Resource Allocation and Scheduling in Wireless Mesh Networks

by

Keivan Ronasi

B.Sc., Isfahan University of Technology, 2004 M.Sc., University of Tehran, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA (Vancouver)

July 2012

 \bigodot Keivan Ronasi, 2012

Abstract

The unreliability of wireless mesh networks creates challenge in designing high performance wireless networks in terms of network throughput, end-to-end delay, and fairness provisioning. In this thesis, the goal is to improve the network performance in terms of these metrics. We explore several techniques such as multipath routing, channel coding, network coding, and interference alignment. We consider resource allocation both in terms of average data rate provisioning and scheduling policies in a time slot basis.

First, we propose data rate and channel code rate allocation algorithms for networks with multiple paths to maximize the network throughput while all users can fairly exploit the network resources. We study the effect of adaptive and non-adaptive channel coding schemes. We also consider the end-to-end delay that each network flow experiences for data transmission. For that purpose, we formulate the problem of decreasing the end-to-end delay for network flows while improving the network throughput. Simulation results show that we can decrease the delay at the cost of a slight decrease in network throughput. We also formulate a data rate allocation problem in networks with network coding. Simulation results show that considering link reliabilities in the network coding design dramatically increases the network performance.

Data rate allocation algorithms provide the average data rates at which the source must transmit data. They do not determine scheduling on a time slot basis. To address that, we consider transmission scheduling in wireless networks. We also compare the suggested algorithm with a centralized optimal data rate allocation algorithm to verify that our algorithm follows the optimal solution. Through simulations, we show that fairness provisioning leads to higher network performance. We show that the suggested algorithm outperforms the current algorithms in the literature in terms of both network throughput and fairness provisioning.

Finally, we consider transmission scheduling in wireless multi-input multi-output (MIMO) systems. We formulate the problem of joint scheduling, interference alignment, and admission control in those networks and use Lyapunov stability theory to solve it. We also develop a heuristic approach to solve the problem in a semi-distributed manner.

Preface

Hereby, I declare that I am the first author of this thesis. Chapters 2-6 encompass work that has been published or is under review and they are co-authored by Dr. Vincent Wong and Dr. Sathish Gopalakrishnan who supervised me through this research. Chapters 2, 3, and 4 also include published work co-authored by Dr. Robert Schober and Dr. Amir-Hamed Mohsenian-Rad, who provided valuable comments for the works. The following publications are accomplished through this research.

Journal Papers, Published

- Keivan Ronasi, Amir-Hamed Mohsenian-Rad, Vincent W.S. Wong, Sathish Gopalakrishnan, and Robert Schober, "Delay-Throughput Enhancement in Wireless Networks With Multipath Routing and Channel Coding," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 3, pp. 1116-1123, Mar. 2011.
- Keivan Ronasi, Amir-Hamed Mohsenian-Rad, Vincent W.S. Wong, Sathish Gopalakrishnan, and Robert Schober, "Optimal Data Transmission and Channel Code Rate Allocation in Multipath Wireless Networks," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 8, pp. 3963-3974, Oct. 2011.
- Keivan Ronasi, Vincent W.S. Wong, and Sathish Gopalakrishnan, "Distributed Scheduling in Multihop Wireless Networks with Maxmin Fairness Provisioning," *IEEE Trans.* on Wireless Communications, vol. 11, no. 5, pp. 1753-1763, May 2012.

Conference Papers, Published

- Keivan Ronasi, Sathish Gopalakrishnan, and Vincent W.S. Wong, "Flow Starvation Mitigation for Wireless Mesh Networks," in *Proc. of IEEE Wireless Communications* and Networking Conference (WCNC), Budapest, Hungary, April 2009.
- Keivan Ronasi, Amir-Hamed Mohsenian-Rad, Vincent W.S. Wong, Sathish Gopalakrishnan, and Robert Schober, "Reliability-Based Rate Allocation in Wireless Inter-Session Network Coding Systems," in *Proc. of IEEE Global Communications Conference (Globecom)*, Honolulu, HI, December 2009.
- Keivan Ronasi, Amir-Hamed Mohsenian-Rad, Vincent W.S. Wong, Sathish Gopalakrishnan, and Robert Schober, "Optimal Data Transmission and Channel Code Rate Allocation in Multi-Path Wireless Networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Kyoto, Japan, June 2011.

Table of Contents

Ał	Abstract							
\mathbf{Pr}	Preface iv							
Та	Table of Contents vi							
Lis	st of	Tables	3					
Lis	st of	Figure	es xi					
Lis	st of	Abbre	eviations					
Ac	knov	vledgn	nents					
1 Introduction								
	1.1	Perfor	mance Attributes					
		1.1.1	Fairness					
		1.1.2	Reliability 4					
	1.2	Techni	ques					
		1.2.1	Network Utility Maximization					
		1.2.2	Network Coding					
		1.2.3	Channel Coding					
		1.2.4	Multipath Routing					

		1.2.5	Interference Alignment	8
	1.3	Mathe	ematical Background	10
		1.3.1	Convex Optimization	10
		1.3.2	Geometric Programming	12
		1.3.3	Non-linear Mixed Integer Programming	13
		1.3.4	Generalized Benders Decomposition (GBD)	14
	1.4	Contri	ibutions and Results	16
	1.5	Thesis	Organization	20
າ	Ont	imal I	Data Transmission and Channel Code Rate Allocation	91
2	0 pt	Introd	uction	21 91
	2.1	nitrou		21
	2.2	Syster	n Model And Problem Formulation	24
		2.2.1	System Model	24
		2.2.2	Problem Formulation	28
	2.3	Optim	al Transmission Rate and Channel Code Rate Allocation \ldots .	29
		2.3.1	Adaptive Channel Coding	29
		2.3.2	Non-adaptive Channel Coding	34
	2.4	Perfor	mance Evaluation	36
		2.4.1	Multipath vs. Single-path Routing	37
		2.4.2	Channel Coding vs. No Channel Coding	38
		2.4.3	Convergence Properties of Algorithm 2.1	39
		2.4.4	Impact of Various Design and System Parameters	41
		2.4.5	Adaptive vs. Non-adaptive Channel Coding	42
		2.4.6	The Effect of Dynamic Changes on the System Performance \ldots .	45
		2.4.7	The Effect of Mobility on the System Performance	45
		2.4.8	Impact of Fading	47

Table of C

	2.5	Summary	49
3	Del	ay-Throughput Enhancement	51
	3.1	Introduction	51
	3.2	System Model	53
	3.3	Delay-aware Optimal Data Rate and Coding Rate Allocation	58
	3.4	Numerical Results	63
	3.5	Summary	68
4	Rel	iability-based Rate Allocation with Inter-session Network Coding	69
	4.1	Introduction	69
	4.2	System Model	71
		4.2.1 Pair-wise Inter-session Network Coding	71
		4.2.2 Rate Allocation	73
		4.2.3 Interference	73
		4.2.4 Link Failure	74
		4.2.5 Network Utility Maximization Formulation	75
	4.3	End-to-End Reliability	76
	4.4	Reliability-based Rate Allocation	78
	4.5	Performance Evaluation	83
	4.6	Summary	86
5	Dis	tributed Scheduling with Maxmin Fairness Provisioning	88
	5.1	Introduction	88
	5.2	System Model	91
	5.3	Decentralized and Stable Scheduling	95
	5.4	Geometric Programming Formulation	106

	5.5	Performance Evaluation	110
	5.6	Summary	115
6	Sche	eduling and Interference Alignment for MIMO Wireless Systems .	118
	6.1	Introduction	118
	6.2	System Model	120
	6.3	Problem Formulation	124
	6.4	Scheduling and Interference Alignment Algorithm	129
	6.5	Semi-Distributed Scheduling and Interference Alignment Algorithm	137
	6.6	Performance Evaluation	140
	6.7	Summary	144
7	Con	clusions and Future Work	146
	7.1	Research Contributions	146
	7.2	Suggestions for Future Work	148
Bi	bliog	raphy	150

List of Tables

List of Figures

2.1	An example downtown area with 25 access points (forming a wireless mesh	
	infrastructure). The access point at the center serves as the gateway. There	
	are 10 vehicles in the system, each one uses the nearest access point to	
	connect to the Internet	27
2.2	A sample network topology with 20 nodes randomly located in a 5×5 grid.	
	The network includes five sessions: $1 \rightarrow 16, 3 \rightarrow 13, 2 \rightarrow 8, 14 \rightarrow 17$, and	
	$6 \rightarrow 20$. There are 4, 2, 2, 1, and 3 routing paths available for these sessions,	
	respectively.	37
2.3	Comparison between the performance of adaptive channel coding in single-	
	path and multipath routing networks in terms of the achieved normalized	
	minimum throughout.	38
2.4	Comparison between the performance of multipath routing with and with-	
	out per-link channel coding in terms of the achieved normalized minimum	
	throughput among the end-to-end sessions, when the scale of the network	
	increases and the number of nodes varies from 6 to 42	40
2.5	Convergence of Algorithm 2.1 with respect to solving problem (2.8) . We	
	can see that the algorithm converges after 50 iterations	40
2.6	The impact of the choice of design parameter N_e in approximation (2.12).	
	The average optimality error decreases as N_e increases. It becomes almost	
	zero for $N_e > 12$	41

2.7	The impact of choosing different coding block lengths T on the network	
	performance for three different random topologies. We observe that the	
	performance improves when the coding block length increases. \ldots .	43
2.8	Comparison between adaptive and non-adaptive channel coding	43
2.9	Optimal non-adaptive code rate versus adaptive code rate distribution among	
	all wireless links of the network topology in Fig. 2.2	44
2.10	Convergence speed for Algorithm 2.1 in presence of dynamic changes in the	
	network. We compare two cases where the previous solution is exploited	
	and the case where the previous solution is ignored. Every 100 time slots:	
	(a) five links are added and five links are removed, (b) a new pair of source-	
	destination nodes is added, (c) a pair of source-destination node is removed,	
	(d) a random pair is added and another random pair is removed	46
2.11	The convergence of the algorithm is shown for different speeds. Recalcula-	
	tions occur every 5 seconds if needed	47
2.12	Performance of the algorithm is studied in the downtown area of Fig. 2.1	
	when vehicles move with different speeds	48
2.13	Performance trend in a fading channel for 50 channel snapshots	49
3.1	A sample topology with five unicast multipath data sessions. Data sessions	
	are $(2 \to 15), (7 \to 18), (9 \to 20), (12 \to 23)$, and $(3 \to 25)$. These sessions	
	have 6, 1, 2, 2, and 4 available paths, respectively.	54
3.2	The average delay decreases as its importance weight in the objective func-	
	tion increases. Delay values are normalized over the corresponding values	
	for which $w = 0$. We observe a decrease of almost 58% when $w = 0.5$	64

3.3	The normalized maximum delay among the users in the network decreases	
	as the corresponding importance weight in the objective function increases.	
	Delay values are normalized over the values corresponding to $w = 0$. We	
	observe almost 37% decrease in the maximum delay in the network when w	
	reaches 0.5	65
3.4	The trade-off between the maximum delay and the minimum throughput in	
	the network. We may move on the curve by tuning w , the delay importance	
	weight. Delay and throughput are normalized over their corresponding val-	
	ues at $w = 0$	66
3.5	Fairness index is shown for both problem (3.8) (aggregate utility maximiza-	
	tion) and problem (3.9) (maxmin fairness)	66
3.6	Average maximum delay is shown when there is no guarantee on the maxi-	
	mum delay, there is a guarantee of $10 ms$, and $20 ms$. Since the guarantee	
	is imposed on the upper bound of the delay, the delay may not reach the	
	guaranteed value	67
4.1	(a) An example directed acyclic network graph with 6 nodes and 7 links.	
	One data session exists between s_1 and t_1 and another one exists between	
	s_2 and t_2 . (b) The corresponding (undirected) contention graph	72
4.2	Aggregate network throughput in all 10 simulated topologies $with$ and $with$ -	
	out reliability consideration when the utility functions are selected as $U_i(x) =$	
	x for all $i \in \mathcal{S}$. Taking reliability information into account increases the	
	throughput by 36.2% on average, among all ten scenarios.	84

- 4.3 Comparison between our algorithm (with reliability consideration) and the rate allocation algorithm which does *not* use link reliability information:
 (a) Maximizing the aggregate network throughput. (b) Maximizing the aggregate network utility when the utility functions are logarithmic. 85

- 5.2 The performance of the DisF algorithm is compared with the solution of centralized geometric programming problem. Each point represents the average performance obtained over 50 random topologies. The DisF algorithm follows the centralized approach as the number of flows increases. 111
- 5.3 In a sample topology, fairness is provided under the DisF algorithm while some flows starve under the DisA algorithm that does not consider fairness. 112
- 5.5 The tradeoff between minimum throughput in the network (a) and aggregate throughput in the network (b) is shown under both DisF and DisA algorithms.113

