \cdot \alpha_3 = 2 + 0 = 2$ and $|\mathcal{T}_l| - 1 + (1 - \alpha_l)N = |\mathcal{T}_1| - 1 + (1 - \alpha_1)15 = 2$. Hence, link 1 satisfies constraint (7.6). Similarly, we can check for links 2 and 3 also satisfies (7.6). Therefore, $(\alpha_1, \alpha_2, \alpha_3) = (1, 1, 0)$ is feasible for this network.

Following the same token, we can check $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(1, 0, 1)$, $(0, 1, 1)$, $(1, 1, 1)$ and find that except $(1, 1, 1)$, all the others are feasible. The feasible region for this network is plotted in Fig. 7.6, where each point represents a feasible solution for $(\alpha_1, \alpha_2, \alpha_3)$.



(a) A network with three sessions. Each oval contains a node along a session's path and a set of nodes acting as its neuts.



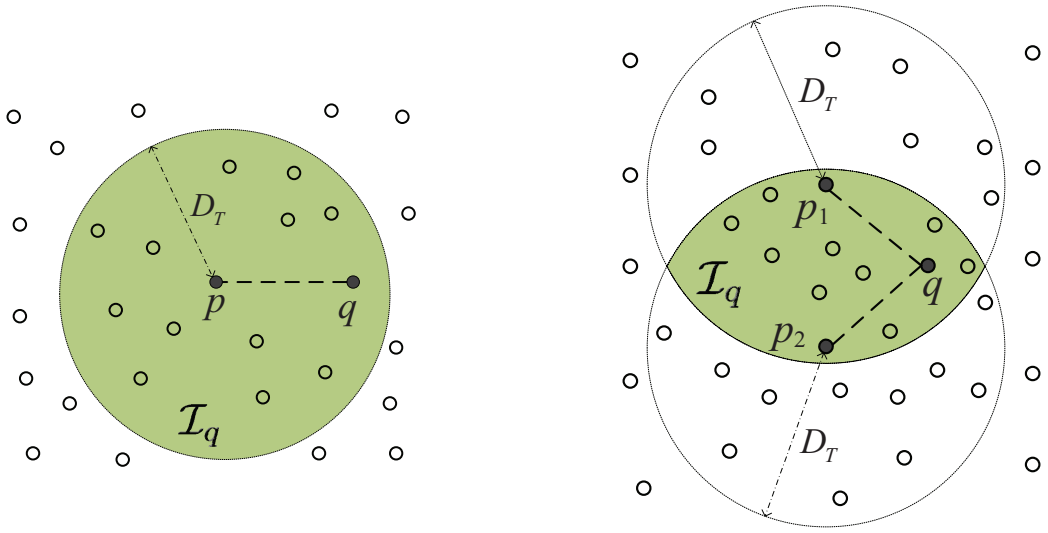
(b) IN among three active links in a time slot.

Figure 7.7: An example of IN in a multi-hop wireless network.

7.4 IN in Multi-hop Networks

In this section, we study IN in multi-hop networks. Consider a multi-hop network consisting of a set of nodes as shown in Fig. 7.7(a). Among the nodes, there is a set of sessions $\mathcal{S}$, with $\text{src}(s)$ and $\text{dst}(s)$ denoting the source and destination nodes of session $s \in \mathcal{S}$, respectively. Denote $r(s)$ as the end-to-end data rate of session $s \in \mathcal{S}$. We assume that the routing path of each session is obtained through some routing protocol for ad hoc networks. Based on the routing paths, the nodes in the network can be classified into two subsets: $\mathcal{N}_{\text{s}}$ and $\mathcal{N}_{\text{idle}}$, where $\mathcal{N}_{\text{s}}$ is the set of nodes on routing paths (marked as filled circles in Fig. 7.7(a)), and $\mathcal{N}_{\text{idle}}$ is the set of remaining nodes (marked as empty circles). Denote $\mathcal{L}_{\text{s}}$ as the set of links that are traversed by the set of sessions $\mathcal{S}$. Assume that transmission scheduling among the links in $\mathcal{L}_{\text{s}}$ is based on a frame that consists of T time slots.

For each multi-hop session, the idle nodes along the path can be used as *neuts* for IN, as we discussed in Fig. 7.2. Fig. 7.7(a) illustrates how some idle nodes along the path are chosen as neuts by the nodes along each path. For example, $\{N_2, N_4, N_7\}$ are chosen as N_5's neuts, $\{N_8\}$ is chosen as N_9's neut, $\{N_{12}, N_{14}\}$ are chosen as N_{10}'s neuts, and so forth. When receiving, the group of neuts can receive the same information as the corresponding node on the path from its one-hop upstream node. For example, $\{N_2, N_4, N_7\}$, which are N_5's neuts, can receive from N_3 since they are all within the transmission range of N_3. When transmitting, based on Lemma 11, $\{N_2, N_4, N_5, N_7\}$ can neutralize their interferences to three other unintended receivers via the joint design of their precoding coefficients. Fig. 7.7(b) shows an example where IN can enable simultaneous transmission in a multi-hop wireless network. Suppose all nodes are in the same interference domain. $\{N_2, N_4, N_5, N_7\}$ can neutralize their interferences to receive nodes N_{32}, N_{46}, and N_{47}. Similarly, the set of transmitters $\{N_{16}, N_{17}, N_{23}, N_{29}, N_{31}\}$ can neutralize their interferences to receive nodes N_8, N_9, N_{46}, and N_{47}; the set of transmitters $\{N_{40}, N_{43}, N_{44}, N_{45}\}$ can neutralize their interferences to receive nodes N_8, N_9, and N_{32}. Since all the interferences are neutralized at the receivers, the three aggregate links can be active in the same time slot. Note that without IN, only one of the three links ($N_5 \to N_9$, $N_{44} \to N_{47}$, and $N_{31} \to N_{32}$) can be active in a time slot.



(a) The case when q is traversed by only one session.

(b) The case when q is traversed by multiple sessions.

Figure 7.8: An illustration of $\mathcal{A}_q$ — the set containing eligible neuts for a node q.

It is important to note that for a node $q \in \mathcal{N}_s$, how to choose a subset of the neighboring nodes as neuts is the key problem. A small number of neuts will limit the IN capability (see Lemma 11) while a large number of neuts may unnecessarily increase the interference footprint. For the same number of neuts, the locations of the neuts are important as they will charaterize the overall shape and size and interference footprint. Therefore, one must choose the set of neuts for each $q \in \mathcal{N}_s$ meticulously to maximize the benefits of IN. In our study for IN in a multi-hop network, we model neut selection as part of the optimization problem.

7.4.1 Neut Selection and IN

Referring to Fig. 7.7(a), for a node on a path, say N_5, not every idle node is eligible to be its neut. An idle node is eligible to be N_5's neut only if it can receive the same information from N_5's one-hop upstream node, i.e., N_3. In general, for a node $q \in \mathcal{N}_s$, denote $\mathcal{I}_q$ as the set of idle nodes that are eligible to be its neuts. Referring to Fig. 7.8(a), if node q has only

one one-hop upstream node, say p, then $\mathcal{I}_q$ is the set of idle nodes within the transmission range of node p, i.e., $\mathcal{I}_q = \{i : d(i,p) \leq D_{\mathrm{T}}, i \in \mathcal{N}_{\mathrm{idle}}\}$, where $d(i,p)$ is the distance between nodes i and p and D_{T} is a node's transmission range. In the case when node $q \in \mathcal{N}_{\mathrm{s}}$ is on the route of multiple sessions (see, e.g., Fig. 7.8(b)), then node q has multiple one-hop upstream nodes and $\mathcal{I}_q$ is the set of idle nodes that fall in the intersection of the transmission ranges of these nodes. That is,

$$\mathcal{I}_q = \bigcap_{(p,q)\in\mathcal{L}_{\mathrm{s}}} \Big\{ i : d(i,p) \leq D_{\mathrm{T}}, i \in \mathcal{N}_{\mathrm{idle}} \Big\}, \qquad q \in \mathcal{N}_{\mathrm{s}},$$

where (p,q) represents a one-hop link in $\mathcal{L}_{\mathrm{s}}$.

Consider an idle node $i \in \mathcal{N}_{\mathrm{idle}}$. It may be eligible to serve as a neut for multiple nodes in $\mathcal{N}_{\mathrm{s}}$. Denote $\mathcal{M}_i$ as the set of nodes in $\mathcal{N}_{\mathrm{s}}$ where each node in $\mathcal{M}_i$ may choose idle node i its neut. Since idle node i can only serve as neut for at most one node in $\mathcal{M}_i$, denote $\lambda_{q,i}$ as a binary variable to indicate whether or not node i is assigned to serve as a neut for node $q \in \mathcal{M}_i$. Specifically, $\lambda_{i\to q} = 1$; and 0 otherwise.

$$\lambda_{i\to q} = \begin{cases} 1 & \text{if node } i \text{ is assigned to be node } q\text{'s neut,} \\ 0 & \text{otherwise.} \end{cases}$$

Since an idle node can serve as a neut for at most one node, we have the following constraint:

$$\sum_{q\in\mathcal{M}_i} \lambda_{i\to q} \leq 1, \qquad i \in \mathcal{N}_{\mathrm{idle}}. \tag{7.8}$$

For a node $q \in \mathcal{N}_{\mathrm{s}}$, we define $\mathcal{A}_q$ as the set of idle nodes that are eligible to be its neuts (i.e., $\mathcal{I}_q$), plus node q itself. That is,

$$\mathcal{A}_q = \mathcal{I}_q \cup \{q\}. \tag{7.9}$$

We also define $\mathcal{B}_q$ as the set of nodes in $\mathcal{A}_q$ that are actually chosen as node q's neuts, plus node q itself. For explanation convenience, we define $\lambda_{q\to q} = 1$ for $q \in \mathcal{N}_{\mathrm{s}}$. Then we have

$$\mathcal{B}_q = \big\{ i : i \in \mathcal{A}_q, \lambda_{i\to q} = 1 \big\}. \tag{7.10}$$

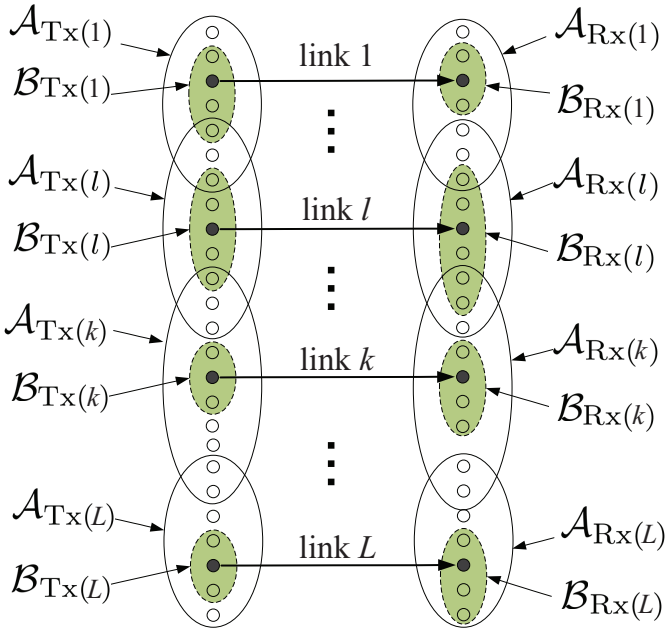


Figure 7.9: IN constraints and link scheduling constraints.

For transmission scheduling among the links in $\mathcal{L}_{\mathrm{s}}$, we assume it is based on time slots. Consider all links in $\mathcal{L}_{\mathrm{s}}$ in a time slot t (see Fig. 7.9). Obviously not all these links can be active at the same time (due to half duplex and limitation of IN in Lemma 11). Denote $\alpha_l(t)$ as a binary variable to indicate whether or not link $l \in \mathcal{L}_{\mathrm{s}}$ is active in time slot t, i.e.,

$$\alpha_l(t) = \begin{cases} 1 & \text{if link } l \text{ is active in time slot } t, \\ 0 & \text{otherwise.} \end{cases}$$

Consider link l's transmitters in $\mathcal{B}_{\mathrm{Tx}(l)}$ and a node j among another link k's receivers in $\mathcal{A}_{\mathrm{Rx}(l)}$. We define a binary variable $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ to indicate whether or not the transmitters in $\mathcal{B}_{\mathrm{Tx}(l)}$ neutralize their interferences to node $j \in \mathcal{A}_{\mathrm{Rx}(k)}$ in time slot t. That is,

$$\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = \begin{cases} 1 & \text{if the transmitters in } \mathcal{B}_{\mathrm{Tx}(l)} \text{ neutralize their interference} \\ & \text{to node } j \in \mathcal{A}_{\mathrm{Rx}(k)} \text{ in time slot } t, \\ 0 & \text{otherwise,} \end{cases}$$

Then, we have two cases:

- $\alpha_l(t) = 1$: In this case, the nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$ are active transmitters. Based on Lemma 11, we know that the transmitters in $\mathcal{B}_{\mathrm{Tx}(l)}$ can neutralize their interferences to

at most $(|\mathcal{B}_{\mathrm{Tx}(l)}|-1)$ unintended receivers. Based on (7.10), we have $|\mathcal{B}_{\mathrm{Tx}(l)}|-1 = (\sum_{i\in\mathcal{I}_{\mathrm{Tx}(l)}}\lambda_{i\to\mathrm{Tx}(l)}+1)-1 = \sum_{i\in\mathcal{I}_{\mathrm{Tx}(l)}}\lambda_{i\to\mathrm{Tx}(l)}$. Therefore, we have the following constraint:

$$\sum_{j\in\cup_{k\in\mathcal{L}_{\mathrm{s}}}^{k\neq l}\mathcal{A}_{\mathrm{Rx}(k)}}\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le |\mathcal{B}_{\mathrm{Tx}(l)}|-1 = \sum_{i\in\mathcal{I}_{\mathrm{Tx}(l)}}\lambda_{i\to\mathrm{Tx}(l)} \ .$$

- $\alpha_l(t)=0$: In this case, the nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$ are inactive. They neither produce any interference to other nodes nor possess IN capability. Therefore, we have $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)=0$ for all $j\in\mathcal{A}_{\mathrm{Rx}(k)}$, $k\in\mathcal{L}_{\mathrm{s}}\backslash\{l\}$.

It is easy to verify that the above two cases can be combined into one equivalent set of constraints:

$$\sum_{j\in\cup_{k\in\mathcal{L}_{\mathrm{s}}}^{k\neq l}\mathcal{A}_{\mathrm{Rx}(k)}}\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le \Big(\sum_{i\in\mathcal{I}_{\mathrm{Tx}(l)}}\lambda_{i\to\mathrm{Tx}(l)}\Big)\cdot\alpha_l(t) \ , \qquad l\in\mathcal{L}_{\mathrm{s}}, 1\le t\le T. \tag{7.11}$$

In our development for (7.11), we implicitly assume that node $j\in\mathcal{A}_{\mathrm{Rx}(k)}$ falls in the intersection of interference ranges of all nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$ when $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)=1$. We now develop necessary constraints to ensure that this assumption always holds. Define a binary indicator (constant) $E_{i,j}$ to indicate whether or not node j is within node i's interference range. That is,

$$E_{i,j} = \begin{cases} 1 & \text{if node } j \text{ is within node } i\text{'s interference range,} \\ 0 & \text{otherwise.} \end{cases}$$

If $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)=1$, then node j must be within the interference range of every node in $\mathcal{B}_{\mathrm{Tx}(l)}$, i.e., $E_{i,j}=1$ for $i\in\mathcal{B}_{\mathrm{Tx}(l)}$. Recall that $\lambda_{i\to\mathrm{Tx}(l)}=1$ for $i\in\mathcal{B}_{\mathrm{Tx}(l)}$ while $\lambda_{i\to\mathrm{Tx}(l)}=0$ for $i\in\mathcal{A}_{\mathrm{Tx}(l)}\backslash\mathcal{B}_{\mathrm{Tx}(l)}$. We have $\lambda_{i\to\mathrm{Tx}(l)}\le E_{i,j}$ for $i\in\mathcal{A}_{\mathrm{Tx}(l)}$. Otherwise (i.e., $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)=0$), $\mathcal{B}_{\mathrm{Tx}(l)}$ do not neutralize their interferences to node j. Then there is no requirement on whether node j is within the interference range of nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$. Combining these two cases, we have

$$\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)\cdot\lambda_{i\to\mathrm{Tx}(l)} \le \pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)\cdot E_{i,j} \ , \qquad l\in\mathcal{L}_{\mathrm{s}}, i\in\mathcal{A}_{\mathrm{Tx}(l)}, k\in\mathcal{L}_{\mathrm{s}}\backslash\{l\}, j\in\mathcal{A}_{\mathrm{Rx}(k)}, 1\le t\le T. \tag{7.12}$$

7.4.2 Link Scheduling Constraints

For each node, we assume that it has a single transmit/receive antenna and operates with a half-duplex radio. When acting as a transmitter (or receiver), we assume a node can only be used by at most one active link in a time slot. Then we have

$$\sum_{q\in\left\{\mathrm{Tx}(l),\mathrm{Rx}(l)\right\}} \alpha_l(t) \le 1, \quad q \in \mathcal{N}_\mathrm{s}, 1 \le t \le T. \tag{7.13}$$

Consider node $j \in \mathcal{A}_{\mathrm{Rx}(k)}$ in Fig. 7.9. We define a binary variable $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ to indicate whether or not node j is within the interference range of at least one active transmitter in $\mathcal{B}_{\mathrm{Tx}(l)}$. That is,

$$\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = \begin{cases} 1 & \text{if node } j \text{ is within the interference range of at least one} \\ & \text{active transmitter in } \mathcal{B}_{\mathrm{Tx}(l)} \text{ in time slot } t, \\ 0 & \text{otherwise.} \end{cases}$$

We now explore the relationship between $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ and $\alpha_l(t) = 1$ as follows. If $\alpha_l(t) = 1$, then the nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$ are active transmitters. Based on the definition of $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$, $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 0$ if and only if node j is out of the interference range of every node $i \in \mathcal{B}_{\mathrm{Tx}(l)}$, i.e., $E_{i,j} = 0$ for $i \in \mathcal{B}_{\mathrm{Tx}(l)}$. Therefore, we have

$$\frac{1}{N}\sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j} \le \theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le \sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j} \ ,$$

where N is the total number of nodes in the network. Otherwise (i.e., $\alpha_l(t) = 0$), the nodes in $\mathcal{B}_{\mathrm{Tx}(l)}$ are inactive. So they do not interfere with node j, i.e., $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 0$. Combining these two cases, we have

$$\Big[\frac{1}{N}\sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j}\Big]\cdot\alpha_l(t) \le \theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le \Big[\sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j}\Big]\cdot\alpha_l(t) \ ,$$
$$k \in \mathcal{L}_\mathrm{s}, j \in \mathcal{A}_{\mathrm{Rx}(k)}, l \in \mathcal{L}_\mathrm{s}\backslash\{k\}, 1 \le t \le T. \tag{7.14}$$

Consider a receive node $j \in \mathcal{B}_{\mathrm{Rx}(k)}$, $k \in \mathcal{L}_\mathrm{s}$, in Fig. 7.9. Since it has a single antenna, it does not have capability to cancel interference while receiving. If node j is active and being

interfered with by at least one node in $\mathcal{B}_{\mathrm{Tx}(l)}$, then $\mathcal{B}_{\mathrm{Tx}(l)}$ must neutralize such interference to node j. To meet this requirement, we have the following constraints for $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ and $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ in two cases:

- $\alpha_k(t) = 1$: In this case, link k is active and thus node j is an active receiver. If $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 1$, node j is interfered with by $\mathcal{B}_{\mathrm{Tx}(l)}$. To neutralize such interference to node j, we must have $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 1$. Otherwise (i.e., $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 0$), node j is not interfered with by $\mathcal{B}_{\mathrm{Tx}(l)}$. Then $\mathcal{B}_{\mathrm{Tx}(l)}$ do not need to neutralize their interferences to node j, i.e., $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = 0$. Combining these two cases, we have $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) = \pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$.
- $\alpha_k(t) = 0$: In this case, link k is an inactive link and thus node j is an inactive receiver. Then node j can be interfered with by any unintended transmitters, i.e., $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ can be either 0 or 1. Further, since node j is inactive, there is no requirement for $\mathcal{B}_{\mathrm{Tx}(l)}$ to neutralize their interferences to node j, i.e., $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ can be either 0 or 1.

Combining these two bullets, we have the following constraint:

$$-[1-\alpha_k(t)] \leq \theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) - \pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \leq [1-\alpha_k(t)]\,,$$
$$k \in \mathcal{L}_{\mathrm{s}}, j \in \mathcal{B}_{\mathrm{Rx}(k)}, l \in \mathcal{L}_{\mathrm{s}} \backslash \{k\}, 1 \leq t \leq T. \tag{7.15}$$

Collectively, constraints (7.8), (7.11), (7.12), (7.13), (7.14), and (7.15) constitute an IN model, which allows us to determine which subsets of links can be active simultaneously without the need of getting involved into the complex signal design at the PHY layer.

7.5 An Application of IN: A Throughput Maximization Problem

In this section, we apply IN to study a throughput maximization problem in a multi-hop network. The objective is to maximize the minimum achievable data rate among all sessions. To formulate this throughput maximization problem, consider a link $l \in \mathcal{L}_{\mathrm{s}}$ in the network. It may be traversed by multiple sessions' routing paths. Denote $\mathcal{S}_l$ as the set of sessions that share wireless link l along their paths. Then the aggregate data rate requirement on link

l is $\sum_{s\in\mathcal{S}_l} r(s)$. For link l, we assume that fixed modulation and coding scheme (MCS) is used for its data stream transmission and one data stream in one time slot corresponds to one unit data rate. Then the achievable data rate of link l (averaged over T time slots) is $\frac{1}{T}\sum_{t=1}^{T}\alpha_l(t)$. Since the aggregate data rate requirement (for its traversing sessions) cannot exceed the achievable data rate, we have

$$\sum_{s\in\mathcal{S}_l} r(s) \le \frac{1}{T}\sum_{t=1}^{T}\alpha_l(t), \quad l\in\mathcal{L}. \tag{7.16}$$

By denoting $r_{\min}$ as the minimum achievable rate among all sessions, we have

$$r_{\min} \le r(s), \quad s\in\mathcal{S}. \tag{7.17}$$

Collectively, we can formulate this optimization problem as follows:

OPT-IN$^{\text{raw}}$: max $r_{\min}$
s.t. Neut selection and IN: (7.8), (7.11), (7.12);
Link scheduling: (7.13), (7.14), (7.15);
Link rate constraints: (7.16);
Min rate constraints: (7.17).

In this optimization formulation, (7.11), (7.12), (7.14), and (7.15) are nonlinear constraints and the other are all linear. We now show how to linearize (7.11), (7.12), (7.14), and (7.15).