5.6	Performance of the network with channel coding is compared with the case	
	in which channel coding is not being used	114
5.7	The effect of increasing parameter V is shown on the minimum throughput	
	(a). This is at the expense of increasing the congestion in the network (b).	115
5.8	DisF algorithm is compared with the utility-based approach in terms of	
	fairness provisioning using Jain's fairness index. It is shown that for large	
	values of β , the utility-based approach has a better performance comparing	
	with DisF algorithm. This is at the cost of a decrease in the minimum	
	throughput. (Fig. 5.9)	116
5.9	DisF algorithm is compared with the utility-based approach when different	
	number of flows are using the network. It is shown that increasing β is at	
	the cost of decreasing the minimum throughput	116
6.1	Network topology for wireless MIMO system with K users	121
6.2	Convergence of SIA algorithm is shown in one time slot when there are 5	
	and 10 users in the network	141
6.3	The optimal value of problem (6.20) changes for both the SIA and SDSIA	
	algorithms when interference leakage threshold ϵ and receiver threshold P_{th}	
	change	142
6.4	The number of transmissions per user is shown when the number of users	
	K increases	143
6.5	The average per user achievable data rate is shown when the number of	
	users increases from 1 to 20	144

List of Abbreviations

ACK	Acknowledgement
AODV-BR	Ad hoc On-demand Distance Vector with Backup Routing
BEC	Binary Erasure Channel
BPSK	Binary Phase Shift Keying
CSMA	Carrier Sense Multiple Access
CSO	Channel State Only
DisA	Distributed Aggregate
DisF	Distributed Fair
DoF	Degrees of Freedom
FTP	File Transfer Protocol
GBD	Generalized Benders Decomposition
GMS	Greedy Maximal Scheduling
GP	Geometric Programming
HTTP	Hypertext Transfer Protocol
IA	Interference Alignment
LAN	Local Area Network
MAC	Medium Access Control
MAODV	Multipath Ad hoc On-demand Distance Vector
MIMO	Multiple-Input Multiple-Output

MIP	Mixed Integer Programming
MP-DSR	Multi-Path Dynamic Source Routing
MSR	Multipath Source Routing
NUM	Network Utility Maximization
OA	Outer Approximation
SIA	Scheduling and Interference Alignment
P2P	Peer-to-peer
QoS	Quality of Service
SDSIA	Semi-Distributed Scheduling and Interference Alignment
SMR	Split Multipath Routing
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network

Acknowledgments

First of all, I want to express my deepest sense of gratitude to my supervisors Dr. Vincent Wong and Dr. Sathish Gopalakrishnan without whom this research would not be possible. I want to thank them for their continuous support, patience, technical and non-technical guidance that helped me accomplish this research. I also want to thank them for their understanding of the conditions and problems that I was involved with as a new international student.

I want to declare my special thanks to Dr. Robert Schober and Dr. Amir Hamed Mohsenian-Rad for their enormous help, advice, and comments through my research. It was not possible to accomplish some parts without their very helpful advice.

I am thankful to my friends Ahmad Ashouri, Saloome Motavas, Behnam Faraji, Sahar Edelkhani, Pedram Samadi, Vahid Shahmansouri, and Ehsan Vahedi for their always being supportive and accountable. I also want to thank Man Hon, Nina, Nazanin, Azadeh, Nassim, Anahita, Sadaf, Helia, Payman, Morteza, Ali, and all my other friends that definitely I cannot name them all here.

Most importantly, I feel indebted to my great family, my father, my mother and my sister for their everlasting love, support, and protectivity not only in these last four years but also throughout my entire life.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Chapter 1

Introduction

Recent technological advances in wireless communications, digital electronics, and radio frequency systems have placed wireless networks at the forefront of today's data transmission systems. Wireless mesh networks are wireless systems in which no central infrastructure exists. In these networks, users (mesh nodes) send and receive data to and from the gateway(s) through possibly multiple hops. Having no central structure, mesh nodes are responsible for routing and relaying data to the destination nodes. This is in contrary to the existence of routers and access points in wired and infrastructure based wireless networks.

Since network users are responsible for routing and relaying in the network wireless mesh networks are scalable. This means that increasing the number of users (mesh nodes) will not only maintain the performance of the network, but it may also improve it. That is because a higher number of nodes (resources) are available for data transmission. Moreover, because of not relying on infrastructure, wireless mesh networks are inexpensive and easy to deploy. This makes these networks suitable for certain applications such as disaster recovery support, urgent applications, and military missions.

Wireless nature and the decentralized structure of the wireless mesh networks makes them exposed to unreliability. This means that at each instant of time, a packet may be corrupted because of node outages, weather conditions, and environmental obstacles. Unreliability degrades the performance of the network especially in applications of high quality of service (QoS) requirements such as multimedia and realtime applications. Another issue in wireless networks is interference. In the absence of a central node managing data transmissions in the network, each node needs to use a distributed algorithm to schedule data transmissions such that no interference occurs among the wireless channels. Fairness provisioning, admission control, and end-to-end delay are other issues that affect the performance of wireless mesh networks.

In this thesis, we study the performance improvement in wireless mesh networks. We develop rate allocation approaches, flow control algorithms, and scheduling policies such that the performance of the network in terms of network throughput and end-to-end delay improves while data transmissions remain reliable. We also consider fairness provisioning while achieving our goals. To this end, we use several techniques such as multipath routing, channel coding, network coding, and interference alignment. In this chapter, we first discuss the fairness and reliability in wireless networks. Then, we briefly summarize network utility maximization, network coding, channel coding, multipath routing, and interference alignment. After that, we describe the mathematical background that are used in this thesis. Finally, we itemize our contributions and describe the thesis organization at the end of this chapter.

1.1 Performance Attributes

1.1.1 Fairness

One important measure for network performance is fairness. Fairness provisioning in the network allows all users enjoy a fair portion of network resources according to their utility functions. Without considering fairness in a wireless network, some users may starve while the others over utilize the resources in the network.

Starvation may have its origins in different layers such as medium access control (MAC)

layer and transport layer. Shi *et al.* considered the starvation caused with a one-hop transmission for a two-hop transmission in [1]. They especially considered the case in which two users compete for the wireless channel with initially equal contention windows. After several failures when their contention windows are large enough, one of them gets successful in transmission and therefore it resets its contention window to the minimum default size. However, the contention window of the other one remains large and this leads the previously successful user to send its data more aggressively and makes the other one starve.

Another origin of starvation is unfair data rate allocation. When the goal is maximizing the aggregate throughput of the network, sometimes it is promising to devote network resources completely to some particular users. This leads to starvation. One of the goals in this thesis is to avoid starvation due to data rate allocation and transmission scheduling. As an appropriate quantitative measure of network performance in terms of fairness provisioning, we use Jain's fairness index [2].

Fairness provisioning in wireless networks has been considered [3–10]. Lin and Shroff considered the impact of imperfect scheduling on performance of the network [3]. Proportional fairness is provided in single hop wireless networks using token counter mechanisms [4]. Fairness provisioning is also studied using back-pressure combined with random access algorithms [5]. Fairness is provided in the cellular networks, where there is only one transmitter and all transmissions are one hop [6]. A distributed fair carrier sense multiple access (CSMA) algorithm is provided in [11] and its convergence under several practical assumptions is considered [12]. Fair power allocation for cooperative wireless networks is also considered [13].

1.1.2 Reliability

Unlike wired networks, links in wireless networks can be unreliable and prone to transmission error due to channel imperfections, background noise, environmental obstacles, weather conditions, and user mobility [14]. Unreliable links can degrade network performance particularly for applications with tight QoS requirements such as voice-over-IP and video streaming [15]. Therefore, it is crucial to develop efficient strategies in order to improve the reliability of data transmission in wireless networks [16].

The ability of a network to overcome failures can be characterized through three important measures [17]: First, the network ability to perform its designated tasks under certain conditions with guaranteed performance. Secondly, the ability of network to perform its tasks at any instance of time, and finally network's ability to survive some network failures. In this thesis, we improve the reliability of the network as a combination of the three measures above. To study the reliability problem in wireless networks, reliability can be modeled and analyzed as a utility function over the nodes participating in the network. QoS requirements and constraints need to be jointly considered while optimizing network behavior.

A good measure for reliability can be the fraction of packets that are successfully received by the receiver. Another measure may be the mean value of transmission efforts for each successful transmission [17, 18]. Markov process analysis is used to model and analyze the end-to-end data transmission in a wireless network, and the tradeoff between delay and reliability is investigated [19]. Data transmission reliability in wireless mesh networks is summarized [20]. Varshney *et al.* [21] considered reliability and survivability in infrastructure-oriented wireless networks and provided some reliability measures and modeling.

Different approaches are used to make wireless networks more reliable. They include

rate allocation [22, 23], channel coding [22], network coding [18, 24], and multipath routing [25–27]. Many rate allocation approaches are based on variations of the network utility maximization [28–32].

1.2 Techniques

1.2.1 Network Utility Maximization

Following the work by Kelly *et al.* [33], network utility maximization (NUM) has been widely used as a framework to systematically devise resource allocation strategies that can enhance the network performance subject to various capacity and QoS constraints [22, 34–36].

Most resource allocation problems can be formulated as NUM problems. The utility function represents an objective that is to be maximized and the constraints model the different underlying network characteristics. The NUM approach has been applied in different problems, including energy minimization [37], congestion control [38], joint throughput maximization and power control [39], joint congestion and power control [34], MAC [40, 41] joint congestion and MAC control [35], and cross-layer optimization [42]. Other rate allocation approaches are also considered. In [43], a rate control protocol with rate mismatch and queue size feedback has been considered and a control theoretic analysis of the system has been provided. However, rate allocation approaches do not automatically result in a packet scheduling and transmission policy.

1.2.2 Network Coding

One approach to increase the network capacity while improving the reliability of the network is network coding. Through network coding, each intermediate node combines the received codewords from different neighbor nodes and encodes them into new codewords. The combined code words are then sent to the next hop on the path. This continues until the destination node receives enough encoded codewords that it can decode them and obtain the original packets. Network coding can be performed by *jointly* encoding multiple packets either from the *same* source or from *different* sources. The former is called *intra-session* network coding [44] while the latter is called *inter-session* network coding [45].

Following the seminal paper by Ahlswede *et al.* [44], a rich body of work has focused on developing techniques to improve network performance using network coding. Reliability in intra-session network coding is studied by Lun *et al.* in [46]. Network coding has been shown as a promising approach in communication networking, particularly for maximizing capacity in wireless networks [47]. Other network coding design objectives include maximizing network lifetime [48], minimizing energy consumption [49], and maximizing the network throughput [50].

1.2.3 Channel Coding

Channel coding (cf. [22, 51]) is commonly used as a tool to leverage reliable transmissions over *lossy* wireless links. With channel coding, the transmitter node of each link encodes the transmitted packets by adding *auxiliary* or *redundant* bits, which can increase the distance among the codewords and decrease the *packet error probability*. The higher the number of redundant bits is, the lower is the probability that the packet is corrupted in the channel such that it cannot be decoded at the receiver. If the number of extra bits is the same across all links, then channel coding is *non-adaptive*. On the other hand, if we change the amount of redundant bits for each link based on its current state, then channel coding is *adaptive*. Adaptive channel coding may result in better performance compared to non-adaptive channel coding; however, it entails a higher complexity. In general, channel coding usually introduces a *tradeoff* between reliability and data transmission rate. In fact, by changing the *code rate*, i.e., the ratio of data bits to data plus redundant bits, we can change the data rate at which the information is transmitted over each wireless link. In particular, the code rate can be decreased in order to improve (reduce) the probability of error at the cost of having lower data rates. Similarly, we can increase the code rate to increase the transmission data rate, but at the cost of increasing the probability of error. Adaptive channel coding has been used in [22] to enhance the network reliability, when single-path routing is being used. The rate-reliability tradeoff introduced through channel coding is studied in [22, 36, 51, 52].

Channel coding schemes fall into two main types: block codes and convolutional codes. In convolutional codes, the encoded packets do not depend only to the latest data packets but also to the past encoded data. However, in block codes each codeword depends only to a number of latest data bits. Golay codes are an example for block codes.

1.2.4 Multipath Routing

Multipath routing can be used to compensate for the data rate reduction due to channel coding. This is done by distributing the load over multiple routing paths. Multipath routing can provide *fault tolerance* against link failures and also achieve *load balancing* in order to better utilize the available network capacity [53–55]. Multipath routing brings two important advantages. One is that data can be sent from the source node with higher data rate than single-path routing because multiple paths (more resources) are being used simultaneously between the source node and the destination node. This gives the opportunity of using the network resources more efficiently. Second, having multiple paths available, we can push higher data transmission rate to the paths with higher reliability instead of having to use vulnerable paths with high probability of failure. This improves the reliability of the network by adjusting the data rates on different routing paths according to the state of wireless channels and other data flows in the network. The improvements lead to reducing network congestion, increasing throughput, and also higher energy efficiency [56]. The impact of multipath routing on energy consumption is examined in [57]. The effectiveness of multiple paths in meeting delay constraints is also studied in [58].

Multipath routing has been studied in wired networks [59] and wireless networks [25, 60] scenarios. Many standard protocols have been introduced for multipath routing in wireless networks such as ad hoc on-demand distance vector with backup routing (AODV-BR) [61], multipath source routing (MSR) [62], multipath dynamic source routing (MP-DSR) [63], split multipath routing (SMR) [64], multipath ad hoc on-demand distance vector (MAODV) [65], and the work in [66].

1.2.5 Interference Alignment

Multiple-input multiple-output (MIMO) wireless systems are wireless networks in which transmitters and receivers may have more than one antenna. They offer the possibility of spatial multiplexing of data streams in addition to time and frequency multiplexing. Spatial multiplexing leads to capacity increase linear to $\min(M, N)$, where M and Nare the number of transmitter and receiver antennas, respectively [67]. The challenge in designing such MIMO systems, however, is in dealing with interference from concurrent signal transmissions. In such systems, wireless interference shows up among different sender antennas at each receiver. Interference alignment is among the several techniques which has been recently used to overcome this problem.

When the interference is strong, the interfering signal can be decoded as well as the desired signal [68]. This may be at the cost of degradation in the users' sending rates. On the other hand, in case the interference is very weak, it can be treated as noise. More recently, interference alignment [68] has been suggested for the cases when the strength of interference is comparable to the strength of the desired signal. Interference alignment techniques involve the use of suitable encoding and decoding matrices, at the transmitter and receiver respectively, such that at each receiver the interference caused by all undesired transmitters is projected onto a separate interference subspace. This permits the receiver to easily extract the desired signal from the corresponding interference-free subspace. Cadambe and Jafar have shown that while the per-user sum-rate for an K-user interference channel without interference alignment is $\frac{1}{K} \log(SNR) + O(SNR)$, where SNR is the signal-to-noise ratio, the sum-rate per-user can be increased to $\frac{1}{2} \log(SNR) + O(SNR)$ with interference alignment [68].

Most of the work in the field of interference alignment is related to determining the maximum possible degrees of freedom (DoF) as well as studying its feasibility and achievability through finding optimal encoding and decoding matrices in a (M, N) system with K users (where K is as small as 2, 3) such that the interference is minimum at undesired receivers.

The effectiveness of interference alignment in fully connected wireless networks with more than two users is considered [68, 69]. While global knowledge of channels is required in the work by Cadambe *et al.* [68], Gomadam *et al.* [70] provide a distributed approach using the reciprocity of wireless networks to make the local channel knowledge adequate. The achievability of high DoF in wireless networks when instantaneous channel state information is not available is studied [71].

The feasibility of interference alignment in MIMO systems has been considered in [72, 73] by relating the problem to the problem of determining the solvability of a multivariate polynomial system. In addition to interference alignment which aims at minimizing the interference strength projected outside the interference subspace, the work in [74] also minimizes the signal strength that is spilled inside the interference subspace. Interference alignment and cancellation in MIMO systems is shown in [75] to almost double the throughput of MIMO based wireless local area networks (WLANs). Interference alignment with the goal of achieving the maximum sum-rate has been considered in MIMO [76] and cognitive MIMO systems [77].

Interference alignment techniques have also been exploited for downlink cellular systems where channel state information exchange is not required across base stations of different cells [78]. Interference alignment has also been used to model and solve other problems such as network coding [79, 80].

1.3 Mathematical Background

In this section, we provide a brief discussion on mathematical background that is needed in this thesis.

1.3.1 Convex Optimization

Before presenting the standard form of convex optimization problems [81], we first introduce some related concepts. Set \mathcal{X} is convex if for any two points $x_i, x_j \in \mathcal{X}$, the line segment connecting x_i to x_j is in set \mathcal{X} as well. That is, we have $\theta x_i + (1 - \theta)x_j \in \mathcal{X}$ for $\theta \in [0, 1]$. Function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is convex if **dom** f is a convex set and we have $f(\theta x_i + (1 - \theta)x_j) \leq \theta f(x_i) + (1 - \theta)f(x_j)$ for $\theta \in [0, 1]$. A convex optimization problem has the following standard form.

minimize
$$f_0(\mathbf{x})$$

subject to $f_i(\mathbf{x}) \le 0, \quad i = 1, \dots, m,$
 $h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p,$ (1.1)

where $f_i(\mathbf{x})$, i = 0, ..., m, are convex functions, and $h_i(\mathbf{x}) = a_i^T \mathbf{x} - b_i$, i = 1, ..., p, are affine functions. Problem (1.1) addresses the problem of finding the optimization variable $x \in \mathbb{R}^n$ such that

- inequality constraints are satisfied. That is, $f_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is less than or equal to zero for all i = 1, ..., m,
- equality constraints are satisfied. That is, $a_i^T \mathbf{x} = b_i$, for all i = 1, ..., p, and
- the objective function $f_0(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ obtains its minimum value among all $\mathbf{x} \in \mathbb{R}^n$ that satisfy the above constraints.

A convex optimization problem can be solved through the dual method. The basic idea in the dual method is combining the weighted equality and inequality conditions with the objective function in a new function called *Lagrangian*. The Lagrangian of problem (1.1)is as follows.

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x}).$$
(1.2)

We refer to λ and ν as Lagrangian multipliers corresponding to inequality and equality constraints, respectively. We define the dual function $g(\lambda, \nu) : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as the infimum value of the Lagrangian function over $\mathbf{x} \in \mathcal{X}$, that is

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{X}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}).$$
(1.3)

11

Note that for any $\lambda \succeq 0, \nu, g(\lambda, \nu)$ yields a lower bound for the optimal value of problem (1.1). As opposed to *primal problem* (1.1), consider *Lagrangian dual problem* as follows.

maximize
$$g(\boldsymbol{\lambda}, \boldsymbol{\nu})$$

subject to $\boldsymbol{\lambda} \succeq \mathbf{0}.$ (1.4)

The optimal value of problem (1.4) yields the best lower bound for the optimal value of problem (1.1). Under the strong duality conditions, this lower bound equals to the optimal value of the primal problem. Note that for convex optimization problems, *Slater's condition* is enough for strong duality [81, p. 226]. Slater's condition requires the existence of at least one feasible point inside the feasible set of the primal problem.

1.3.2 Geometric Programming

Geometric programming (GP) is a type of non-convex optimization problems that can be converted to convex optimization problem and be solved accordingly. We first define a monomial function $m(\mathbf{x}) : \mathbb{R}_{++}^{\mathbf{n}} \to \mathbb{R}$ as $dx_1^{a(1)}x_2^{a(2)}\dots x_n^{a(n)}$, where d > 0, and $a(i) \in \mathbb{R}, i = 1, 2, \dots, n$. A posynomial is the summation of K monomials as in $p(\mathbf{x}) =$ $\sum_{k=1}^{K} d_k x_1^{a_k(1)} x_2^{a_k(2)} \dots x_n^{a_k(n)}$.

Now, we can introduce the standard form of GP problems [82] as

minimize
$$f_0(\mathbf{x})$$

subject to $f_i(\mathbf{x}) \le 1, \quad i = 1, \dots, m,$ (1.5)
 $h_i(\mathbf{x}) = 1, \quad i = 1, \dots, p,$

where $f_i(\mathbf{x}) = \sum_{k=1}^{K^i} d_k^i x_1^{a_k^i(1)} x_2^{a_k^i(2)} \dots x_n^{a_k^i(n)}$ are posynomials for any $i = 0, \dots, m$, and $h_i(\mathbf{x}) = d^i x_1^{a^i(1)} x_2^{a^i(2)} \dots x_n^{a^i(n)}$ are monomials for each $i = 1, \dots, p$. GP is not convex in its standard form. However, it can be converted to a convex problem by the logarithmic

change of its all variables. Having $y_i = \log x_i$, we have

minimize
$$\sum_{k=1}^{K^0} \exp(b_k^0 + a_k^0(1)y_1 + \ldots + a_k^0(n)y_n)$$

subject to
$$\sum_{k=1}^{K^i} \exp(b_k^i + a_k^i(1)y_1 + \ldots + a_k^i(n)y_n) \le 1, \quad i = 1, \ldots, m, \quad (1.6)$$
$$\exp(b^i + a^i(1)y_1 + \ldots + a^i(n)y_n) = 1, \quad i = 1, \ldots, p,$$

where $b_k^i = \log d_k^i$, for any $i = 1 \dots m$, $k = 1, \dots, K^i$. We also have $b^i = \log d^i$, for any $i = 1, \dots, p$. Considering that the logarithmic function is an increasing function, we can take the logarithm of the objective function as well as the constraints and obtain a convex optimization problem. This is due to the fact that the *log-sum-exp* function is convex [83]. Therefore a GP can easily be solved by using the techniques to tackle convex optimization problems.

1.3.3 Non-linear Mixed Integer Programming

Sometimes it is not possible to accept solutions in which some optimization variables are not integers. This happens for example when we are seeking for the optimal number of people working in a particular section or the optimal number of wireless links which are active in one particular time slot. Another important instance of this is in scheduling problems in which the scheduling variables must be equal to either 0 or 1, which corresponds to whether a link is off or on. Such problems must be formulated as integer programming problems. Note that in case there are both integer and non-integer variables in the problem, it is called a mixed-integer programming problem. The standard form of a mixed integer linear programming problem is as follows [84].

minimize
$$\sum_{n=1}^{N} c_n x_n$$

subject to $\sum_{n=1}^{N} a_{mn} x_n \leq b_m, \quad m = 1, \dots, M,$
 $x_n \geq 0, \qquad n = 1, \dots, N,$
 $x_i \text{ is integer}, \qquad i \in \mathcal{I},$

$$(1.7)$$

where \mathcal{I} is a subset of the set of variables $\{n = 1, \ldots, N\}$.

In contrast with linear programming, it is not straightforward to solve integer programming problems when they are not linear, that is there exist non-linear functions of optimization variables as the objective function and/or constraints. Some methods are suggested for solving non-linear (mixed) integer programming problems such as Generalized Benders Decomposition (GBD) and Outer Approximation (OA) approaches. In the next section, we introduce the GBD method.

1.3.4 Generalized Benders Decomposition (GBD)

Consider the following non-linear mixed integer programming problem.

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} & f(\mathbf{x},\mathbf{y}) \\ \text{subject to} & h_i(\mathbf{x},\mathbf{y}) = 0, \qquad i = 1, \dots, m, \\ & g_i(\mathbf{x},\mathbf{y}) \leq 0, \qquad i = 1, \dots, p, \\ & \mathbf{x} \in X \subseteq \mathbb{R}^n, \\ & \mathbf{y} \in Y = \{0,1\}^q, \end{array}$$

$$(1.8)$$

where X is a nonempty, and convex set, $h_i : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ for i = 1, ..., m, is a linear function for each fixed $\mathbf{y} \in Y$, and $f : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ and $g_i : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ for i = 1, ..., p, are convex functions for each fixed $\mathbf{y} \in Y$. We have also set $\mathcal{Z}_{\mathbf{y}} = \{\mathbf{z} = (z_1, ..., z_p) \in$ \mathbb{R}^p : $h_i(\mathbf{x}, \mathbf{y}) = 0, g_j(\mathbf{x}, \mathbf{y}) \le z_j, i = 1, \dots, m, j = 1, \dots, p$, for some $\mathbf{x} \in X$ } is a closed set. Moreover, for each feasible $\mathbf{y} \in Y$, we have either problem (1.8) has a finite solution and optimal Lagrangian multipliers vector, or problem (1.8) is unbounded.

The GBD method [84] is an iterative algorithm. At each iteration, we formulate two problems: a *primal problem*, which yields an upper bound on the optimal value, and a *master problem* that yields a lower bound on the optimal value. The primal problem is formulated by fixing the integer variables in some particular point and solving problem (1.8) in that point. Consider the following primal problem at the k^{th} iteration.