Reformation of (7.11). To linearize (7.11), we define a new binary variable $\gamma_{i\to\text{Tx}(l)}(t) = \lambda_{i\to\text{Tx}(l)}\cdot\alpha_l(t)$. Then, (7.11) can be equivalently transformed to the following linear constraint:

$$\sum_{j\in\cup_{k\in\mathcal{L}_\text{s}}^{k\neq l}\mathcal{A}_{\text{Rx}(k)}} \pi_j^{\mathcal{B}_{\text{Tx}(l)}}(t) \le \sum_{i\in\mathcal{I}_{\text{Tx}(l)}} \gamma_{i\to\text{Tx}(l)}(t) l\in\mathcal{L}_\text{s}, 1\le t\le T. \tag{7.18}$$

Based on the definition of $\gamma_{i\to\text{Tx}(l)}(t)$, we can enumerate all possible cases for $\lambda_{i\to\text{Tx}(l)}$ and $\alpha_l(t)$ (both are binary variables) and obtain the following linear constraint for $\lambda_{i\to\text{Tx}(l)}$:

$$\lambda_{i\to\text{Tx}(l)} + \alpha_l(t) - 1 \le \gamma_{i\to\text{Tx}(l)}(t) = \frac{1}{2}[\lambda_{i\to\text{Tx}(l)} + \alpha_l(t)],$$
$$l\in\mathcal{L}, i\in\mathcal{A}_{\text{Tx}(l)}, 1\le t\le T. \tag{7.19}$$

Reformation of (7.12). In (7.12), $E_{i,j}$ is a constant. But we have a product of two binary variables $\lambda_{i\to\mathrm{Tx}(l)}$ and $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$. By enumerating all the possibilities for these two binary variables, it is easy to verify that (7.12) is equivalent to the following linear constraint:

$$\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) + \lambda_{i\to\mathrm{Tx}(l)} \le 1 + E_{i,j} \ , \\ l \in \mathcal{L}_{\mathrm{s}}, i \in \mathcal{A}_{\mathrm{Tx}(l)}, j \in \cup_{k\in\mathcal{L}_{\mathrm{s}}}^{k\neq l} \mathcal{A}_{\mathrm{Rx}(k)}, 1 \le t \le T. \tag{7.20}$$

Reformation of (7.14). In (7.14), $E_{i,j}$ is a constant but $\mathcal{B}_{\mathrm{Tx}(l)}$ is a variable set. To linearize this constraint, we classify the nodes in $\mathcal{A}_{\mathrm{Tx}(l)}$ into two subsets: $\mathcal{B}_{\mathrm{Tx}(l)}$ and $\mathcal{A}_{\mathrm{Tx}(l)}\backslash\mathcal{B}_{\mathrm{Tx}(l)}$. For node $i \in \mathcal{B}_{\mathrm{Tx}(l)}$, $\lambda_{i\to\mathrm{Tx}(l)} = 1$. For node $i \in \mathcal{A}_{\mathrm{Tx}(l)}\backslash\mathcal{B}_{\mathrm{Tx}(l)}$, $\lambda_{i\to\mathrm{Tx}(l)} = 0$. Then we have

$$\begin{aligned}\sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j} &= \sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)} \\ &= \sum_{i\in\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)} + \sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}\backslash\mathcal{B}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)} \\ &= \sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)} \ .\end{aligned}$$

Based on the above result, constraint (7.14) can be equivalently written as:

$$\Big[\frac{1}{N}\sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)}\Big]\cdot\alpha_l(t) \le \theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le \Big[\sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\lambda_{i\to\mathrm{Tx}(l)}\Big]\cdot\alpha_l(t), \\ k \in \mathcal{L}_{\mathrm{s}}, j \in \mathcal{A}_{\mathrm{Rx}(k)}, l \in \mathcal{L}_{\mathrm{s}}\backslash\{k\}, 1 \le t \le T. \tag{7.21}$$

Constraint (7.21) is still nonlinear as it involves product of variables $\lambda_{i\to\mathrm{Tx}(l)}\cdot\alpha_l(t)$. Based on our previous definition of $\gamma_{i\to\mathrm{Tx}(l)}(t) = \lambda_{i\to\mathrm{Tx}(l)}\cdot\alpha_l(t)$, nonlinear constraint (7.21) can be further linearized as

$$\frac{1}{N}\sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\gamma_{i\to\mathrm{Tx}(l)}(t) \le \theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t) \le \sum_{i\in\mathcal{A}_{\mathrm{Tx}(l)}} E_{i,j}\cdot\gamma_{i\to\mathrm{Tx}(l)}(t) \ , \\ k \in \mathcal{L}_{\mathrm{s}}, j \in \mathcal{A}_{\mathrm{Rx}(k)}, l \in \mathcal{L}_{\mathrm{s}}\backslash\{k\}, 1 \le t \le T. \tag{7.22}$$

Reformation of (7.15). For (7.15), it appears linear, but it is actually not. This is

because the set of nodes in $\mathcal{B}_{\mathrm{Rx}(k)}$ is unknown, which is determined by variable $\lambda_{j\to\mathrm{Rx}(k)}$.[1] To address this problem, we want to extend the range from $\mathcal{B}_{\mathrm{Rx}(k)}$ to $\mathcal{A}_{\mathrm{Rx}(k)}$ (a constant set) without any change to the solution space. Recall that $\lambda_{j\to\mathrm{Rx}(k)}=1$ for $j\in\mathcal{B}_{\mathrm{Rx}(k)}$ and $\lambda_{j\to\mathrm{Rx}(k)}=0$ for $j\in\mathcal{A}_{\mathrm{Rx}(k)}\backslash\mathcal{B}_{\mathrm{Rx}(k)}$. Therefore, for $j\in\mathcal{A}_{\mathrm{Rx}(k)}$, we have the following two cases:

- $\lambda_{j\to\mathrm{Rx}(k)}=1$: In this case, node j must be in $\mathcal{B}_{\mathrm{Rx}(k)}$. Then node j must satisfies (7.15).
- $\lambda_{j\to\mathrm{Rx}(k)}=0$: In this case, node j must be in $\mathcal{A}_{\mathrm{Rx}(k)}\backslash\mathcal{B}_{\mathrm{Rx}(k)}$. Then node j does not need to satisfy (7.15).

Combining these two cases, $j\in\mathcal{B}_{\mathrm{Rx}(k)}$ in (7.15) can be expanded to $j\in\mathcal{A}_{\mathrm{Rx}(k)}$ by rewriting (7.15) as follows:

$$-[2-\alpha_k(t)-\lambda_{j\to\mathrm{Rx}(k)}]\le\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)-\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)\le[2-\alpha_k(t)-\lambda_{j\to\mathrm{Rx}(k)}]\,,$$
$$k\in\mathcal{L}_{\mathrm{s}},j\in\mathcal{A}_{\mathrm{Rx}(k)},l\in\mathcal{L}_{\mathrm{s}}\backslash\{k\},1\le t\le T. \tag{7.23}$$

Note that when $\lambda_{j\to\mathrm{Rx}(k)}=1$, (7.23) is exactly (7.15); when $\lambda_{j\to\mathrm{Rx}(k)}=0$, (7.23) does not impose any additional constraint on $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$, $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$, and $\alpha_k(t)$.

In summary, by replacing nonlinear constraints (7.11), (7.12), (7.14), and (7.15) with (7.18), (7.19), (7.20), (7.22), and (7.23), we have reformulated OPT-IN$^{\mathrm{raw}}$ as follows:

OPT-IN:	max	$r_{\min}$
	s.t.	Neut selection and IN: (7.8), (7.18), (7.19), (7.20);
		Link scheduling: (7.13), (7.22), (7.23);
		Link rate constraints: (7.16);
		Min rate constraints: (7.17);

where N, T, and $E_{i,j}$ are constant; $\mathcal{N}_{\mathrm{s}}$, $\mathcal{N}_{\mathrm{idle}}$, $\mathcal{L}_{\mathrm{s}}$, $\mathcal{S}$, $\mathcal{A}_{\mathrm{Tx}(l)}$, and $\mathcal{A}_{\mathrm{Rx}(l)}$ are given sets; $\alpha_l(t)$, $\lambda_{i\to q}$, $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$, $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$, and $\gamma_{i\to\mathrm{Tx}(l)}(t)$ are binary variables; $r_{\min}$ and $r(s)$ are non-negative continuous variables. Note that the reformulation does not change the optimality of the problem, i.e., OPT-IN$^{\mathrm{raw}}$ and OPT-IN have the same optimal objective value.

[1] In (7.15) and other constraints, the superscript $\mathcal{B}_{\mathrm{Tx}(l)}$ in variables $\theta_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ and $\pi_j^{\mathcal{B}_{\mathrm{Tx}(l)}}(t)$ is only a notation, but not a variable set.

OPT-IN is in the form of mixed-integer linear program (MILP). Although a general MILP problem is NP-hard [74], there exist tractable optimal algorithms (e.g., branch-and-bound with cutting planes [35, Ch. 5]) and efficient heuristic algorithms (e.g., sequential fixing algorithm [35, Ch. 10]) to solve it. For networks with a few hundred nodes, an off-the-shelf (commercial) solver such as CPLEX [104] can be used. Since our goal is to study the performance gain of IN in multi-hop networks, we will employ CPLEX in our performance evaluation.

7.6 Performance Evaluation

The goal of this section is twofold. First, we want to use a case study to illustrate how a solution to OPT-IN actually works in a multi-hop network. Second, we present results from a large number of cases to show how IN affect throghput performance under different settings.

As a benchmark for performance comparison, we formulate a baseline throughput maximization problem when IN is not used and denote it as OPT-noIN. Recall that $\lambda_{i\to q}$ is the binary variable that indicates whether or not node i is assigned to be node q's neut. So it suffices to disable the IN capability by simply not selecting any neut for each node $q \in \mathcal{N}_{\rm s}$, i.e., $\lambda_{i\to q} = 0$ for $q \in \mathcal{N}_{\rm s}$ and $i \in \mathcal{N}_{\rm idle}$. Therefore, OPT-noIN can be written as

OPT-noIN: max	$r_{\min}$	
s.t.	Neut selection and IN: (7.8), (7.18), (7.19), (7.20);	
	Link scheduling: (7.13), (7.22), (7.23);	
	Link rate constraints: (7.16);	
	Min rate constraints: (7.17);	
	$\lambda_{i\to q} = 0,$	$i \in \mathcal{N}_{\rm idle},\, q \in \mathcal{N}_{\rm s}$;

7.6.1 Simulation Setting

Without loss of generality, we normalize the units of distance, data rate, bandwidth, and time with appropriate dimensions. We consider a multi-hop network consisting of a set of

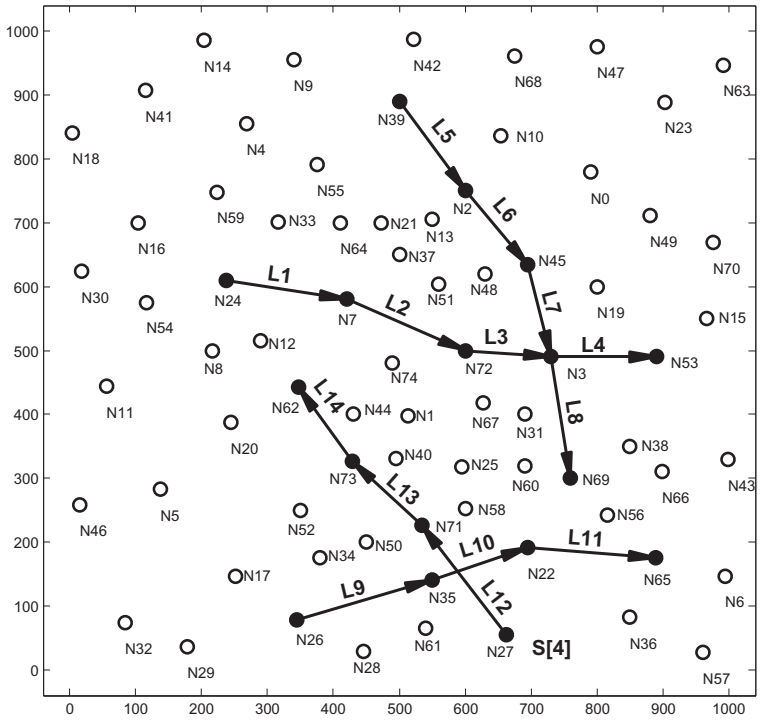
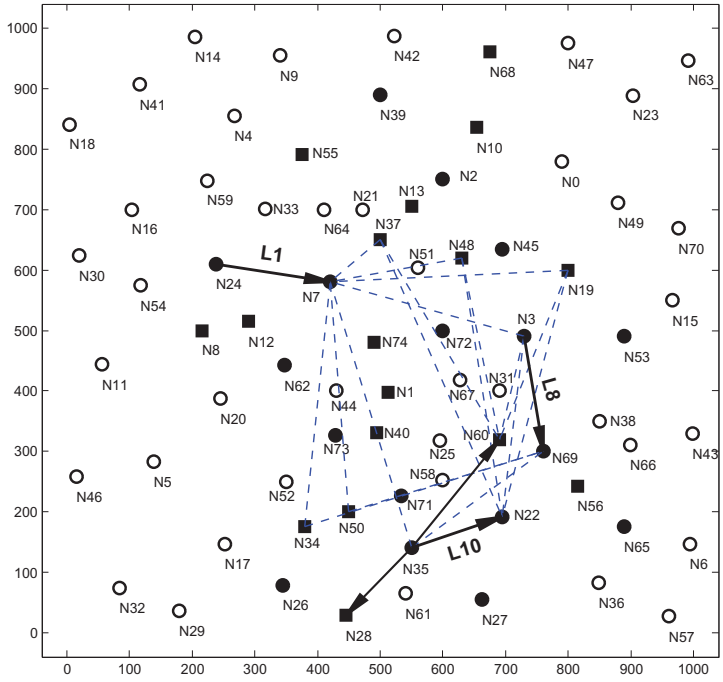


Figure 7.10: A network instance for case study.

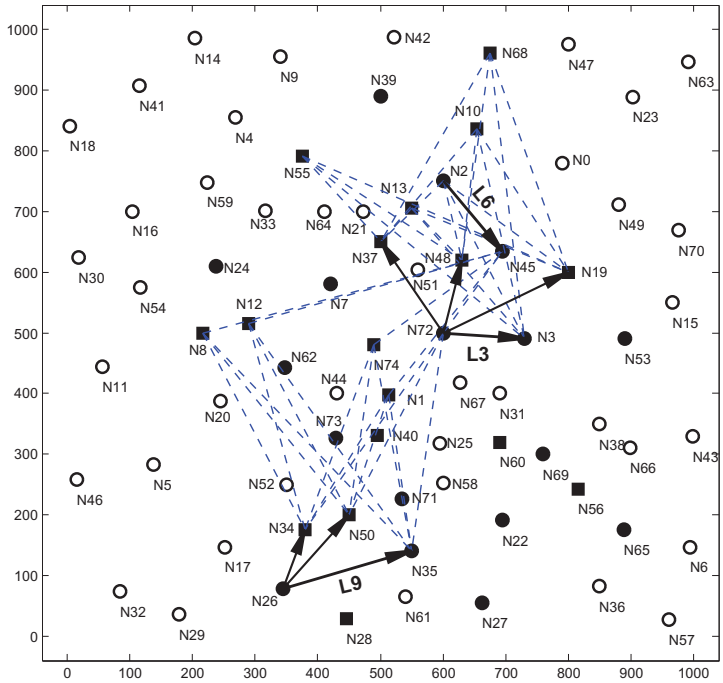
randomly generated nodes that are uniformly distributed in a 1000×1000 square region. Each node is equipped a single antenna. The transmission and interference ranges are set to 250 and 500, respectively. There are a set of sessions in the network, with their source and destination nodes being randomly chosen among all the nodes. The routing path of each session is computed based on a node's transmission range and the shortest path routing algorithm. We assume each source node has persistent traffic for transmission. We also assume that a frame has $T = 15$ time slots.

7.6.2 A Case Study

Fig. 7.10 shows a network instance that consists of 75 randomly distributed nodes. Among the nodes, there are 4 sessions, with their source and destination nodes as well as their routing paths showing in the figure. Specifically, the first session is $N_{24} \rightarrow N_7 \rightarrow N_{72} \rightarrow N_3 \rightarrow N_{53}$; the second session is $N_{39} \rightarrow N_2 \rightarrow N_{45} \rightarrow N_3 \rightarrow N_{69}$; the third session is



(a) Time slot 4



(b) Time slot 7

Figure 7.11: IN behavior in two time slots in the case study.

$N_{26} \rightarrow N_{35} \rightarrow N_{22} \rightarrow N_{65}$; and the fourth session is $N_{27} \rightarrow N_{71} \rightarrow N_{73} \rightarrow N_{62}$. By solving OPT-IN for this network instance, we obtained the optimal objective value of 2/15, meaning that each session's minimum end-to-end achievable rate is 2/15. In contrast, for OPT-noIN, the optimal objective value is 1/15. This means that the use of IN can increase the session throughput by 100% for this case study.

Table 7.2 presents the neut selection result in the solution from solving OPT-IN. In this table, $\mathcal{A}_q$ is the set of idle nodes that are eligible to be node q's neuts, plus node q itself; $\mathcal{B}_q$ is the set of idle nodes that are chosen to be node q's neuts, plus node q itself. For each source node of a session, it does not have a one-hop upstream node. Based on the definitions of $\mathcal{A}_q$ and $\mathcal{B}_q$, both $\mathcal{A}_q$ and $\mathcal{B}_q$ only contains q. For other non-source nodes q's on a session's routing paths, each of them typically have multiple nodes in $\mathcal{A}_q$ and a subset of nodes in $\mathcal{A}_q$ are chosen by $\mathcal{B}_q$. As an example, for node N_2, since its upstream node is N_{39}, the idle nodes within N_{39}'s transmission range include N_4, N_9, N_{10}, N_{13}, N_{21}, N_{37}, N_{42}, N_{55}, N_{64}, N_{68}. Including N_2 itself, we have set $\mathcal{A}_{N_2}$ shown in Table 7.2. Among these nodes in $\mathcal{A}_{N_2}$, nodes N_{10}, N_{13}, N_{55}, and N_{68} are chosen to be N_2's neuts, which form the set $\mathcal{B}_{N_2}$. For a node q that is on multiple session's path, then node q has multiple upstream nodes and $\mathcal{A}_q$ is the set of nodes that fall in the intersection of transmission ranges of these nodes. For example, for N_3, it is traversed by two sessions and has two one-hop upstream nodes: N_{45} and N_{72}. Only those idle nodes within both N_{45}'s and N_{72}'s transmission ranges are eligible to be N_3's neuts. The set of idle nodes within N_{45}'s transmission range is $\{N_0, N_{10}, N_{13}, N_{19}, N_{21}, N_{31}, N_{37}, N_{48}, N_{49}, N_{51}, N_{67}\}$. The set of idle nodes within N_{72}'s transmission range is $\{N_1, N_{13}, N_{19}, N_{21}, N_{25}, N_{31}, N_{37}, N_{40}, N_{44}, N_{48}, N_{51}, N_{58}, N_{60}, N_{67}, N_{74}\}$. The union of these two sets, plus N_3 itself, is $\mathcal{A}_{N_3}$, as shown in Table 7.2.

Since a time frame has $T = 15$ time slots, we have a scheduling solutuion for each time slot. Fig. 7.11 illustrates scheduling results for time slots 4 and 7. In these two figures, a solid arrow line represents an intended transmission and a dashed line represents interference. There are three types of legends for nodes in the figures: a filled circle (•) represents a node on a session's path; a filled square (■) represents an idle node that is chosen as a neut in

Table 7.2: $\mathcal{A}_q$ and $\mathcal{B}_q$ for each node q along a session's path.

$q \in \mathcal{N}_s$	$\mathcal{A}_q$	$\mathcal{B}_q$
N_2	N_4, N_9, N_{10}, N_{13}, N_{21}, N_{37}, N_{42}, N_{55}, N_{64}, N_{68}, N_2	N_{10}, N_{13}, N_{55}, N_{68}, N_2
N_3	N_{13}, N_{19}, N_{21}, N_{31}, N_{37}, N_{48}, N_{51}, N_{67}, N_3	N_{19}, N_{37}, N_{48}, N_3
N_7	N_4, N_8, N_{11}, N_{12}, N_{16}, N_{20}, N_{30}, N_{33}, N_{54}, N_{55}, N_{59}, N_{64}, N_7	N_7
N_{22}	N_{25}, N_{28}, N_{34}, N_{40}, N_{50}, N_{52}, N_{58}, N_{60}, N_{61}, N_{22}	N_{28}, N_{60}, N_{22}
N_{24}	N_{24}	N_{24}
N_{26}	N_{26}	N_{26}
N_{27}	N_{27}	N_{27}
N_{35}	N_{17}, N_{28}, N_{29}, N_{34}, N_{50}, N_{52}, N_{61}, N_{35}	N_{34}, N_{50}, N_{35}
N_{39}	N_{39}	N_{39},
N_{53}	N_1, N_{15}, N_{19}, N_{25}, N_{31}, N_{38}, N_{48}, N_{51}, N_{60}, N_{66}, N_{67}, N_{74}, N_{53}	N_{53}
N_{45}	N_0, N_{10}, N_{13}, N_{21}, N_{37}, N_{42}, N_{48}, N_{51}, N_{55}, N_{64}, N_{68}, N_{45},	N_{45}
N_{62}	N_1, N_{12}, N_{20}, N_{25}, N_{34}, N_{40}, N_{44},N_{50}, N_{52}, N_{58}, N_{67}, N_{74}, N_{62}	N_{62}
N_{65}	N_{25}, N_{31}, N_{36}, N_{38}, N_{40}, N_{50}, N_{56}, N_{58}, N_{60}, N_{61}, N_{66}, N_{67}, N_{65}	N_{65}
N_{69}	N_1, N_{15}, N_{19}, N_{25}, N_{31}, N_{38}, N_{48}, N_{51}, N_{60}, N_{66}, N_{67}, N_{74}, N_{69}	N_{69}
N_{71}	N_{28}, N_{36}, N_{56}, N_{58}, N_{61}, N_{71},	N_{56}, N_{71}
N_{72}	N_1, N_8, N_{12}, N_{13}, N_{21}, N_{33}, N_{37}, N_{44}, N_{48}, N_{51}, N_{55}, N_{64}, N_{74}, N_{72}	N_1, N_8, N_{12}, N_{74}, N_{72}
N_{73}	N_1, N_{25}, N_{28}, N_{31}, N_{34}, N_{40}, N_{44}, N_{50}, N_{52}, N_{58}, N_{60}, N_{61}, N_{67}, N_{73}	N_{40}, N_{73}

the solution; and an empty circle ($\circ$) represents an idle node that is not chosen as a neut in the solution. Fig. 7.11(a) exhibits transmissions and IN in time slot 4. In this time slot, there are 3 active transmissions: L1 ($N_{24} \rightarrow N_7$), L8 ($N_3 \rightarrow N_{69}$), and L10 ($N_{35} \rightarrow N_{22}$). For transmissions L1 and L8, their respective intended receivers N_7 and $N-69$ do not have any neut. But for transmission L10, its intended receiver $N_2 2$ has 2 neuts: N_{28} and N_{60}. So the transmission L10 is received by N_{22}, N_{28}, and N_{60}. We now consider the interferences among the three simultaneous transmissions. For transmitter N_{24}, its interference range does not cover L8's receiver (i.e., N_{69}). Nor does it covers L10's receiver and its neuts (i.e., N_{28}, N_{60}, and N_{69}). So N_{24} does not interfere those receivers in this time slot. For transmitter N_3, its interference range covers L1's receiver N_7, L10's receiver N_{22} and its neut N_{60}. Since N_3 has 3 neuts (N_{19}, N_{37}, and N_{48}), the interferences from $\{N_3, N_{19}, N_{37}, N_{48}\}$ to receivers $\{N_7, N_{22}, N_{60}\}$ can be completely neutralized. For transmitter N_{35}, its interference range covers L1's receiver N_7 and L8's receiver N_{69}. Since N_{35} has two neuts (N_{34} and N_{50}), the interferences from $\{N_{35}, N_{34}, N_{50}\}$ to receivers $\{N_7, N_{69}\}$ can be completely neutralized. Since all interferences from the transmitters can be neutralized, each receiver can receive the data from its intended transmitter free of interference. Therefore, these three transmissions ($N_{24} \rightarrow N_7$, $N_3 \rightarrow N_{69}$, and $N_{35} \rightarrow N_{22}$) can be active in the same time slot, albeit their receivers are within other transmitter's interference range. Note that without IN, these three transmissions cannot be active in the same time slot.