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}, \mathbf{y}^{\mathbf{k}}) \\ \text{subject to} & h_i(\mathbf{x}, \mathbf{y}^{\mathbf{k}}) = 0, \quad i = 1, \dots, m, \\ & g_i(\mathbf{x}, \mathbf{y}^{\mathbf{k}}) \le 0, \quad i = 1, \dots, p, \\ & \mathbf{x} \in X \subset \mathbb{R}^n, \end{array} \tag{1.9}$$

where $\mathbf{y}^{\mathbf{k}}$ is a feasible point in set Y. Clearly, since the integer variables are fixed, the optimal value of problem (1.8) is always equal or better (smaller) than the optimal value of problem (1.9) and therefore, the primal problem provides an upper bound for the optimal value. In case problem (1.9) is infeasible for $\mathbf{y} = \mathbf{y}^{\mathbf{k}}$, we formulate a feasibility problem. Consider the l_1 -minimization problem at iteration k.

$$\begin{array}{ll}
\underset{\mathbf{x}, \ \beta \succeq \mathbf{0}}{\text{minimize}} & \sum_{i=1}^{p} \beta_{i} \\
\text{subject to} & h_{i}(\mathbf{x}, \mathbf{y}^{\mathbf{k}}) = 0, \quad i = 1, \dots, m, \\
& g_{i}(\mathbf{x}, \mathbf{y}^{\mathbf{k}}) \leq \beta_{i}, \quad i = 1, \dots, p, \\
& \mathbf{x} \in X \subseteq \mathbb{R}^{n}.
\end{array}$$
(1.10)

Let \mathcal{M} and \mathcal{M}' denote the set of all iterations k such that the primal problems are feasible and infeasible, respectively. Using the solutions to the primal problems (1.9) and (1.10) in all previous iterations and the information on Lagrange multipliers, we formulate the master problem as follows:

$$\begin{array}{ll} \underset{\mu,y}{\text{minimize}} & \mu \\ \text{subject to} & \mu \geq L_1(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k), \quad \forall \ k \in \mathcal{M}, \\ & 0 \geq L_2(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k), \quad \forall \ k \in \mathcal{M}', \\ & \mathbf{y} \in Y = \{0, 1\}^q, \end{array}$$

$$(1.11)$$

where $L_1(\mathbf{x}, \mathbf{y}, \lambda, \nu) = f(\mathbf{x}, \mathbf{y}) + \lambda^T h(\mathbf{x}, \mathbf{y}) + \nu^T g(\mathbf{x}, \mathbf{y})$ and $L_2(\mathbf{x}, \mathbf{y}, \lambda, \nu) = \lambda^T h(\mathbf{x}, \mathbf{y}) + \nu^T g(\mathbf{x}, \mathbf{y})$. The GBD method works as follows. At the first iteration, the primal problem (1.9) is solved in an initial point. The initial point must be chosen such that the first primal problem is feasible. Then, at each iteration, the master problem (1.11) is formulated using the information obtained in the solution of the previous primal and feasibility problems and a lower bound is provided. Then, the primal problem (1.9) is formulated by fixing \mathbf{y} at the integer point obtained in the master problem and a new upper bound is obtained. In case the primal problem is infeasible, feasibility problem (1.10) is formulated instead. This continues until the upper bound and the lower bound obtained in the primal and master problems converge together.

1.4 Contributions and Results

In this thesis, we aim to improve performance of wireless mesh networks under reliability constraints. Our contribution is to develop rate allocation algorithms, scheduling policies, and flow control solutions with the goal of improving the reliable throughput of the network. While achieving that goal, we also consider fairness provisioning. In the following, we describe our results and contributions in summary.

- In Chapter 2, we consider *unreliability* of wireless networks due to varying channel conditions, especially when users are mobile. We formulate a joint max-min fair channel coding and end-to-end data rate allocation problem in multipath wireless networks. We maximize the minimum throughput available among the network users. To cope with the fast and frequent changes in dynamic environments such as wireless mesh networks in vehicular environments, we address both *adaptive* and *non-adaptive* channel coding scenarios. We also redesign the adaptive algorithm to use the available information related to the current optimal solution leading to a faster convergence. Unlike similar formulations in *single-path* routing networks, which involve solving either a *convex* optimization problem or problems that can be transformed into a convex optimization problem, in the multipath routing case we face an optimization problem that is *non-convex* and is usually difficult to solve. We tackle non-convexity by using *function approximation* and iterative techniques from signomial programming. Simulation results confirm that by using channel coding jointly with multipath routing, we can significantly improve end-to-end network performance, compared to the case when only one of them is used in the network. Non-adaptive channel coding is also shown to achieve high degree of optimality with much less complexity.
- In Chapter 3, we consider delay in wireless networks with multipath routing and channel coding. We show that a *combination* of multipath routing and adaptive channel coding can improve throughput and reduce delay, and that it is possible to trade off delay for throughput and vice versa. This is in contrast to the general expectation that higher throughput can only be achieved with noticeable degradations in the end-to-end network delay. In this regard, we jointly formulate the end-to-end data rate allocation and adaptive channel coding (at the physical layer) within the

general framework of network utility maximization (NUM). Depending on the choice of the objective function, we formulate *two* NUM problems: one aiming to maximize the aggregate network utility; another one aiming to maximize the minimum utility among the end-to-end flows in order to achieve fairness, which is of interest in certain vehicular network applications. Simulation results confirm that we can decrease the average delay significantly at the cost of a small decrease in throughput. This is achieved by maximizing the aggregate utility in the network when fairness is not the dominant concern. Furthermore, we also show that even when resource allocation is performed in order to provide fairness, we can still decrease the maximum end-to-end delay of the network at the cost of a slight decrease in the minimum throughput.

- In Chapter 4, we focus on *inter-session* network coding, where multiple unicast sessions jointly participate in network coding. We consider multi-hop unicast sessions over unreliable links and propose a distributed end-to-end transmission rate adjustment mechanism to maximize the *aggregate network utility* by taking into account the wireless link reliability information. This includes an elaborate modeling of *end-to-end reliability*. Simulation results show that by taking into account the reliability information, we can increase the network throughput by up to 100% for some network topologies. We can also increase the aggregate network utility significantly for various choices of utility functions.
- In Chapter 5, we consider the setting of multihop wireless networks and develop an online flow control and scheduling algorithm for packet admission and link activation that achieves high aggregate throughput while providing different data flows with a fair share of network capacity. For fairness provisioning, we maximize the minimum throughput provided to flows in the network. We also develop an algorithm that does not consider fairness provisioning. To cope with different degrees of data reliability
among the different links in the network, we use different channel code rates as appropriate. While we expect performance improvement using channel coding and multipath routing, the main contribution is a joint treatment of network stability, multipath routing and link-level reliability in meeting the overarching goal of maxmin fairness. We develop a decentralized, and hence practical, scheduling policy that addresses various concerns and demonstrate, via simulations, that it is competitive with respect to an optimal centralized rate allocator. We also evaluate the fairness provisioning under the proposed algorithm and show that channel coding improves the performance of the network significantly. Finally, we show through simulations that the proposed algorithm outperforms a class of existing approaches on fairness provisioning, which are developed based on utility maximization.

• In Chapter 6, we study scheduling, interference alignment, and packet admission control in MIMO wireless systems with the goal of maximizing system throughput subject to stability constraints. We formulate the joint scheduling, packet admission control, and interference alignment problem as a stochastic network optimization problem and propose an efficient algorithm. In each time slot, the algorithm schedules some users to send data among many competing ones. It also determines the optimal encoding and decoding matrices for those selected users. Packet admission control is performed in each time slot. We also propose an efficient heuristic. Via simulation, we evaluate the performance of both the proposed optimal algorithm and heuristic under different number of users and showed that the heuristic follows the optimal central solution with some degree of sub-optimality.

1.5 Thesis Organization

This thesis is organized as follows. We first develop an optimal data rate and channel code rate allocation algorithm in multipath routing wireless networks with channel coding in Chapter 2. The goal is to improve the reliability and throughput in the wireless mesh networks with fairness provisioning. In Chapter 3, we incorporate the end-to-end delay in the utility function of the problem solved in Chapter 2 and reformulate the problem of data and channel code rate allocation with the goal of decreasing the end-to-end delay for network flows. In Chapter 4, we use inter-session network coding for performance improvement and obtain the optimal data rate allocation for both routing paths and network coding schemes. To address the achievability of the rate allocation design obtained in Chapter 2, we consider transmission scheduling as well as flow control in a time slot based setting in Chapter 5. In Chapter 6, we study the problem of joint scheduling, interference alignment, and packet admission in MIMO systems. Finally, the thesis is concluded and some potential future work is introduced in Chapter 7. Each of the main chapters in this thesis is self-contained and included in separate journal articles or conference papers. A review of the related work is given for each chapter accordingly. The notations are defined separately for each chapter.

Chapter 2

Optimal Data Transmission and Channel Code Rate Allocation in Multipath Wireless Networks

2.1 Introduction

As discussed in Chapter 1, multipath routing and channel coding are promising techniques for performance improvement in wireless mesh networks. In this chapter, our focus is to *jointly* use channel coding and multipath routing in an optimization-based framework to further improve reliability compared to using *only* channel coding or *only* multipath routing. We are interested in answering the following question: *How shall we select the end-to-end data transmission rates over different paths and per-link channel code rates in order to achieve the optimal rate-reliability tradeoff in multipath wireless networks?*

Our main contribution is a solution to the fair resource allocation problem in networks that jointly employ multipath routing and link-specific channel coding. In this regard, our work is closely related to [22]. However, here we introduce three key extensions. First, Lee *et al.* in [22] assume that the links in the network are either *wired* or *interferencefree* wireless. On the contrary, here we have explicitly incorporated the impact of wireless interference. Second, unlike the system model in [22] which addresses only single-path routing, here we consider the case where there are multiple end-to-end routing paths available across the network. Clearly, this includes single-path routing as a special case. Third, we formulate the problem such that the minimum throughput among the individual users is maximized. This leads to fairness provisioning which is of great importance in certain applications such as vehicular networks where vehicles frequently switch among stationary mesh nodes to receive connectivity. In this case, different mesh nodes must be provided with fair and consistent data rates. The aforementioned three extensions introduce several challenges in solving the formulated optimization problem and have not been addressed before. Those are due to various *non-convexities* that cannot be directly transformed into a convex optimization problem using the well-known logarithmic change of variables as in [22]. Although our proposed method is centralized, it may be used in vehicular network applications such as those in which stationary access points provide connectivity for the vehicles in their coverage zone. Moreover, it can shed light on how per-link channel coding can improve end-to-end performance in a multipath routing wireless network. The centralized solution may also be used as a benchmark for evaluating distributed approaches which may be developed in the future. To the best of our knowledge, rate allocation with the goal of fairness and reliability enhancement using multipath routing and channel coding has not been addressed in any prior work.

The contributions of this chapter are as follows.

- We formulate max-min fair resource allocation in multipath wireless networks employing channel coding as an optimization problem. Our system model includes both *adaptive* and *non-adaptive* channel coding.
- We tackle the non-convexity of the formulated optimization problem in two steps. First, we use *function approximations* to reformulate the problem as a *signomial programming* problem (which is still non-convex). Next, we develop an iterative algo-

rithm to solve the signomial programming problem by solving a chain of tractable *geometric programming* problems. We introduce a non-adaptive channel coding scheme with much lower degree of complexity, which can find a sub-optimal solution. We design our algorithm such that it can quickly find the new solution, whenever there is a change in the network topology and the number of users.

- To motivate the joint use of multipath routing and channel coding, we show through simulations that our proposed scheme significantly improves the network performance when compared to the case with multipath routing, but *without* channel coding. We also show that our joint scheme outperforms channel coding in single-path routing systems.
- We investigate the convergence properties of the proposed algorithm as well as its efficiency. The latter is studied particularly by evaluating the impact of the approximations made in the derivation of the algorithm.
- We compare the *adaptive* coding scheme with the *non-adaptive* coding scheme with less computational complexity. We evaluate the proposed algorithm in a dynamic vehicular environment where the data traffic pattern changes due to mobility. Finally, we study the effects of fading on the performance of the algorithm.

The rest of this chapter is organized as follows. We present the system model and formulate the joint data rate and channel code rate allocation problem in Section 2.2. In Section 2.3, we reformulate the problem as a geometric programming problem and propose a reliability-based rate allocation algorithm to solve it. We also introduce a non-adaptive channel coding scheme as a sub-optimal solution with lower complexity. Simulation results are presented in Section 2.4. The chapter is summarized in Section 2.5.

2.2 System Model And Problem Formulation

2.2.1 System Model

Consider an ad-hoc wireless network. We can model the network topology as a directed graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ..., V\}$ is the set of nodes and \mathcal{E} is the set of wireless links. Let $\mathcal{I} = \{1, 2, ..., I\}$ denote the set of all unicast sessions in the network. For each session $i \in \mathcal{I}$, the source and destination nodes are denoted by s_i and t_i , respectively. Furthermore, we denote $\mathcal{K}_i = \{1, 2, ..., K_i\}$ as the set of all available routing paths from source node s_i to destination node t_i . For each session $i \in \mathcal{I}$, each link $e \in \mathcal{E}$, and each $k \in \mathcal{K}_i$, we have

$$a_i^{ek} = \begin{cases} 1, & \text{if link } e \text{ belongs to } k^{\text{th}} \text{ routing path for session } i, \\ 0, & \text{otherwise.} \end{cases}$$
(2.1)

We assume that static routing is used and the routing information is given a priori.

For each session $i \in \mathcal{I}$, let α_i^k denote the data rate of source s_i on its k^{th} routing path, $k \in \mathcal{K}_i$. The aggregate *transmission* rate for session *i* is obtained as

$$\sum_{k \in \mathcal{K}_i} \alpha_i^k. \tag{2.2}$$

Since the packets are retransmitted whenever they are lost in the network, the effective *receiving* rate at destination node t_i is the same as (2.2).

Channel coding can improve reliability on lossy channels by adding redundant bits to the data packets transmitted. In this regard, we define R_e as the *code rate* of link $e \in \mathcal{E}$, i.e., the ratio of the data bits to data plus redundant bits. Notice that if no channel coding is performed, then $R_e = 1$ as there will be no redundant bits in the packet. Let $R_{0e} \leq 1$ denote the *cut-off rate* on wireless link $e \in \mathcal{E}$. We assume that the rate R_e of the adopted coding schemes (e.g., convolutional codes) is limited by the cutoff rate [85]. Given code rate $R_e \leq R_{0e}$ if random coding based on *M*-ary binary coded signals is used, we can bound the packet error probability on link e to be less than $2^{-T(R_{0e}-R_e)}$ as [22, 36, 52, 85]. Therefore, in the worst case, we have

$$P_e = 1 - 2^{-T(R_{0e} - R_e)}, (2.3)$$

where P_e is the successful packet transmission probability on link e and T is the coding block length. In general, the cut-off rate R_{0e} depends on the signal-to-noise ratio (SNR) and the modulation scheme being used. For example, for a *binary phase shift keying* (BPSK) waveform [85], we have

$$R_{0e} = 1 - \log_2(1 + e^{-\gamma_e}), \tag{2.4}$$

where γ_e denotes the SNR at the receiver node of wireless link $e \in \mathcal{E}$. In particular, we have

$$\gamma_e = \Gamma_e \times d_e^{-\sigma} \times |f_e|^2, \quad \forall \ e \in \mathcal{E},$$
(2.5)

where Γ_e depends only on the SNR at transmitter, d_e is the distance between the transmitter and receiver of link e, σ is the path loss exponent (e.g., between 2 and 5), and f_e is the small-scale fading gain. Assuming re-transmission after a packet is lost in the network until reaching a successful transmission, each packet must be sent $1/P_e$ times on average over each link e. Given the source transmission rates $\boldsymbol{\alpha} = (\alpha_i^k, \forall i \in \mathcal{I}, k \in \mathcal{K}_i)$, successful transmission probabilities $\boldsymbol{P} = (P_e, \forall e \in \mathcal{E})$, and the link code rates $\boldsymbol{R} = (R_e, \forall e \in \mathcal{E})$, we can model the aggregate traffic load on link $e \in \mathcal{E}$ as

$$u_e = \frac{1}{R_e P_e} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} a_i^{ek} \alpha_i^k.$$
(2.6)

25

From (2.6), the smaller the code rate R_e , the more redundant data is added to the transmitted packets on link $e \in \mathcal{E}$ leading to more reliable transmission (i.e., transmission with lower error probability). However, this will be at the cost of increasing the traffic load on the link.

We can model the mutual interference among the wireless links in a network by using a contention graph $G_C(\mathcal{V}_C, \mathcal{E}_C)$. In the contention graph G_C , the set of vertices \mathcal{V}_C represents the set of all wireless links \mathcal{E} in the network graph G. There exists an edge between any two vertices in set \mathcal{V}_C if wireless links corresponding to two vertices mutually interfere with each other (i.e., the receiver node of one link is within the interference range of the sender node of the other link). Given the contention graph, each complete subgraph (i.e., a subgraph in which all vertices are connected to all other vertices) is called a *clique*. A maximal clique is then defined as a clique which is not a subgraph of any other clique [86]. Denote the set of all maximal cliques in contention graph G_C by \mathcal{Q}_C . Only one link among all the links corresponding to the vertices of a maximal clique $Q \in \mathcal{Q}_C$ can be active at a time.

For the data link layer, we assume that time division multiple access (TDMA) is used. Let c_e denote the *nominal* data rate of link $e \in \mathcal{E}$. The ratio $\frac{u_e}{c_e}$ denotes the proportion of time at which link $e \in \mathcal{E}$ is active when it is used at data rate c_e . It is required that

$$\sum_{e \in Q} \frac{u_e}{c_e} \le \nu, \qquad \forall \ Q \in \mathcal{Q}_C, \tag{2.7}$$

where $\nu \in (0, 1]$ is called the *clique capacity*. Note that if $\nu = 1$, then (2.7) is only a necessary constraint. It is shown that inequality (2.7) is a sufficient constraint when $\nu = 2/3$ [87]. Note that we assume all clique information given. This can be achieved in a central manner having the complete information about the network topology. Distributed approaches can also be developed which are out of scope of this thesis.

	<u>(</u>	(g)) A		(P)	(⁴))		
	(^e)	(g)) 		^e n	(^g)		<i>"</i> [
	(^p)	(^g))			۲. ۲		
						Щ	
ЦĮ		(^r)		(^g))	<i>¶</i> "		Ĩ
				Ш		Щ	
Ľ	<u></u>	(jr) A		<u>(</u> ?)	<u>(</u> "		Ĵ.
			L				

Figure 2.1: An example downtown area with 25 access points (forming a wireless mesh infrastructure). The access point at the center serves as the gateway. There are 10 vehicles in the system, each one uses the nearest access point to connect to the Internet.

Now we show how the provided modeling covers the vehicular environment. Consider Fig. 2.1 in which an example downtown area is shown. There is an access point in every other cross section and the one at the center of the area is denoted as the gateway. Access points correspond to nodes in set \mathcal{V} . There is a wireless link $e \in \mathcal{E}$ between two adjacent access points. Vehicles move in the streets continuously. Each vehicle at each instant of time finds the nearest access point and connects to it to transmit data to the gateway. The access point $i \in \mathcal{I}$ which is connected to a vehicle, represents source node s_i for flow i. The gateway corresponds to destination node t_i . During the time that vehicles move in the area the set of sources and thus the data traffic pattern changes.

2.2.2 Problem Formulation

Considering (2.2), (2.3), (2.6), and (2.7), the rate-reliability tradeoff can be explained as follows. For each link $e \in \mathcal{E}$, by increasing the code rate R_e we can reduce the traffic load per transmission on each link. Thus, higher transmission rates will be allowed with the same clique capacity. However, this is at the cost of less reliability and leads to more re-transmission attempts as in (2.6). On the other hand, by decreasing the code rate R_e , we can reduce the error probability in (2.3) which leads to higher probability of successful transmission along each routing path. Therefore, we may select either higher transmission rates, but with more packets being prone to error, or lower transmission rates, but with higher percentage of correctly received packets. In this regard, the key question to be answered is: What transmission rates α and code rates \mathbf{R} should be selected to achieve optimal performance?

To answer the above question, we formulate the following optimization problem.

Max-Min Fairness Problem:

$$\begin{array}{ll} \underset{\boldsymbol{\alpha} \succeq \mathbf{0}, \ \mathbf{0} \prec \boldsymbol{R} \preceq \boldsymbol{R}_{\mathbf{0}}}{\text{maximize}} & \underset{i \in \mathcal{I}}{\text{minimum}} & \sum_{k \in \mathcal{K}_{i}} \alpha_{i}^{k} \\ \text{subject to} & \sum_{e \in Q} \frac{1}{P_{e} R_{e} c_{e}} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_{i}} a_{i}^{ek} \alpha_{i}^{k} \leq \nu, \quad \forall \ Q \in \mathcal{Q}, \\ P_{e} = 1 - 2^{-T(R_{0e} - R_{e})}, \qquad \forall \ e \in \mathcal{E}, \end{array}$$

$$(2.8)$$

where $\mathbf{R}_{\mathbf{0}} = (R_{0e}, \forall e \in \mathcal{E})$ denotes the vector of cut-off rates for all links in the network. The objective function in (2.8) is the *minimum* receiving rate among all sessions in the network, where for each session $i \in \mathcal{I}$, the receiving rate is as in (2.2). By solving (2.8), we can find $\boldsymbol{\alpha}$ and \boldsymbol{R} such that the minimum throughput across all sessions is maximized. Notice that we could also maximize the aggregate network throughput. However, the aggregate network throughput maximization problem does *not* take into account any notion of fairness as the objective is to maximize the *total* network throughput. As a result, the optimal solution may lead to starvation in some sessions. Max-min fairness solution avoids starving any of the sessions and balances the performance in the network. We will discuss solving problem (2.8) in Section 2.3.

2.3 Optimal Transmission Rate and Channel Code Rate Allocation

2.3.1 Adaptive Channel Coding

In this section, we propose an *iterative* algorithm to solve the max-min fairness optimization problem to achieve *optimal* allocation of source transmission rates α as well as *optimal* channel code rates \mathbf{R} in the network. In general, problem (2.8) is *non-convex* and difficult to solve. Note that the non-convexities in problem (2.8) come from the following three sources: (a) The *minimum* term in the objective function. (b) The *exponential* forms in the equality constraints with respect to error probabilities. (c) The *fractional* forms in the inequality constraints with respect to clique capacities.

Most of these challenges are caused by the fact that, unlike many of the existing related work in the literature on rate-reliability tradeoff (e.g., in [22]), we take into account multipath routing and wireless interference. For example, if the network is wired such that no interference occurs among transmissions, then the clique capacity constraints would reduce to several *linear* link capacity constraints such that for each link $e \in \mathcal{E}$ we have

$$\frac{1}{P_e R_e c_e} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} a_i^{ek} \alpha_i^k \le 1 \Rightarrow \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} a_i^{ek} \alpha_i^k \le P_e R_e c_e.$$
(2.9)

However, these techniques are not applicable where multipath routing is used and wireless

transmissions incur interference. In fact, we need to go through more elaborate steps in order to be able to solve problem (2.8) in the general case as will be explained in detail next.

Recall that problem (2.8) is a non-convex optimization problem due to the three reasons listed earlier, where one of them is the exponential forms in the equality constraints with respect to error probabilities. We start by tackling this source of non-convexity. First, we replace this equality with an inequality. This does not degrade the performance of the algorithm because it overestimates the unreliability in the network. For notational simplicity, we rewrite the error probability (2.3) as

$$P_e \le 1 - X_e \, \exp\left(L_e \, R_e\right), \quad \forall \, e \in \mathcal{E}, \tag{2.10}$$

where $X_e = 2^{-TR_{0e}}$, and $L_e = T \ln 2$.

Recall that for each link $e \in \mathcal{E}$, we have $0 < R_e \leq R_{0e}$. We use *Taylor series expansion* to write inequality (2.10) as

$$P_e \le 1 - X_e \sum_{n=0}^{\infty} \frac{(L_e R_e)^n}{n!}, \quad \forall e \in \mathcal{E}.$$
(2.11)

Clearly, for some *bounded* integer $N_e \gg 1$, we have

$$P_e \le 1 - X_e \sum_{n=0}^{N_e - 1} \frac{(L_e R_e)^n}{n!}, \quad \forall e \in \mathcal{E}.$$
(2.12)

Unlike the *exponential* error probability model in (2.10), the model in (2.12) is in *polynomial* form. For (2.12) to approximate (2.11) accurately, we need N_e to be large enough such that $(L_e R_e)^{N_e} \ll N_e!$. We investigate the value of N_e necessary for obtaining a good approximation in Section 2.4.4.

By exploiting the *worst-case* packet error probability (2.12) in problem (2.8), we rewrite the max-min fairness problem as

$$\begin{array}{ll}
\begin{array}{l} \underset{\boldsymbol{\alpha} \succ \mathbf{0}, \ \mathbf{0} \prec \mathbf{R} \preceq \mathbf{R}_{\mathbf{0}}, \mathbf{P} \succ \mathbf{0}}{\text{maximize}} & \underset{i \in \mathcal{I}}{\text{minimum}} & \sum_{k \in \mathcal{K}_{i}} \alpha_{i}^{k} \\ \text{subject to} & \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_{i}} a_{i}^{ek} \ \alpha_{i}^{k} \ P_{e}^{-1} \ R_{e}^{-1} \ c_{e}^{-1} \leq \nu, \quad \forall \ Q \in \mathcal{Q}, \\ & \frac{P_{e}}{1 - X_{e}} + \frac{X_{e}}{1 - X_{e}} \sum_{n=1}^{N_{e}-1} \frac{(L_{e}R_{e})^{n}}{n!} \leq 1, \quad \forall \ e \in \mathcal{E}. \end{array} \right. \tag{2.13}$$

The objective in (2.13) is to maximize the utility of the transmission session with the minimum value. We can replace the minimum function in the objective function by introducing a new *auxiliary* variable t and a set of new constraints as

$$\begin{array}{l} \underset{t>0, \ \boldsymbol{\alpha} \succ \mathbf{0}, \ \mathbf{0} \prec \mathbf{R} \leq \mathbf{R}_{0}, \mathbf{P} \succ \mathbf{0}}{\text{minimize}} t^{-1} \\ \text{subject to} \quad t \leq \sum_{k \in \mathcal{K}_{i}} \alpha_{i}^{k}, \qquad \forall i \in \mathcal{I}, \\ \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_{i}} a_{i}^{ek} \alpha_{i}^{k} P_{e}^{-1} R_{e}^{-1} c_{e}^{-1} \leq \nu, \quad \forall Q \in \mathcal{Q}, \\ \frac{P_{e}}{1-X_{e}} + \frac{X_{e}}{1-X_{e}} \sum_{n=1}^{N_{e}-1} \frac{(L_{e}R_{e})^{n}}{n!} \leq 1, \quad \forall e \in \mathcal{E}. \end{array}$$

$$(2.14)$$

The objective function and constraints in problem (2.14) are *signomials*, i.e., polynomials with *both* positive and negative terms. Therefore, we can apply *signomial programming* techniques [82] to solve problem (2.14).