Fig. 7.11(b) shows transmissions and IN behavior in time slot 7 in our solution. In this time slot, there are also three active transmissions: L3 ($N_{72} \rightarrow N_3$), L6 ($N_2 \rightarrow N_{45}$), and L9 ($N_{26} \rightarrow N_{35}$). For node N_3, it has three neuts: N_{19}, N_{37}, and N_{48}. So the transmission from node N_{72} is received by nodes N_3, N_{19}, N_{37}, and N_{48} simultaneously. For node N_{45}, it does not have neut. So data from transmitter N_2 is only received by N_{45}. For node N_{35}, it has two neuts: N_{34} and N_{50}. So the data from the transmitter N_{26} is received by receivers N_{35}, N_{34}, and N_{50}. To see how the three transmissions are possible within the same time slot, we examine interferences at each receiver. For L3's transmitter N_{72}, its interference range covers L6's and L9's receivers as well as their neuts (i.e., N_{45}, N_{35}, N_{34}, and N_{50}).

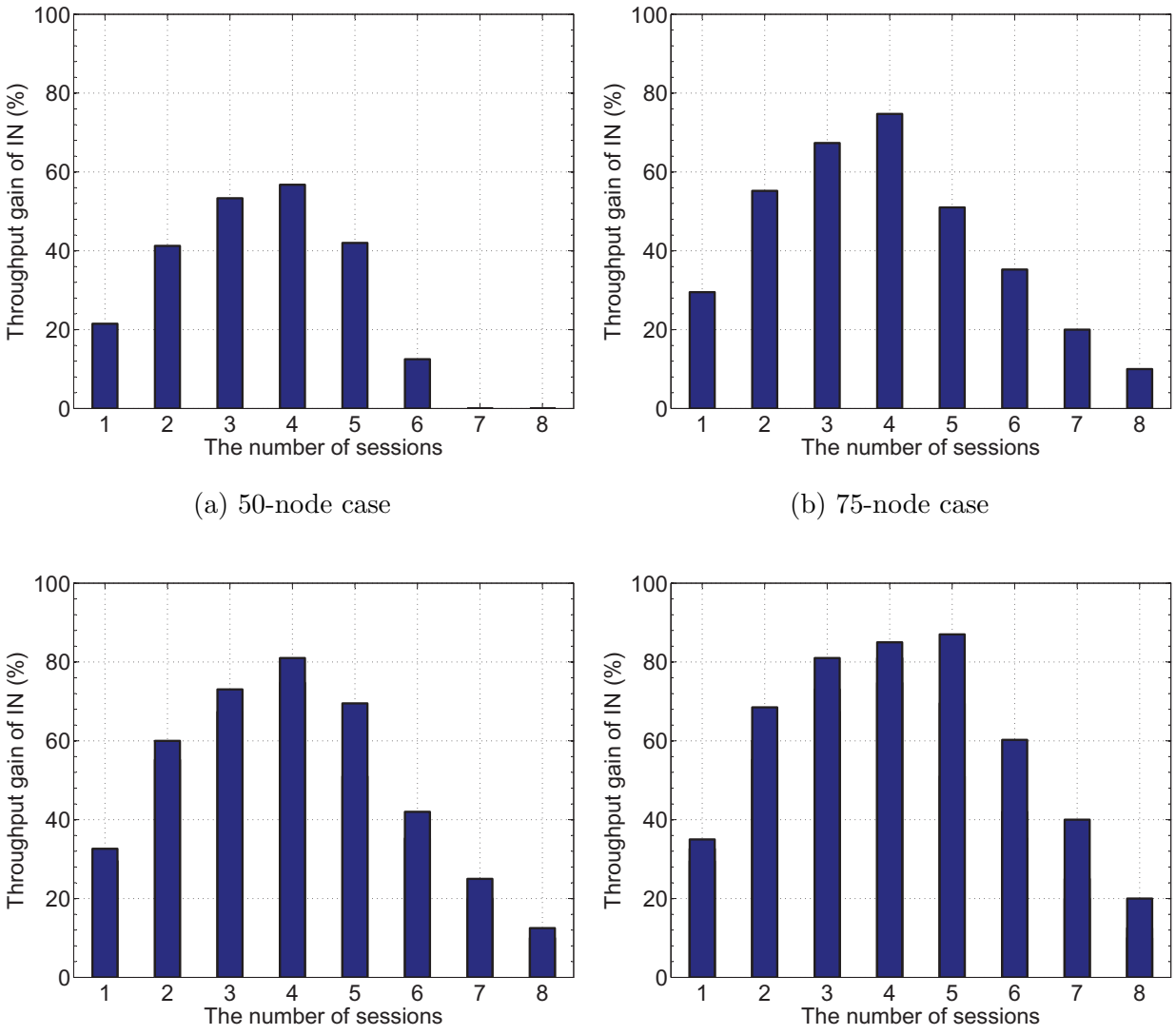
According to Table 7.2, N_{72} has 4 neuts: N_1, N_8, N_{12}, and N_{74}. So the interferences from $\{N_{72}, N_1, N_8, N_{12}, N_{74}\}$ to receivers $\{N_{45}, N_{35}, N_{34}, N_{50}\}$ can be neutralized. Similarly, for L6's transmitter N_2, it has neuts $\{N_{10}, N_{13}, N_{15}, N_{68}\}$. N_2, $\{N_{10}, N_{13}, N_{15}, N_{68}\}$ can neutralize their interferences to $\{N_3, N_{19}, N_{37}, N_{48}\}$. For L9's transmitter N_{26}, its interference range does not cover any active receivers, So there is no interference that needs to be neutralized. As a result, all interferences among the 3 transmissions are neutralized and the 3 transmissions can be active in the same time slot.

7.6.3 Complete Results

We now present complete simulation results to show the throughput gain of IN in multi-hop networks. For each randomly generated network instance, the throughput gain of IN is calculated by $(\hat{r}^*_{\min} - \tilde{r}^*_{\min})/\tilde{r}^*_{\min}$, where $\hat{r}^*_{\min}$ and $\tilde{r}^*_{\min}$ are the optimal objective values from solving OPT-IN and OPT-noIN for this network instance, respectively.

Impact of Traffic Volume. We first study the impact of traffic volume on the throughput gain of IN in multi-hop networks. Fig. 7.12 presents our simulation results for four network sizes: (a) 50 nodes; (b) 75 nodes; (c) 100 nodes; and (d) 125 nodes. For each figure, the x-axis is the number of sessions in the network and the y-axis is the average throughput gain over 100 randomly generated network instances.

Let's first look at Fig. 7.12(a). When the number of sessions increased (until 4), the throughput gain of IN also increases. This is because in this regime, the potential of IN has not yet be fully exploited as the interferences among the links in each time slot is limited. Increasing the number of sessions will bring in more interferences, thereby more potential of IN can be exploited. When the number of sessions reaches 4, the throughput gain reaches its maximum, which is about 55%. When the number of sessions is more than 4, the throughput gain decreases. This is because the number of idle nodes in the network are being exhuasted and fewer and fewer can be used for neuts. When there are 7 or more sessions in the network, there is no throughput gain by IN, indicating that there are no enough idle nodes available for IN.



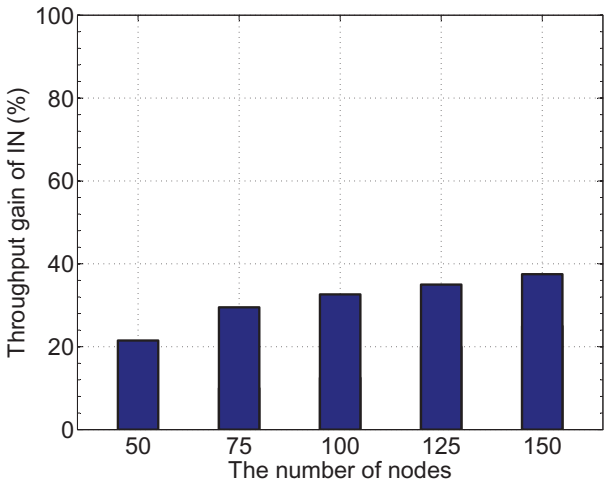
(a) 50-node case

(b) 75-node case

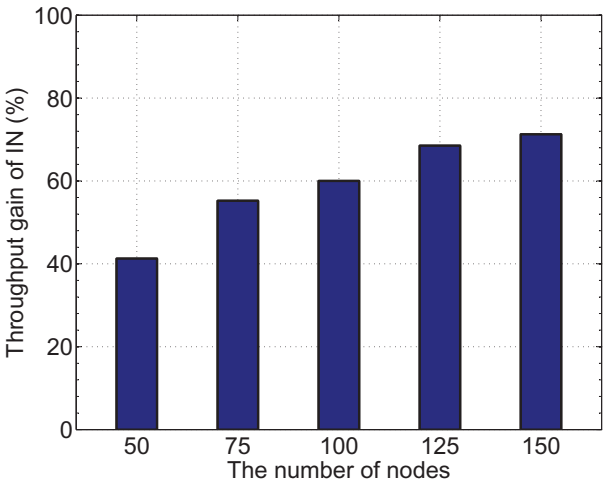
(c) 100-node case

(d) 125-node case

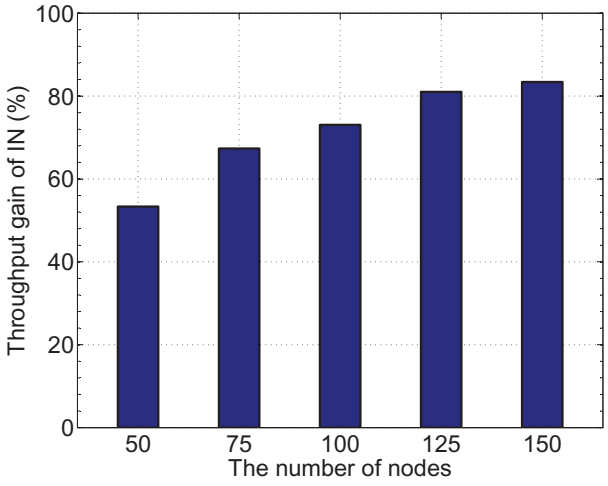
Figure 7.12: Impact of the number of sessions in the network.