Consider the first constraint in (2.14). We follow the signomial programming techniques [82] to approximate the polynomial on the right-hand side of this inequality, which is a function of only α , as a *monomial*, i.e., a polynomial with only *one* term and *positive* multiplier. This approximation can be performed around some initial point $\hat{\alpha}$. For a

parameter $f_s > 1$, which is close to 1, we have

$$\sum_{k \in \mathcal{K}_i} \alpha_i^k \approx \left(\sum_{k \in \mathcal{K}_i} \hat{\alpha}_i^k \right) \prod_{k \in \mathcal{K}_i} \left(\frac{\alpha_i^k}{\hat{\alpha}_i^k} \right)^{\hat{\alpha}_i^k / \left(\sum_{k' \in \mathcal{K}_i} \hat{\alpha}_{i'}^{k'} \right)}, \quad \forall \ \boldsymbol{\alpha} \in \left[\hat{\boldsymbol{\alpha}} / f_s, f_s \hat{\boldsymbol{\alpha}} \right],$$
(2.15)

for any $i \in \mathcal{I}$, where $[\hat{\alpha}/f_s, f_s \hat{\alpha}]$ is a small neighborhood around initial point $\hat{\alpha}$. The closer f_s is to 1, the more accurate the approximation of (2.15) will be at the cost of slower convergence of the algorithm. For simplicity of notation, for any $i \in \mathcal{I}$, we define $\hat{\Lambda}_i$ which only depends on the initial point $\hat{\alpha}$, as

$$\hat{\Lambda}_i^{-1} = \sum_{k \in \mathcal{K}_i} \hat{\alpha}_i^k.$$
(2.16)

From (2.15) and (2.16), the first constraint can be approximated around the initial point $\hat{\alpha}$ as

$$\hat{\Lambda}_{i} t \prod_{k \in \mathcal{K}_{i}} \left(\frac{\alpha_{i}^{k}}{\hat{\alpha}_{i}^{k}} \right)^{-\hat{\alpha}_{i}^{k} \hat{\Lambda}_{i}} \leq 1, \qquad \forall i \in \mathcal{I}.$$
(2.17)

The above constraint is a *posynomial*, i.e., a polynomial with only *positive* terms. Replacing the first constraint in (2.14) with (2.17), the max-min fairness problem becomes

$$\begin{array}{l} \underset{t>0, \ \hat{\alpha}/f_{s} \leq \boldsymbol{\alpha} \leq f_{s} \hat{\alpha}, \ \mathbf{0} \prec \mathbf{R} \leq \mathbf{R}_{\mathbf{0}}, \ \mathbf{P} \succ \mathbf{0}}{\text{minimize}} t^{-1} \\ \text{subject to} \quad \hat{\Lambda}_{i} t \prod_{k \in \mathcal{K}_{i}} \left(\frac{\alpha_{i}^{k}}{\hat{\alpha}_{i}^{k}}\right)^{-\hat{\alpha}_{i}^{k} \hat{\Lambda}_{i}} \leq 1, \qquad \forall \ i \in \mathcal{I}, \\ \quad \frac{1}{\nu} \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_{i}} a_{i}^{ek} \ \alpha_{i}^{k} \ R_{e}^{-1} \ c_{e}^{-1} \leq 1, \qquad \forall \ Q \in \mathcal{Q}, \\ \quad \frac{P_{e}}{1-X_{e}} + \frac{X_{e}}{1-X_{e}} \sum_{n=1}^{N_{e}-1} \frac{(L_{e}R_{e})^{n}}{n!} \leq 1, \quad \forall \ e \in \mathcal{E}. \end{array} \right.$$

$$(2.18)$$

The above problem is a *geometric program*, which can be converted into a *convex* problem (cf. [82, 83]). Thus, problem (2.18) is a tractable optimization problem that can be solved

efficiently using *convex programming* techniques such as the *interior point method* [81]. We can solve the signomial programming problem (2.14) by iteratively solving (2.18).

We now present Algorithm 2.1 to solve the max-min fairness problem in (2.8). Algorithm 2.1 starts by initializing various system parameters. The initial end-to-end transmission rates $\hat{\alpha}$ are selected such that problem (2.18) is feasible. Several iterations are performed, where in each iteration, we solve the geometric programming problem (2.18) in Line 5 by using the interior point method [81]. Given the optimal transmission rates α_{opt} in each iteration, we update parameters $\hat{\Lambda}_i$ for any $i \in \mathcal{I}$ according to (2.16) and correspondingly reformulate problem (2.18) to be solved again in the next iteration. The iterations continue until the optimal objective value t_{opt} which is obtained in the current iteration does *not* change compared to the optimal objective value t_{old} in the previous iteration. The convergence of the algorithm in each iteration is guaranteed since the interior point method is used [88]. The convergence of Algorithm 2.1 is also guaranteed [82, p. 115].

Algorithm 2.1 : Algorithm to solve max-min fair resource allocation problem (2.8).

- 1: Initialize f_s , T, N_e , R_{0e} , X_e , L_e , c_e , ν , and $\hat{\alpha}_i^k$ for each $e \in \mathcal{E}$, $i \in \mathcal{I}$, and $k \in \mathcal{K}_i$.
- 2: Set $t_{opt} := -\infty$; $\epsilon := 10^{-5}$.
- 3: repeat
- 4: $t_{old} := t_{opt}$.
- 5: Solve problem (2.18) to obtain α_{opt} , R_{opt} , and t_{opt} .
- 6: Update $\hat{\boldsymbol{\alpha}} := \boldsymbol{\alpha}_{opt}$ and update Λ_i as in (2.16) for each $i \in \mathcal{I}$.
- 7: **until** $|t_{opt} t_{old}| \leq \epsilon$.
- 8: Optimal end-to-end data rates := α_{opt} .
- 9: Optimal per-link code rates := \mathbf{R}_{opt} .

In case any change happens in the network (e.g., change in the network topology, the number of network users or the traffic pattern), the input parameters of the formulated problem are updated and the corresponding new solution is obtained. Clearly, this can be time consuming if the changes are very frequent. To cope with frequent changes in dynamic environments, we modify the proposed algorithm such that it updates the last end-to-end data rate vector $\boldsymbol{\alpha}_{opt}$ to obtain the new initial point $\hat{\boldsymbol{\alpha}}$ for the new problem. This improves the convergence speed of the algorithm compared to the case where the existing solution is ignored and the problem is solved from scratch. The update process is to remove the entries for the users who left the network and also to add new entries for the new users who have just joined the network. The new entries must be chosen such that the problem remains feasible (i.e., small values must be chosen). In case of topology changes, the algorithm finds new routing paths and updates $\hat{\boldsymbol{\alpha}}$ accordingly. As mentioned before, the algorithm is executed in a central node (e.g., the gateway) and the required information (e.g., channel state information, location of users) is transferred through control messages. In case of the example vehicular network in Fig. 2.1, the problem is reformulated and solved in specific time instants in the gateway and the solution is passed on to the access points through control messages.

We note that Algorithm 2.1 needs to be used to update the code rates as well as end-to-end data rates whenever new channel measurements are available, particularly in a fading or mobile environment. We will investigate the impact of our design in a fast fading environment in Section 2.4.8. Moreover, we will discuss non-adaptive channel coding in Section 2.3.2 for the case when parameters change faster than the time required for the algorithm to converge.

2.3.2 Non-adaptive Channel Coding

In this section, we simplify the system model in Section 2.3.1 and assume that the channel code rate is *fixed* and is no longer an optimization variable in our design. That is,

$$R_e = R, \quad \forall e \in \mathcal{E}. \tag{2.19}$$

The impact of such an assumption is two-fold. First, it can simplify the clique capacity constraints in problem (2.8) as for each maximal clique $Q \in \mathcal{Q}$, we have

$$\sum_{e \in Q} \frac{1}{R_e P_e c_e} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} a_i^{ek} \alpha_i^k = \frac{1}{RP} \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \frac{a_i^{ek}}{c_e} \alpha_i^k \le \nu$$

$$\Rightarrow \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \frac{a_i^{ek}}{c_e} \alpha_i^k \le R P \nu,$$
(2.20)

which is simply a *linear* inequality constraint. Second, since we are adding the extra equality constraints into problem (2.8), any solution we achieve would be *sub-optimal*. In the non-adaptive channel coding case, the max-min fair resource allocation problem (2.8) is reformulated as

$$\begin{array}{ll} \underset{\alpha \succ 0}{\operatorname{maximize}} & \underset{i \in \mathcal{I}}{\operatorname{minimum}} & \sum_{k \in \mathcal{K}_i} \alpha_i^k \\ \text{subject to} & \sum_{e \in \mathcal{Q}} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \frac{a_i^{ek}}{c_e} \, \alpha_i^k \le RP\nu, \quad \forall \ Q \in \mathcal{Q}, \end{array}$$

$$(2.21)$$

where $P = 2^{T(R-R_0)}$. By introducing an auxiliary variable t and considering the worst case for error probabilities, problem (2.21) becomes

$$\begin{array}{ll} \underset{\boldsymbol{\alpha} \succ \mathbf{0}, t}{\operatorname{maximize}} & t \\ \text{subject to} & t \leq \sum_{k \in \mathcal{K}_i} \alpha_i^k, & \forall i \in \mathcal{I}, \\ & \sum_{e \in \mathcal{Q}} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \frac{a_i^{ek}}{c_e} \, \alpha_i^k \leq RP\nu, & \forall Q \in \mathcal{Q}, \end{array}$$

$$(2.22)$$

which is a *linear programming* problem. To find the best fixed code rate, we can solve problem (2.22) for different values of $R \in [0, R_0]$ and choose the solution with the highest objective value. With non-adaptive channel coding, we significantly decrease the computational complexity of solving the problem at some cost in performance. This can particularly help in dynamic environments where there are frequent changes in the system parameters.

2.4 Performance Evaluation

In this section, we assess the performance of our proposed joint channel coding and transmission data rate allocation algorithm (Algorithm 2.1). In our simulation model, we consider network topologies where $V = |\mathcal{V}| = m(m-1)$ wireless nodes are positioned on an $m \times m$ square grid with randomly selected grid locations. As an example, for the network in Fig. 2.2, we have m = 5 and V = 20. The network includes m source and destination pairs, with potentially many available routing paths from the source node to the destination node. In Fig. 2.2, there are *four* available routing paths from source node 1 to destination node 16. They include: $\{(1,2), (2,3), (3,8), (8,11), (11,16)\}, \{(1,2), (2,7), (7,8), (8,11), (11,16)\}, \{(1,6), (6,7), (7,8), (8,11), (11,16)\}, and <math>\{(1,6), (6,10), (10,14), (14,15), (15,16)\}$. Unless stated otherwise, the rest of the system parameters are selected as follows: $T = 10, N_e = 15, f_s = 1.1, R_{0e} = 1, \nu = \frac{2}{3}$ [87].

Without loss of generality, we choose the link capacity, c_e for each link $e \in \mathcal{E}$, to be equal to 1. Therefore, the transmission data rates, $\boldsymbol{\alpha}$, obtained in the optimal point can be interpreted as the vector of *normalized* transmission rates. If the algorithm is being executed for the first time, we set the initial data rates to be *small*, i.e., $\hat{\alpha}_i^k = 0.01$ for all $i \in \mathcal{I}$ and any $k = 1, \ldots, K_i$, in order to guarantee a feasible starting point for Algorithm 2.1, as we already discussed in Section 2.3.1. Otherwise, in case of updating the current rate vector, we set the new entries for the new routing paths equal to 0.01. To solve the geometric programming problems, we use MOSEK [89].



Figure 2.2: A sample network topology with 20 nodes randomly located in a 5×5 grid. The network includes five sessions: $1 \rightarrow 16$, $3 \rightarrow 13$, $2 \rightarrow 8$, $14 \rightarrow 17$, and $6 \rightarrow 20$. There are 4, 2, 2, 1, and 3 routing paths available for these sessions, respectively.