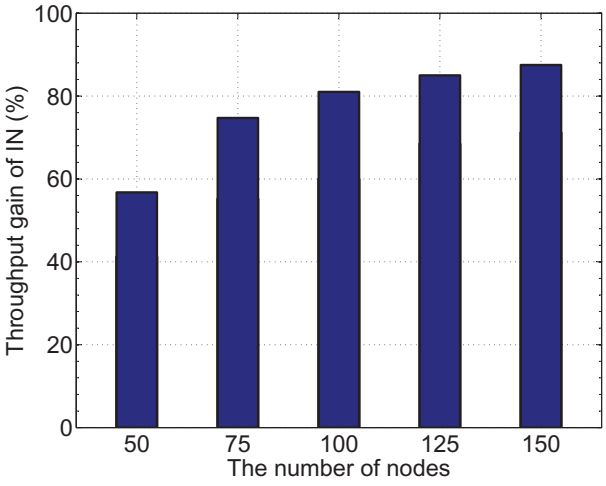
(a) 1-session case



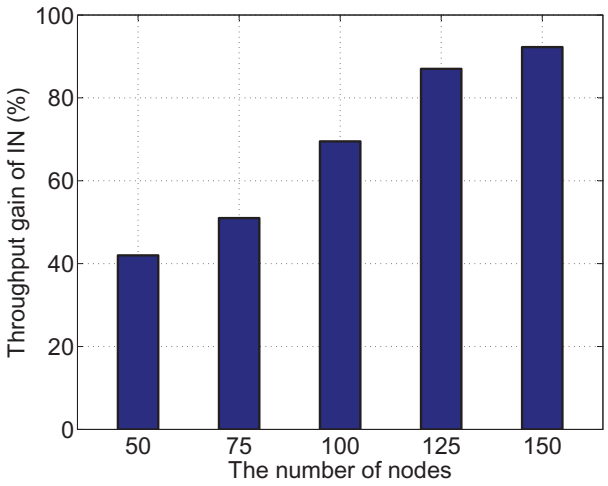
(b) 2-session case



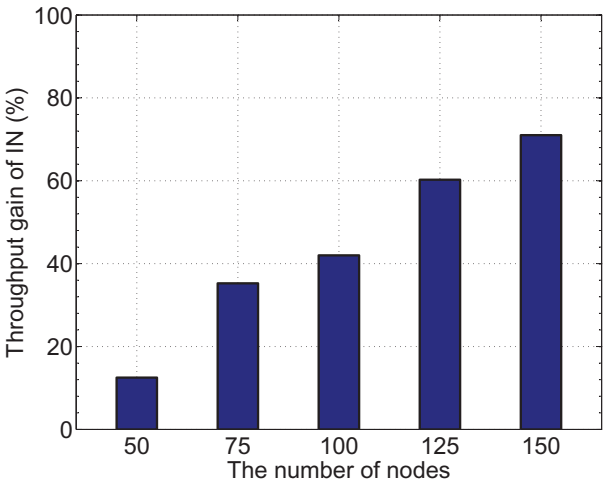
(c) 3-session case



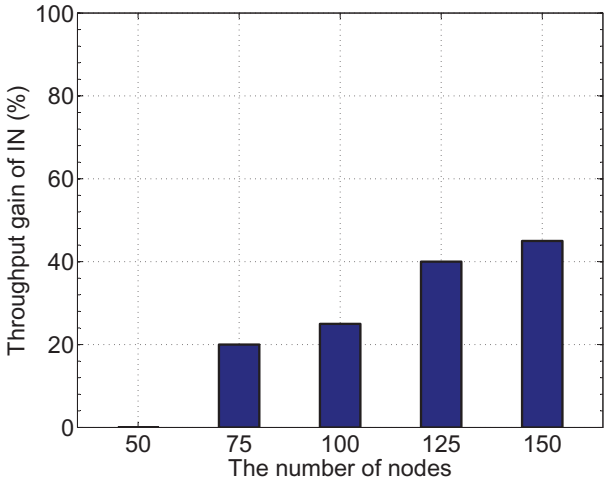
(d) 4-session case



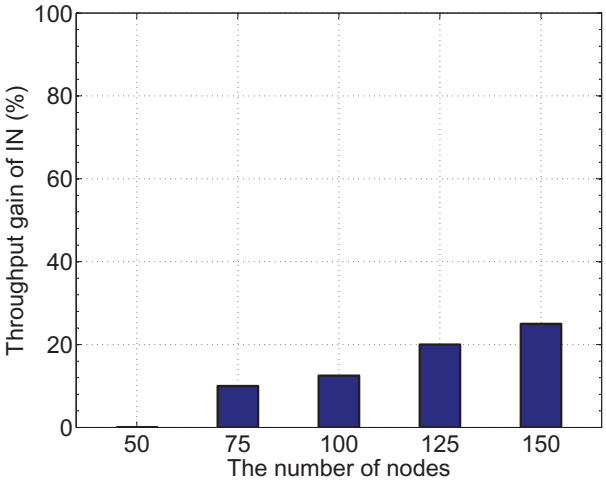
(e) 5-session case



(f) 6-session case



(g) 7-session case



(h) 8-session case

Figure 7.13: Impact of node density in the network.

Similar observations hold for other network sizes (Fig. 7.12(b)–(d)). Note that when the network size is 125-node, the maximum throughput gain reaches 85% when the number of sessions is 5.

Impact of Node Density. Based on our understanding in Fig. 7.12, we further explore the impact of node density on IN. Fig. 7.13 presents our simulation results. For each figure, the x-axis is the number of nodes in the network and the y-axis is the average throughput gain of IN over 100 randomly generated network instances.

As shown from Fig. 7.13(a) to (h), we observe that, for a given number of sessions in the network, the throughput gain of IN increases with node density. This is because the more nodes in the network, the more idle nodes are available to be used as neuts. The larger pool of neuts allows more interferences to be neutralized and thus allowing more transmissions to be carried out simultaneously. In the extreme case (not shown in the figures), when the node density is sufficiently large (e.g., ∞), all the interference can be neutralized and the network operates in an "interference-free" regime.

7.7 Chapter Summary

In this chapter, we studied IN in a general multi-hop wireless network. We found that a multi-hop wireless network (particularly a dense one) provides a fertile environment for us to unlock IN's potential. By identifying eligible idle nodes as neuts that can be exploited for IN, we establish a mathematical framework for neut selection, IN, and scheduling in a multi-hop wireless network. This mathematical framework allows us to determine the feasible space for a set of links can be active simultaneously through optimal neut selection, IN among interference links, and scheduling. As an application, we applied this framework to study a throughput maximization problem and showed that IN can indeed boost throughput performance. In particular, we found that the benefits of IN are most profound when the network is dense and there is sufficient number of idle nodes that can be chosen for IN.

Chapter 8

Summary and Future Work

8.1 Summary

In this dissertation, we studied three interference management techniques (IC, IA, and IN) in wireless networks, with the goal of exploring their potential capabilities from a networking perspective. The work in this dissertation consists of three parts: IC in distributed multi-hop MIMO networks, different forms of IA in multi-hop, cellular, and UWA networks, and IN in multi-hop single-antenna networks. Collectively, the findings constitute a body of work on interference management that contributes to the networking research community. The results offer insights and guidance on how different interference management techniques may be incorporated to improve network throughput performance.

In the first part (Chapter 2), we studied IC in multi-hop MIMO networks where resource allocation is achieved through neighboring node coordination and local information exchange. Based on a well-established DoF MIMO model, we developed a distributed DoF scheduling algorithm with the objective of maximizing network-level throughput while guaranteeing solution feasibility at the PHY layer. We showed that our developed algorithm has a number of merits, including polynomial-time complexity, amenability to local implementation, a guarantee of feasibility at the PHY layer, and competitive throughput performance. Our results offered a definitive "yes" answer to the question — Can the node-ordering DoF model

be deployed in a distributed multi-hop MIMO network? In particular, we showed that the essence of the DoF model — a global node ordering, can be implicitly achieved via local operations, albeit it is invisible to individual node.

In the second part (Chapter 3–6), we offered a comprehensive study on IA from a networking perspective. Specifically, we studied IA in three different domains: spatial domain, spectral domain, and temporal domain.

- In the spatial domain (Chapters 3 & 4), we studied IA for multi-hop MIMO networks. We derived a set of simple constraints to characterize IA capability at the PHY layer. We proved that as long as the set of simple constraints are satisfied, one can always construct precoding/decoding vectors at the PHY layer so that the data streams on each link can be transported free of interference. Therefore, instead of dealing with the complex design of precoding and decoding vectors, these IA constraints only require simple algebraic addition/subtraction operations. Such simplicity allows us to study network-level IA problems without being distracted by the tedious details in signal design at the PHY layer. Based on these IA constraints, we developed an optimization framework for unicast and multicast communications.
- In the spectral domain (Chapter 5), we studied IA in OFDM-based cellular networks. For the uplink, we derived a set of simple IA constraints to characterize a feasible DoF region for a cellular network. We showed how to construct the precoding and decoding vectors at the PHY layer so that each data stream can be transported free of interference. Based on the set of IA constraints, we studied a user throughput maximization problem and showed the throughput improvement over two other schemes via numerical results. For the downlink, we found that we could exploit the uplink IA constraints to the downlink case simply by reversing the roles of user and base station. Further, the downlink throughput maximization problem has the same formulation as the uplink problem and thus can be solved in the exactly same way.
- In the temporal domain (Chapter 6), we studied IA in multi-hop UWA networks. We proposed a temporal IA scheme based on propagation delay, nicknamed PD-IA, for

multi-hop UWA networks. We first derived a set of PD-IA constraints to guarantee PD-IA feasibility at the PHY layer. Then we developed a distributed PD-IA scheduling algorithm, called Shark-IA, to maximally overlap interference in a multi-hop UWA network. Our proposed Shark-IA algorithm has a number of merits, including polynomial-time complexity, amenability to local implementation, a guarantee of feasibility at the PHY layer, and competitive throughput performance. We showed that PD-IA can turn the adverse propagation delays to throughput improvement in multi-hop UWA networks.

In the third part (Chapter 7), we studied IN in multi-hop networks with a single antenna at each node. To apply IN in a multi-hop network, the fundamental problem is node selection. We first established an IN reference model to characterize the IN capability at the PHY layer. Based on this reference model, we developed a set of IN constraints that can be used to quickly determine whether or not a subset of links can be active simultaneously. By identifying each eligible neutralization node as a neut, we studied IN in a multi-hop network and derived the necessary constraints to characterize neut selection, IN, and scheduling behaviors. These constraints allow us to study IN problems from a networking perspective but without the need of getting into signal design issues at the PHY layer. By applying our IN constraints to study a throughput maximization problem, we showed that the use of IN could help increase network throughput. Further, we found that throughput gain is most significant when there is a sufficient number of neuts in the network.

8.2 Future Work

Interference is a fundamental problem in wireless networks. In a network environment, an effective solution to address interference usually calls for a cross-layer approach. Our work in this dissertation is among the first that jointly studies advanced physical-layer interference techniques with link-layer scheduling algorithms in wireless networks. Research in this area is still in its infancy and, as expected, there exist many open problems that remain to be explored. We outline some open problems as follows.

- **Open problems in Chapter 2 (IC in multi-hop networks).** In Chapter 2, based on the well-established node-ordering DoF model in [78], we developed a distributed DoF scheduling algorithm with the objective of maximizing network throughput while guaranteeing feasibility at the PHY layer. There are many important problems that remain to be addressed. First, our algorithm in Chapter 2 is heuristic, since we are not able to theoretically characterize the performance gap between the objective value of the solution yielded by our algorithm and that of an optimal solution. How to develop a distributed algorithm that can offer some theoretical conclusion on its performance remains open. Second, the algorithm that we developed in Chapter 2 can only work in a multi-hop network with stationary nodes and time-invariant traffics (see the assumptions that we made in Section 2.5.1). But in reality, a network may be far from being static due to node mobility and frequent birth/death of traffic sessions. How to devise a distributed algorithm that can adapt to the node mobility and traffic dynamics while ensuring its solution feasibility at the PHY layer is another open problem.
- **Open problems in Chapters 3 and 4 (spatial IA in multi-hop networks).** Our work in Chapters 3 and 4 studied spatial IA in multi-hop MIMO networks with unicast and multicast communication sessions, respectively. Some of open problems are listed as follows. First, our proposed spatial IA schemes in Chapters 3 and 4 rely on the assumption that each node in the network has the same number of antennas. Such an assumption may not hold in heterogeneous networks. It is thus desirable to devise a spatial IA scheme that can be applied to a network with any antenna configuration. Second, while there exist some results on blind IA scheme (i.e., IA scheme without requiring CSI at the transmitter side), most of them are limited to toy-sized networks (e.g., only three links). Little progress has been made so far in the development of blind IA schemes for general complex networks (e.g., multi-hop networks). Since it is prohibitively expensive to obtain CSI at the transmitter side in practical systems, developing a blind IA scheme is of significant practical value. Third, our proposed IA schemes are guaranteed to be feasible at the PHY layer. But there is no theoretical

characterization of their performance w.r.t. the optimum. An optimal IA scheme is of great importance for us to fully understand the capability of IA in a network.