2.4.1 Multipath vs. Single-path Routing

We first study the performance enhancement achieved by using multipath routing compared to single-path routing. In the latter case, each source only uses one (out of possibly several) of the available shortest paths to its corresponding destination. We compare our proposed algorithm with the one in [22], where both channel coding and transmission rate allocation is performed in a single-path routing system.

By solving the max-min fair resource allocation problem (2.8) for the single-path routing (as in [22]) and also for the multipath routing cases (as in our proposed design), the optimal end-to-end data rates are obtained. Recall that the objective value in problem (2.8) is the *minimum* throughput among all five sessions. In Fig. 2.3, each point represents the averaged performance gain over 50 random topologies. We can see that the performance gain (i.e., the ratio of the averaged performance under multipath routing to the averaged performance under single-path routing) directly depends on the number of available (and



Figure 2.3: Comparison between the performance of adaptive channel coding in single-path and multipath routing networks in terms of the achieved normalized minimum throughout.

used) routing paths. It monotonically increases as the number of available routing paths increases. This increase is due to the availability of additional paths, the algorithm can distribute the load to the paths which experience less interference. Therefore, the sending rates are increased. In this case, the minimum network throughput can be enhanced by 22% on average when the average number of paths for each session is *only two*. This enhancement increases to 40% when the average number of routing paths increases to three. This is because the algorithm can inject the packets into the paths experiencing less interference.

2.4.2 Channel Coding vs. No Channel Coding

Next, we study how channel coding can improve the achieved network throughput in a *multipath* routing system. Since equality (2.3) models the worst case condition (i.e., provides upper bound on the error probability) and the error probability is equal to 1 in the absence of channel coding, we use the following exact successful packet transmission probability model for BPSK modulation for the case without channel coding:

$$P_e = \left(1 - Q(\sqrt{2\gamma_e})\right)^T,\tag{2.23}$$

where Q(.) denotes the Gaussian Q-function:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{u^2}{2}) du,$$
 (2.24)

and γ_e denotes the SNR at the receiver node of wireless link $e \in \mathcal{E}$. For a received SNR equal to 3 dB, we have $P_e = 0.4$ for T = 40. Our comparison reveals the *minimum* achievable performance gain by the use of channel coding. This is because we use the exact P_e for the case without channel coding but a lower bound (worst case) for the case with channel coding. As shown in Fig. 2.4, a major performance gain can be achieved with channel coding. The achieved performance degrades in both cases when the size of the network increases. This is because as the number of users increases, the interference in the network increases. For the results in Fig. 2.4, each point represents the normalized throughput averaged over 50 randomly generated network topologies.

2.4.3 Convergence Properties of Algorithm 2.1

Recall that each iteration of Algorithm 2.1 includes a function approximation step and a geometric programming step. Considering the network topology in Fig. 2.2, the convergence of the objective value for problem (2.8), when Algorithm 2.1 is used, is shown in Fig. 2.5. The objective value for problem (2.8) is the minimum throughput among all sessions. From the results in Fig. 2.5, Algorithm 2.1 converges after around 50 iterations. Similar results can be obtained for other network topologies.



Figure 2.4: Comparison between the performance of multipath routing *with* and *without* per-link channel coding in terms of the achieved normalized minimum throughput among the end-to-end sessions, when the scale of the network increases and the number of nodes varies from 6 to 42.



Figure 2.5: Convergence of Algorithm 2.1 with respect to solving problem (2.8). We can see that the algorithm converges after 50 iterations.



Figure 2.6: The impact of the choice of design parameter N_e in approximation (2.12). The average optimality error decreases as N_e increases. It becomes almost zero for $N_e > 12$.

2.4.4 Impact of Various Design and System Parameters

Parameter N_e

In Section 2.3, we use the approximation in (2.12) to convert problem (2.8) into a tractable geometric programming problem as in (2.13). We can improve the accuracy of the approximation in (2.12) by increasing the value of N_e . However, this would be at the cost of making problem (2.13) more complicated to solve. In this section we are interested in choosing N_e to obtain a reasonable accuracy with low computational complexity. Considering 50 random topologies, the simulation results, when N_e varies from 1 to 20, are shown in Fig. 2.6, where each point indicates the average *optimality error* observed for all 50 topologies. By obtaining the difference between the achieved network throughput at a particular choice of N_e and that at $N_e = 20$ (as the optimal throughput) and computing the ratio of this difference to the optimal throughput, we can define a measure for assessing the *optimality error*. Fig. 2.6 shows that the optimality error approaches zero when N_e is around 12 or higher.

Parameter f_s

Another approximation in Section 2.3 is the monomial approximation in (2.15). The approximation is made at each iteration within a close neighborhood of initial point $\hat{\alpha}$. The size of the neighborhood is denoted by design parameter f_s . In general, although we can increase the speed of convergence by increasing the value of f_s , it would be at the cost of a lower accuracy in the approximation. Considering such a tradeoff and based on our simulation results, we select $f_s = 1.1$, for a relatively good performance in terms of approximation accuracy, with a fast convergence speed.

Parameter T

In general, when we increase the coding block length T for a given code rate, the probability of error decreases. This can be seen in (2.3). By increasing T, one can allocate a higher code rate to a wireless link, while achieving the same probability of error, i.e., the same reliability measure. On the other hand, the more reliable links let the algorithm allocate higher end-to-end data rates, leading to improved optimal objective values in problem (2.8). This is shown for three random network topologies in Fig. 2.7, where the coding block length T varies from 10 to 100. The minimum throughput in the network increases in all three topologies when the coding block length (and thus the reliability) increases.

2.4.5 Adaptive vs. Non-adaptive Channel Coding

In this section, we show how choosing the code rate for each link individually (i.e., adaptive channel coding) can lead to different optimality and computational complexity results, compared to the case when channel coding is non-adaptive. Recall from Section 2.3.2 that in a non-adaptive channel coding scenario, we assume that all wireless links use the *same* code rate R as expressed in (2.19). In this case, for each fixed R, problem (2.8) becomes a



Figure 2.7: The impact of choosing different coding block lengths T on the network performance for three different random topologies. We observe that the performance improves when the coding block length increases.



Figure 2.8: Comparison between adaptive and non-adaptive channel coding.



Figure 2.9: Optimal non-adaptive code rate versus adaptive code rate distribution among all wireless links of the network topology in Fig. 2.2.

linear programming problem. This can significantly reduce the computational complexity, but it may result in a loss in performance.

Consider the network topology in Fig. 2.2. Here, we examine various choices of nonadaptive code rate R within the feasible range $[0, R_0]$. We can see in Fig. 2.8 that by using non-adaptive channel coding, the highest throughput is achieved when the code rate on all links is equal to 0.74. At this point, we reach almost the optimal value that is achievable by using adaptive channel coding. It is also interesting to investigate the distribution of the optimal adaptive code rates of all wireless links, compared to the optimal non-adaptive code rate. We can see in Fig. 2.9 that in the adaptive channel coding case, the optimal code rates for various links can be significantly different. It is interesting to note that the code rates corresponding to the links which are not in any routing path (i.e., link 21) are chosen to be 1. Moreover, links which are used in many routing paths have code rates close to the corresponding non-adaptive channel code rate (0.74).

2.4.6 The Effect of Dynamic Changes on the System Performance

In this section, we study the effect of dynamic topology changes as well as changes in the number of network users on the network performance. As mentioned in Section 2.3.1, whenever the setting of the network changes, the algorithm solves the new problem by updating the last obtained end-to-end data rate vector, which is used as the new initial point for faster convergence. This may be beneficial especially in dynamic environments such as vehicular networks where the vehicles move constantly. Fig. 2.10 shows the convergence of the algorithm when changes happen in the network and compares it with the case when the algorithm does not use the available information related to the previous state of the network. In Fig. 2.10 (a), five randomly chosen links are added to and five random links are removed from the current topology every 100 time slots. In Fig. 2.10 (b), a new pair of source-destination node is removed from the network. Finally, in Fig. 2.10 (d), a randomly chosen pair is added and a randomly chosen pair is removed every 100 time slots. Fig. 2.10 shows that using the available information from the previous state of the network significantly increases the convergence speed of the algorithm.

2.4.7 The Effect of Mobility on the System Performance

In this section, we study the effect of mobility in a vehicular network on the performance of our proposed design. In a vehicular network, users (i.e., vehicles) are always moving and in each instant, they connect to the nearest access point (a mesh node) for network provisioning. This results in dynamic changes in the traffic pattern which in turn leads to performance degradation in the system. The degree of performance reduction depends to the coverage area of the access points as well as the speed at which the vehicles move.



Figure 2.10: Convergence speed for Algorithm 2.1 in presence of dynamic changes in the network. We compare two cases where the previous solution is exploited and the case where the previous solution is ignored. Every 100 time slots: (a) five links are added and five links are removed, (b) a new pair of source-destination nodes is added, (c) a pair of source-destination node is removed, (d) a random pair is added and another random pair is removed.

Consider the example downtown area shown in Fig. 2.1. There are ten cars which move in random directions. In each instant, they connect to the nearest access point to communicate with the gateway. The network recalculates the optimal data rates and channel code rates every 5 seconds based on the most recent topology characteristics of the network. Clearly, the higher the speed of the vehicles, the larger will be the changes in the network. This can lead to a performance degradation. Fig. 2.11 shows the convergence of the adaptive scheme compared to the optimal value when the vehicles move with velocities of 20, 40, and 80 km/h. The rates are updated 100 times in a 500 second period. It is shown in Fig. 2.11 that the number of instants where the performance of the network deviates from the optimal solution increases when vehicles speed up. It is interesting that while the vehicles move in the area, the optimal solution does not change. This is because the destination for all data flows is the gateway and therefore there is a bottleneck around that node. Thus, although the source nodes change, the bottleneck remains and the achieved aggregate throughput remains unchanged. However, the allocated rates corresponding to



Figure 2.11: The convergence of the algorithm is shown for different speeds. Recalculations occur every 5 seconds if needed.

different access points change such that the minimum throughput also remains optimal.

The average minimum throughput of the network over 20 random scenarios is shown in Fig. 2.12 when the speed of the vehicles changes from 10 to 100 km/h. We observe that the average performance degrades when the speed increases under adaptive channel coding because more changes occur between two successive problem reformulation. However, the performance *remains optimal* under non-adaptive channel coding. This is because nonadaptive channel coding is less complex and converges faster to the final solution.

2.4.8 Impact of Fading

Finally, we study the impact of *fading* on the system performance when Algorithm 2.1 is used. Recall from Section 2.3.1 that we can incorporate the impact of fading by separately solving problem (2.8) for each wireless channel realization with fading gains f_e and corresponding cut-off rates as in (2.4) and (2.5). In this case, Algorithm 2.1 is invoked every time new channel measurement data becomes available. We refer to each channel measurement data as one *channel snapshot*.

Simulation results for the network topology in Fig. 2.2 for 50 different channel snapshots



Figure 2.12: Performance of the algorithm is studied in the downtown area of Fig. 2.1 when vehicles move with different speeds.

are shown in Fig. 2.13. In our simulation model, we generate the fading gains for each channel snapshot based on a random realization of the Rayleigh fading distribution. For the results in Fig. 2.13, we compare the performance of two design scenarios. The first design is an adaptive channel coding scheme based on the *average* fading information. That is, solving problem (2.8) only *once* by assuming that the fading gains take their average values within the Rayleigh fading distribution. On the other hand, in our second design, we solve problem (2.8) once for *each* channel snapshot. We can see that on average, the latter case (solid line) can improve the minimum throughput among all end-to-end sessions by a factor of 6 compared to the former one (dash line). The achieved performance improvement is at the cost of a significantly higher computational complexity due to the requirement of solving problem (2.8) for each snapshot, which may not always be desired in practice. The snapshots in which the minimum throughput among the sessions is zero denote the scenarios where there is at least one link in all paths of one session that has an instantaneous cutoff rate which is less than its assigned code rate. This does not happen if the code rates are updated according to the channel information in each snapshot.

In summary, we showed that the adaptive channel coding approach converges to the op-



Figure 2.13: Performance trend in a fading channel for 50 channel snapshots.

timal solution in the presence of dynamic changes in the network due to channel variations and mobility. However, if the changes occur too frequently, the algorithm may fail to follow the changes fast enough and the performance degrades. On the other hand, we showed that non-adaptive channel coding is able to follow the dynamic changes and provides a high performance for the network without substantial sub-optimality.