- **Open problems in Chapter 5 (spectral IA in cellular networks).** In Chapter 5, we studied a spectral IA scheme for cellular networks. We identify some open problems as follows. First, the relationship between spectral IA and network topology remains unclear. Previous results (see, e.g., [84, 101]) indicate the potential benefits of spectral IA depend on the network topology. More research efforts are required to explore spectral IA in various network topologies, including both infrastructure-based networks (e.g., cellular networks) and infrastructure-less networks (e.g., wireless ad hoc networks). In particular, spectral IA in multi-hop ad hoc networks is an interesting but unexplored problem. Second, since cognitive radio (CR) networks will be adopted for future wireless communications, it is interesting to study spectral IA in CR networks. Given the heterogeneity of available spectral bands at each node, how to design an efficient spectral IA scheme is challenging. Third, how to jointly perform spectral IA and spatial IA in MIMO-OFDM networks is an interesting but open problem. Given the popularity of MIMO-OFDM in today's wireless networks, a hybrid technology combining IA in both spatial and spectral domains is appealing. But how to combine the best of both worlds is an open problem.
- **Open problems in Chapter 6 (temporal IA in UWA networks).** While Chapter 6 offered some preliminary results of temporal IA in UWA networks, research of temporal IA in the context of multi-hop networks is far from being mature. Some open problems are listed as follows. First, the Shark-IA algorithm requires relatively accurate propagation delay information between the transmitters and receivers. Obtaining such accurate delay information may be possible for a static network. But for a mobile network, it is hard to do this. So the question is: Will temporal IA be useful if the underlying network is mobile? Second, the Shark-IA algorithm is heuristic in nature. There is no theoretical conclusion on the performance of the Shark-IA algorithm. How to devise a PD-IA algorithm that can provide analytical performance guarantee is an

open problem. Third, the success of the PD-IA scheme relies on the presence of large propagation delays. Such an assumption holds in the UWA network environment. But in a terrestrial wireless network, signal propagation delays are negligible compared to the time duration of symbols. So the question is: Is temporal IA useful at all for terrestrial wireless networks?

- **Open problems in Chapter 7 (IN in multi-hop networks).** Our work in Chapter 7 is the first effort that studies IN for multi-hop networks. Some open problems are outlined as follows. First, our results are limited to multi-hop networks with a single antenna at each node. Since MIMO has been widely deployed in wireless networks (e.g., WiFi and cellular networks), it is necessary to extend IN from single-antenna to MIMO networks. But this extension is not straightforward. How to design an IN scheme that can also exploit MIMO's SM and IC capabilities in a multi-hop MIMO network is an open problem. Second, whether IN can help achieve transparent coexistence between the primary and secondary users in a CR network is an open problem. To achieve transparent coexistence, Yuan et al. [97] proposed an IC scheme to handle the interference between the secondary users and the primary users. But their IC scheme requires each secondary user to be equipped with multiple antennas. IN may relax this requirement for the secondary nodes. But how to design an IN scheme to achieve transparent coexistence is an open problem.

Bibliography

[1] A. Abdel-Hadi and S. Vishwanath, "On multicast interference alignment in multihop systems," in *Proc. IEEE Information Theory Workshop (ITW),* Cairo, Egypt, Jan. 2010.

[2] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks,* vol. 3, no. 3, pp. 257–279, Feb. 2005.

[3] S. Berger and A. Wittneben, "Cooperative distributed multiuser MMSE relaying in wireless ad-hoc networks," in *Proc. IEEE Asilomar Conference,* pp. 1072–1076, Pacific Grove, CA, Oct. 2005.

[4] R. Bhatia and L. Li, "Characterizing achievable multicast rates in multi-hop wireless networks," in *Proc. ACM MobiHoc,* pp. 133–144, Urbana-Champaign, IL, May 2005.

[5] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proc. IEEE INFOCOM,* pp. 2326–2330, Anchorage, AK, May 2007.

[6] F.L. Blasco, F. Rossetto, and G. Bauch, "Time interference alignment via delay offset for long delay networks," in *Proc. IEEE GLOBECOM*, pp. 590–599, Houston, TX, Dec. 2011.

[7] D. Blough, G. Resta, P. Santi, R. Srinivasan, and L.M. Cortes-Pena, "Optimal one-shot scheduling for MIMO networks," in *Proc. IEEE SECON,* pp. 377–385, Salt Lake City, Utah, June 2011.

[8] V.R. Cadambe and S.A. Jafar, "Degrees of freedom of wireless networks — What a difference delay makes," in *Proc. IEEE Asilomar Conference on Signals, Systems and Computers (ACSSC)*, pp. 133–137, Pacific Grove, CA, Nov. 2007.

[9] V.R. Cadambe and S.A. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. on Information Theory,* vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[10] V.R. Cadambe and S.A. Jafar, "Interference alignment and the degrees of freedom of wireless X networks," *IEEE Trans. on Information Theory,* vol. 55, no. 9, pp. 3893–3908, Sep. 2009.

[11] V.R. Cadambe, S.A. Jafar, and C. Wang, "Interference alignment with asymmetric complex signaling — Settling the Host-Madsen-Nosratinia conjecture," *IEEE Trans. on Information Theory,* vol. 56, no. 9, pp. 4552–4565, Sep. 2010.

[12] V. Chandrasekhar, W.K. Seah, Y.S. Choo, and H.V. Ee, "Localization in underwater sensor networks: Survey and challenges," in *Proc. ACM International Workshop on Underwater Networks,* pp. 33–40, Los Angeles, CA, Sep. 2006.

[13] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of networks with large propagation delays," *IEEE Journal of Oceanic Engineering,* vol. 37, no. 4, pp. 645–658, Oct. 2012.

[14] L.-U. Choi and R.D. Murch, "A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach," *IEEE Trans. on Wireless Communications,* vol. 3, no. 1, pp. 20–24, Jan. 2004.

[15] O. El Ayach, S.W. Peters, and R.W. Heath, "The feasibility of interference alignment over measured MIMO-OFDM channels," *IEEE Trans. on Vehicular Technology,* vol. 59, no. 9, pp. 4309–4321, Nov. 2010.

[16] A. El-Gamal, and Y.-H. Kim, *Network Information Theory,* Cambridge University Press, ISBN-13: 978-1107008731, New York, NY, 2012,

[17] C. Gao, Y. Shi, Y.T. Hou, and S. Kompella, "On the throughput of MIMO-empowered multi-hop cognitive radio networks," *IEEE Trans. on Mobile Computing,* vol. 10, no. 11, pp. 1505–1519, Nov. 2011.

[18] C. Gao, Y. Shi, Y.T. Hou, H.D. Sherali, and H. Zhou, "Multicast communications in multi-hop cognitive radio networks," *IEEE Journal on Selected Areas in Communications,* vol. 29, no. 4, pp. 784–793, April 2011.

[19] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman and Company, ISBN-13: 978-0716710455, New York, 1979.

[20] W. Ge, J. Zhang, and S. Shen, "A cross-layer design approach to multicast in wireless networks," *IEEE Trans. on Wireless Communications,* vol. 6, no. 3, pp. 1063–1071, March 2007.

[21] V. Genc, S. Murphy, Y. Yang, and J. Murphy, "IEEE 802.16J relay-based wireless access networks: An overview," *IEEE Wireless Communications,* vol. 15 , no. 5, pp. 56–63, Oct. 2008.

[22] D. Gesbert, M. Shafi, D. Shiu, P.J. Smith, and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications,* vol. 21, no. 3, pp. 281–302, Apr. 2003.

[23] S. Gollakotta, S. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proc. ACM SIGCOMM,* vol. 39 no. 4, pp. 159–170, Barcelona, Spain, Oct. 2009.

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

[25] T. Gou, S. A. Jafar, C. Wang, S.-W. Jeon, and S.-Y. Chung, "Aligned interference neutralization and the degrees of freedom of the $2 \times 2 \times 2$ interference channel," *IEEE Trans. on Information Theory,* vol. 58, no. 7, pp. 4381–4395, July 2012.

[26] T. Gou and S.A. Jafar, "Degrees of freedom of the K user $M \times N$ MIMO interference channel," *IEEE Trans. on Information Theory,* vol. 56, no. 12, pp. 6040–6057, Dec. 2010.

[27] L.H. Grokop, D.N. Tse, and R.D. Yates, "Interference alignment for line-of-sight channels," *IEEE Trans. on Information Theory,* vol. 57, no. 9, pp. 5820–5839, Sept. 2011.

[28] X. Guo, M. Frater, and M. Ryan, "Design of a propagation-delay- tolerant MAC protocol for underwater acoustic sensor networks," *IEEE Journal of Oceanic Engineering,* vol. 34, no. 2, pp. 170–180, Apr. 2009.

[29] B. Hamdaoui and K.G. Shin, "Characterization and analysis of multi-hop wireless MIMO network throughput," in *Proc. ACM MobiHoc,* pp. 120–129, Montreal, Quebec, Canada, Sep. 2007.

[30] S. Han, Y. Noh, U. Lee, and M. Gerla, "M-FAMA: A multi-session MAC protocol for reliable underwater acoustic streams," in *Proc. IEEE INFOCOM,* pp. 665–673, Turin, Italy, Apr. 2013.

[31] Z. Ho and E. Jorswieck, "Instantaneous relaying: Optimal strategies and interference neutralization," *IEEE Trans. on Signal Processing,* vol. 60, no. 12, pp. 6655–6668, Dec. 2012.

[32] Y.T. Hou, Y. Shi, and H.D. Sherali, "Optimal base station selection for anycast routing in wireless sensor networks," *IEEE Trans. on Vehicular Technology,* vol. 55, no 3, pp. 813–821, May 2006.

[33] Y.T. Hou, Y. Shi, H.D. Sherali, and J.E. Wieselthier, "Multicast communications in ad hoc networks using directional antennas: A lifetime-centric approach," *IEEE Trans. on Vehicular Technology,* vol. 56, no. 3, pp. 1333–1344, May 2007.

[34] Y.T. Hou, Y. Shi, and H.D. Sherali, "Spectrum sharing for multi-hop networking with cognitive radios," *IEEE Journal on Selected Areas in Communications,* vol. 26, no. 1, pp. 146–155, Jan. 2008.

[35] Y.T. Hou, Y. Shi, and H.D. Sherali, *Applied Optimization Methods for Wireless Networks,* Cambridge University Press, ISBN-13: 978-1107018808, April 2014.

[36] R.A. Horn and C.R. Johnson, *Matrix Analysis,* Cambridge University Press, ISBN-13: 978-0521386326, Feb. 1990.

[37] D. Hu and S. Mao, "Cooperative relay with interference alignment for video over cognitive radio networks," in *Proc. IEEE INFOCOM,* pp. 2014–2022, Orlando, FL, March 2012.

[38] S. Jafar and M. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Trans. on Information Theory,* vol. 53, no. 7, pp. 2637–2642, July 2007.

[39] S.A. Jafar and S. Shamai, "Degrees of freedom region for the MIMO X channel," *IEEE Trans. on Information Theory,* vol. 54, no. 1, pp. 151–170, Jan. 2008.

[40] S.A. Jafar, "Interference alignment: A new look at signal dimensions in a communication network," *Foundations and Trends in Communications and Information Theory,* vol. 7, no. 1, pp. 1-136, 2010.

[41] S.A. Jafar, "The ergodic capacity of phase-fading interference networks," *IEEE Trans. on Information Theory,* vol. 57, no. 12, pp. 7685–7694, Dec. 2011.

[42] S.A. Jafar, "Blind interference alignment," *IEEE Journal of Selected Topics in Signal Processing,* vol. 6, no. 3, pp. 216–227, June 2012.

[43] X. Jia and L. Wang, "A group multicast routing algorithm by using multiple minimum Steiner trees," *Computer Communications,* vol. 20, no. 9, pp. 750–758, Sept. 1997.

[44] C. Jiang, Y. Shi, Y.T. Hou, S. Kompella, "On the asymptotic capacity of multi-hop MIMO ad hoc networks," *IEEE Trans. on Wireless Communications,* vol. 10, no. 4, pp. 1032–1037, Apr. 2011.

[45] B.C. Jung and W.Y. Shin, "Opportunistic interference alignment for interference-limited cellular TDD uplink," *IEEE Communications Letters,* vol. 15, no. 2, pp. 148–150, Feb. 2011.

[46] S.-J. Kim, X. Wang, and M. Madihian, "Cross-layer design of wireless multihop backhaul networks with multiantenna beamforming," *IEEE Trans. on Mobile Computing,* vol. 6, no. 11, pp. 1259–1269, Nov. 2007.

[47] K. Kredo, P. Djukic, and P. Mohapatra, "STUMP: Exploiting position diversity in the staggered TDMA underwater MAC protocol," in *Proc. IEEE INFOCOM,* pp. 2961–2965, Rio de Janeiro, Brazil, Apr. 2009.

[48] N. Lee, J.B. Lim, and J. Chun, "Degrees of freedom of the MIMO Y channel: Signal space alignment for network coding," *IEEE Trans. on Information Theory,* vol. 56, no. 7, pp. 3332–3342, July 2010.

[49] N. Lee and C. Wang, "Aligned interference neutralization and the degrees of freedom of the two-user wireless networks with an instantaneous relay," *IEEE Trans. on Communications,* vol. 61, no. 9, pp. 3611–3619, Sept. 2013.

[50] B. Li, J. Huang, S. Zhou, K. Ball, M. Stojanovic, L. Freitag, and P. Willett, "MIMO-OFDM for high-rate underwater acoustic communications," *IEEE Journal of Oceanic Engineering,* vol. 34, no. 3, pp. 634–644, Oct. 2009.

[51] L.E. Li, R. Alimi, D. Shen, H. Viswanathan, and Y.R. Yang, "A general algorithm for interference alignment and cancellation in wireless networks," in *Proc. IEEE INFOCOM,* pp. 1774–1782, San Diego, CA, March 2010.

[52] Y.-H. Lin, T. Javidi, R.L. Cruz, and L.B. Milstein, "Distributed link scheduling, power control and routing for multi-hop wireless MIMO networks," in *Proc. IEEE Asilomar Conference on Signals, Systems and Computers (ACSSC),* pp. 122–126, Grove, CA, Oct. 2006.

[53] K. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," in *Proc. ACM SIGCOMM,* pp. 146–157, Toronto, Canada, Aug. 2011.

[54] J. Llor, M. Stojanovic, and M.P. Malumbres, "A simulation analysis of large scale path loss in an underwater acoustic network," in *Proc. IEEE OCEANS-Spain*, pp. 1–5, Santander, Spain, June 2011.

[55] J. Ma and W. Lou, "Interference-aware spatio-temporal link scheduling for long delay underwater sensor networks," in *Proc. IEEE SECON,* pp. 431–439, Salt Lake City, UT, June 2011.

[56] H. Maier and R. Mathar, "Cyclic interference neutralization on the full-duplex relay-interference channel," in *Proc. IEEE International Symposium on Information Theory (ISIT),* pp. 2309–2313, Istanbul, Turkey, July 2013.

[57] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra,* Society for Industrial and Applied Mathematics (SIAM), ISBN-13: 978-0898714548, Feb. 2001.

[58] S. Mohajer, S. Diggavi, C. Fragouli, and D. Tse, "Transmission techniques for relay-interference networks," in *Proc. Allerton Conference,* pp. 467–474, Urbana-Champaign, IL, Sep. 2008

[59] S. Mohajer, D. Tse, and S. Diggavi, "Approximate capacity of a class of Gaussian relay-interference networks," in *Proc. IEEE International Symposium on Information Theory (ISIT),* pp. 31–35, Seoul, South Korean, June 2009.

[60] S. Mohajer, S. Diggavi, C. Fragouli, and D. Tse, "Approximate capacity of a class of Gaussian interference-relay networks," *IEEE Trans. on Information Theory,* vol. 57, no. 5, pp. 2837–2864, May 2011.

[61] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC protocol for underwater acoustic networks," in *Proc. IEEE OCEANS — Asia Pacific,* pp. 1–7, Singapore, May 2007.

[62] J. Moy, *Multicast Extensions to OSPF,* IETF, RFC 1584, March 1994.

[63] B. Mumey, J. Tang, and T. Hahn, "Algorithmic aspects of communications in multihop wireless networks with MIMO links," in *Proc. IEEE ICC,* pp. 1–6, Cape Town, South Africa, July 2010.

[64] J. Mundarath, P. Ramanathan, and B.D. Van Veen, "Exploiting spatial multiplexing and reuse in multi-antenna wireless ad hoc networks," *Elsevier Ad Hoc Networks,* vol. 7, no. 2, pp. 281–293, March 2009.

[65] B. Nazer, M. Gastpar, S.A. Jafar, and S. Vishwanath, "Ergodic interference alignment," in *Proc. IEEE International Symposium on Information Theory (ISIT),* pp. 1769–1773, Seoul, South Korea, July, 2009.

[66] B. Niu and A.M. Haimovich, "Interference subspace tracking for network interference alignment in cellular systems," in *Proc. IEEE GLOBECOM,* Honolulu, HI, Nov. 2009.

[67] Y. Noh, P. Wang, U. Lee, D. Torres, and M. Gerla, "DOTS: A propagation delay-aware opportunistic MAC protocol for underwater sensor networks," in *Proc. IEEE ICNP,* pp. 183–192, Kyoto, Japan, Oct. 2010.

[68] J.-S. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: Enabling spatial reuse using MIMO channel-aware MAC," in *Proc. IEEE ICC,* pp. 3642–3646, Seoul, South Korea, May 2005.

[69] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks," *IEEE Communication Letters,* vol. 11, no. 12, pp. 1025–1027, Dec. 2007.

[70] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing,* IETF RFC 3561, July 2003.

[71] D. Qian, D. Zheng, J. Zhang, and N. Shroff, "CSMA-based distributed scheduling in multi-hop MIMO networks under SINR model," in *Proc. IEEE INFOCOM,* pp. 2865–2873, San Diego, CA, March 2010.

[72] B. Rankov and A. Wittneben, "Spectral efficient protocols for half-duplex fading relay channels," *IEEE Journal on Selected Areas Communications,* vol. 25, no. 5, pp. 379–389, Feb. 2007.

[73] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, "Coordinated multipoint transmission/reception techniques for LTE-advanced," *IEEE Wireless Communications,* vol. 17, no. 3, pp. 26–34, June 2010.

[74] A. Schrijver, *Theory of Linear and Integer Programming,* WileyInterscience, ISBN-13: 978-0471982326, New York, NY, June 1998.

[75] S. Sharma, Y. Shi, Y.T. Hou, H.D. Sherali, and S. Kompella, "Cooperative communications in multi-hop wireless networks: Joint flow routing and relay node assignment," in *Proc. IEEE INFOCOM,* pp. 2016–2024, San Diego, CA, March 2010.

[76] H.D. Sherali and W.P. Adams, *A Reformulation—Linearization Technique for Solving Discrete and Continuous Nonconvex Problems,* Kluwer Academic Publishers, ISBN-13: 978-1441948083, Dec. 1998.

[77] Y. Shi, Y.T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," in *Proc. ACM MobiHoc,* pp. 239–248, New Orleans, LA, May, 2009.

[78] Y. Shi, J. Liu, C. Jiang, C. Gao, and Y.T. Hou, "An optimal MIMO link model for multi-hop wireless networks," in *Proc. IEEE INFOCOM,* pp. 1916–1924, Shanghai, China, April 2011.

[79] Y. Shi, Y.T. Hou, J. Liu, and S. Kompella, "Bridging the gap between protocol and physical models for wireless networks," *IEEE Trans. on Mobile Computing,* vol. 12, no. 7, pp. 1404–1416, July 2013.

[80] R. Solis, V.S. Borkar, and P.R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. IEEE Conference on Decision and Control,* pp. 2734–2739, San Diego, CA, Dec. 2006.

[81] E. Sozer, M. Stojanovic, and J. Proakis, "Underwater acoustic networks," *IEEE Journal of Oceanic Engineering,* vol. 25, no. 1, pp. 72–83, Jan. 2000.

[82] Q.H. Spencer, A.L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. on Signal Processing,* vol. 52, no. 2, pp. 461–471, Feb. 2004.

[83] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Communications Magazine,* vol. 47, no. 1, pp. 84–89, Jan. 2009.

[84] C. Suh and D. Tse, "Interference alignment for cellular networks," in *Proc. Allerton Conference on Communication, Control, and Computing,* pp 1037–1044, Urbana-Champaign, IL, Sept. 2008.

[85] C. Suh, M. Ho, and D. Tse, "Downlink interference alignment," *IEEE Trans. on Communications,* vol. 59, no. 9, pp. 2616–2626, Sept. 2011.

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

[87] K. Sundaresan, W. Wang, and S. Eidenbenz, "Algorithmic aspects of communication in ad-hoc networks with smart antennas," in *Proc. ACM MobiHoc*, pp. 298–309, Florence, Italy, May 2006.

[88] A.A. Syed and J.S. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. IEEE INFOCOM*, pp. 1–12, Barcelona, Spain, Apr. 2006.

[89] R. Tresch and M. Guillaud, "Cellular interference alignment with imperfect channel knowledge," in *Proc. IEEE ICC Workshops,* pp. 1–5, Dresden, Germany, June 2009.

[90] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication,* Cambridge University Press, ISBN-13: 978-0521845274, Cambridge, UK, 2005.

[91] C.S. Vaze and M.K. Varanasi, "Beamforming and aligned interference neutralization achieve the degrees of freedom region of the $2 \times 2 \times 2$ MIMO interference network," in *IEEE Information Theory and Application Workshop,* pp. 199–203, San Diego, CA, Feb. 2012.

[92] C. Wang, T. Gou, and S.A. Jafar, "Aiming perfectly in the dark — Blind interference alignment through staggered antenna switching," *IEEE Trans. on Information Theory,* vol. 59, no. 6, pp. 2734–2744, Jun. 2011.

[93] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM,* pp. 585–594, Tel Aviv, Israel, March 2000.

[94] X. Xie, X. Zhang, and K. Sundaresan, "Adaptive feedback compression for MIMO networks," in *Proc. ACM MobiCom,* pp. 477–488, Miami, FL, Sep. 2013.

[95] J. Xu, S.J. Lee, W.S. Kang, and J.S. Seo "Adaptive resource allocation for MIMO-OFDM based wireless multicast systems," *IEEE Transactions on Broadcasting,* vol. 56, no. 1, pp. 98–102, March 2010.

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

[97] X. Yuan, C. Jiang, Y. Shi, Y.T. Hou, W. Lou, and S. Kompella, "Beyond interference avoidance: On transparent coexistence for multi-hop secondary CR networks," in *Proc. IEEE SECON,* pp. 398–405, New Orleans, LA, June 2013.

[98] S. Zaks, "Optimal distributed algorithms for sorting and ranking," *IEEE Trans. on Computers,* vol. 5, no. 1, pp. 376–379, April 1985.

[99] G. Zeng, B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Multicast algorithms for multi-channel wireless mesh networks," in *Proc. of IEEE ICNP*, pp. 1–10, Beijing, China, Oct. 2007.

[100] H. Zeng, Y. Shi, Y.T. Hou, W. Lou, S. Kompella, and S.F. Midkiff, "On interference alignment for multi-hop MIMO networks," in *Proc. IEEE INFOCOM,* pp. 1330–1338, Turin, Italy, April 14–19, 2013.

[101] H. Zeng, Y. Shi, Y.T. Hou, W. Lou, X. Yuan, R. Zhu, and J. Cao, "Increasing user throughput in cellular networks with interference alignment," in *Proc. IEEE SECON,* pp. 346–353, Singapore, June 2014.

[102] X. Zhang, K. Sundaresan, M.A. Khojastepour, S. Rangarajan, and K.G. Shin, "NEMOx: Scalable network MIMO for wireless networks," in *Proc. ACM MobiCom,* pp. 453–464, Miami, FL, Sep. 2013.

[103] L. Zheng and D.N.C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. on Information Theory,* vol. 49, no. 5, pp. 1073–1096, May 2003.

[104] IBM ILOG CPLEX Optimization solver, software available at http://www-01.ibm.com/software/integration/optimization/cplex-optimizer