2.5 Summary

In this chapter, we considered the problem of jointly using per-link channel coding in wireless networks and multipath routing. In this regard, we focused on per-link channel code rate selection and end-to-end transmission data rate allocation and formulated a maxmin fairness optimization problem, which is of interest in vehicular network applications to offer fair and consistent data rates. Unlike the case of single-path routing, solving this problem in a multipath routing network is hard and involves non-convex programming. We tackled the non-convexity by using appropriate function approximations and iterative techniques from signomial programming. We proposed a novel code and data rate selection algorithm which uses the available information related to the latest optimal solution in order to converge faster in highly changing conditions. Moreover, we studied different variations of our proposed per-link channel code rate selection and end-to-end data rate allocation algorithm in order to address both adaptive and non-adaptive channel coding and also the impact of fading. Simulation results confirm that by using channel coding jointly with multipath routing, we can significantly improve the end-to-end network performance compared to the case when only channel coding or only multipath routing is used. We also showed through simulations that as a sub-optimal approach with less complexity, nonadaptive channel coding achieves a high degree of optimality. Although our algorithm needs to be executed in a centralized manner, it can be applied in certain applications such as vehicular networks where stationary mesh nodes provide connectivity for moving vehicles. The centralized solution can also be used as a benchmark for distributed algorithms.

Chapter 3

Delay-Throughput Enhancement in Wireless Networks with Multipath Routing and Channel Coding

3.1 Introduction

While most of multimedia applications have strict QoS requirements [90], the existing best-effort traffic delivery model cannot provide any service guarantee with respect to the minimum throughput and maximum delay of the end-to-end flows. Therefore, it is important to design wireless networks with high performance in regard of delay and throughput.

We investigated the trade-off between reliability and throughput in achieving the highest possible *effective throughput*, which is the end-to-end throughput that the receiver is able to receive in Chapter 2 and [91]. We focused on *multipath routing* wireless systems, where *adaptive channel coding* is also performed at the physical layer. However, we *did not* address the issue of delay in Chapter 2. In this chapter, we explicitly incorporate delay in the utility of each session and propose a joint data rate and coding rate allocation algorithm that leads to maximizing the network aggregate utility across all sessions. Our work complements the existing results in the literature as follows. The recent work by O'Neill *et al.* [92] used NUM with adaptive modulation to determine the optimal sending rates and transmit powers that maximize system performance. The trade-off between data rate, energy consumption, and delay is studied. However, O'Neill et al. did not incorporate delay into the utility function in their problem formulation [92] and the proposed design neither minimizes the delay nor provides a bound on end-to-end delay. On the other hand, Saad et al. [93] used the M/G/1 queueing model to estimate the delay as the summation of transmission delay and queueing delay. The same authors examined upper bounds on delay [94] but did not focus on delay reduction. In work by Kallitsis *et al.* [95], resources are allocated to maximize the throughput of the network and minimize the delay. Delay is modeled using network calculus and is incorporated directly into the utility function. Another research direction focuses on resource allocation to enhance the network performance by only minimizing the delay (e.g., Li et al. [38] and Kalyanasundaram et al. [96]). However, the impact of adaptive channel coding has not been considered in this context. On the other hand, channel coding is considered in [22]; but no analysis is performed related to delay. Finally, our problem is closely related to the recent work by Li *et al.* [51], which only addresses single-path routing within the context of wired networks or wireless networks with fixed capacity links. The contributions of this chapter can be summarized as follows:

- We model a wireless network with several unicast data sessions, multiple routing paths for each session, and adaptive channel coding at the physical layer. To model the end-to-end delay, we use the average waiting time in an M/D/1 queueing system [97]. We then formulate the NUM problem of jointly finding optimal sending rates and code rates in the network to achieve the maximum network utility as a function of throughput and delay.
- We formulate two optimization problems with and without fairness provisioning. In the former one, we aim to maximize the minimum utility in the network. In the latter

case, we maximize the overall utility of the network. Fair resource allocation is of particular interest in vehicular networks in which moving vehicles frequently switch among stationary access points.

- To overcome the non-convexity due to channel coding, multipath routing, delay and reliability consideration, we introduce new variables, constraints, and approximations in the original problem and reformulate it as a series of tractable *geometric programming* problems [83].
- We develop an iterative algorithm to solve the formulated problem. To the best of our knowledge, there has been no prior work on jointly improving throughput and delay in a *wireless multipath* routing network with *channel coding* applied at the physical layer.
- Simulation results for random topologies show that, when fairness is not a concern, we can decrease the average delay by 60% at the cost of only a marginal (< 0.1%) degradation in throughput. We also show that if fairness is addressed, we can decrease the maximum delay across the network by more than 35% with less than 12% decrease in minimum throughput.

The system model and problem formulation are described in Section 3.2. The delay-aware optimal data rate and coding rate allocation approach is introduced in Section 3.3. The numerical results are shown in Section 3.4. The chapter is summarized in Section 3.5.

3.2 System Model

A wireless network is modeled as a directed graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} represents the set of nodes and \mathcal{E} represents the set of wireless links, as it is shown in Fig. 3.1. For each unicast



Figure 3.1: A sample topology with five unicast multipath data sessions. Data sessions are $(2 \rightarrow 15), (7 \rightarrow 18), (9 \rightarrow 20), (12 \rightarrow 23)$, and $(3 \rightarrow 25)$. These sessions have 6, 1, 2, 2, and 4 available paths, respectively.

session $i \in \mathcal{I}$, where $\mathcal{I} = \{1, 2, ..., I\}$, the source and destination nodes are denoted by s_i and t_i , respectively. We define \mathcal{K}_i , with $K_i = |\mathcal{K}_i|$, as the set of all available routing paths from s_i to t_i . Moreover, for each session $i \in \mathcal{I}$, each $k = 1, ..., K_i$, and each link $e \in \mathcal{E}$, we define

$$a_i^{ek} = \begin{cases} 1, & \text{if link } e \in k^{\text{th}} \text{ routing path for session } i, \\ 0, & \text{otherwise.} \end{cases}$$
(3.1)

For each session $i \in \mathcal{I}$, let α_i^k denote the data rate at source s_i on its k^{th} routing path, $k = 1, \ldots, K_i$. Channel coding can improve the reliability over lossy wireless channels by adding redundant bits to data packets. For each link $e \in \mathcal{E}$, we define R_e as the link *coding rate*, i.e., the ratio of the number of data bits at the input of the encoder to the number of data plus redundant bits at the output. Notice that if channel coding is not performed on link e, then $R_e = 1$. Given the data rates at the sources $\boldsymbol{\alpha} = (\alpha_i^k, i \in \mathcal{I}, k = 1, \ldots, K_i)$
and the link coding rates $\mathbf{R} = (R_e, e \in \mathcal{E})$, the aggregate traffic load on each link $e \in \mathcal{E}$ is $u_e = \frac{1}{R_e} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} a_i^{ek} \alpha_i^k$. The smaller the coding rate R_e , the more redundant data is added to the transmitted packets on link $e \in \mathcal{E}$ leading to more reliable transmissions, i.e., transmissions with lower error probability. However, this will be at the cost of exposing the link to higher traffic. Let $\mathbf{R}_0 = (R_{0e}, e \in \mathcal{E})$, where $R_{0e} \leq 1$ is the *cut-off rate* on wireless link $e \in \mathcal{E}$, that is an upper bound for the rate R_e achievable with certain codes (e.g., convolutional codes) [85]. In general, the cut-off rate R_{0e} depends on the received SNR and the modulation scheme being used. Given coding rate $R_e \leq R_{0e}$, the error probability on link e can be modeled as [85]

$$P_e = 2^{-T(R_{0e} - R_e)},\tag{3.2}$$

where T is the coding block length. Based on the link failure model in (3.2), the probability that a packet is successfully transmitted along the k^{th} routing path, $k = 1, \ldots, K_i$ for session $i \in \mathcal{I}$ is given by $\prod_{e \in \mathcal{E}, a_i^{ek} = 1} (1 - P_e) = \prod_{e \in \mathcal{E}} (1 - a_i^{ek} P_e)$. From the above equation, for each session $i \in \mathcal{I}$, the aggregate *effective* throughput at destination t_i becomes

$$\sum_{k=1}^{K_i} \alpha_i^k \prod_{e \in \mathcal{E}} \left(1 - a_i^{ek} P_e \right). \tag{3.3}$$

To obtain the average end-to-end delay, we model each link as a single M/D/1 queue based on the Kleinrock independence approximation [97]. Here, we assume that the arrival rates in the source nodes follow a Poisson distribution. Since the transmission times over all links are deterministic, the number of arrivals for each queue in any time interval can be assumed to follow a Poisson distribution with rate

$$\lambda_e = \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} a_i^{ek} \alpha_i^k / L, \qquad (3.4)$$

55

where L is the packet length. From Little's Theorem [97], the average waiting time for each queue e corresponding to link $e \in \mathcal{E}$ is given by

$$\delta_e^Q = \frac{L \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}}{2c_e R_e (c_e R_e - \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'})},\tag{3.5}$$

where c_e denotes the *nominal* data rate of link $e \in \mathcal{E}$. By adding the waiting time and the transmission time $\delta_e^T = \frac{L}{c_e R_e}$ together, we have $\delta_e = \delta_e^Q + \delta_e^T$ for each link $e \in \mathcal{E}$. Then, the average end-to-end delay for each path $k = 1, \ldots, K_i$ of session $i \in \mathcal{I}$ can be written as

$$\delta_{i}^{k} = \frac{L}{2} \sum_{e \in \mathcal{E}} \frac{a_{i}^{ek}}{c_{e}R_{e}} \left(2 + \frac{\sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}}{c_{e}R_{e} - \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}} \right).$$
(3.6)

To model the mutual interference among the wireless links in the network, we use the concept of contention graph. In the contention graph $G_C(\mathcal{V}_C, \mathcal{E}_C)$ corresponding to network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the set of vertices \mathcal{V}_C represents the set of all wireless links \mathcal{E} in the network graph G. An edge connects any two vertices in set \mathcal{V}_C if the corresponding wireless links in the network graph mutually interfere with each other. That is, if the receiver node of one link is within the interference range of the sender node of the other link. Given the contention graph, each complete subgraph is called a *clique*. A maximal clique is a clique which is not a subgraph of any other clique [86]. We denote the set of all maximal cliques in G_C by \mathcal{Q} . In each instant, only one link among all members of a maximal clique $Q \in \mathcal{Q}$ can be active. The ratio $\frac{u_e}{c_e}$ denotes the portion of time that link $e \in \mathcal{E}$ is active when it is being used with data rate u_e . It is required that $\sum_{e \in \mathcal{Q}} \frac{u_e}{c_e} \leq \nu$ for each clique $Q \in \mathcal{Q}$ where $\nu \in (0, 1]$ is called the *clique capacity*. Note that $\nu = 1$ is a necessary constraint on the clique capacity. It may not always be possible to find feasible schedules that achieve a clique capacity in order to obtain a feasible schedule for the links in the clique [87].

We formulate the problem of jointly allocating coding rates and sending data rates such that the utility of the network be maximized. The utility of each session $i \in \mathcal{I}$ is defined as

$$U_i(\boldsymbol{\alpha}, \boldsymbol{R}) = (1 - w) \sum_{k=1}^{K_i} \alpha_i^k \prod_{e \in \mathcal{E}} \left(1 - a_i^{ek} P_e \right) - w \sum_{k=1}^{K_i} \delta_i^k, \qquad (3.7)$$

where δ_i^k is as in (3.6). Here, the utility of session *i* is a *weighted trade-off* between the session's aggregate effective throughput and its average delay. It is a trade-off because it can be increased by either increasing the throughput or decreasing the delay. We can tune the importance of delay by changing its weight, *w*. By increasing *w*, we move on the trade-off curve towards decreasing delay at the cost of decreasing the throughput. We define the utility of the network as either the summation of all utilities of data sessions $i \in \mathcal{I}$, or just the one with the minimum value.

Maximizing the Aggregate Utility of the Network

This problem is formulated as

$$\begin{array}{ll}
\underset{\boldsymbol{\alpha} \succeq \mathbf{0}, \ \mathbf{0} \prec \boldsymbol{R} \preceq \boldsymbol{R}_{\mathbf{0}}}{\text{maximize}} & (1-w) \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_{i}} \alpha_{i}^{k} \prod_{e \in \mathcal{E}} \left(1 - a_{i}^{ek} P_{e} \right) - w \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_{i}} \delta_{i}^{k} \\
\text{subject to} & \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_{i}} a_{i}^{ek} \alpha_{i}^{k} R_{e}^{-1} c_{e}^{-1} \leq \nu, \\
& \delta_{i}^{k} \leq \delta_{i}^{max}, & i \in \mathcal{I}, \ k = 1, \dots, K_{i}, \\
& (3.8)
\end{array}$$

where δ_i^{max} is the maximum delay that can be tolerated for each path of session $i \in \mathcal{I}$. The set of constraints declare that the portion of time that all links in a maximal clique are active must be less than the clique capacity. The expressions for P_e and δ_i^k are as in (3.2) and (3.6), respectively.

Maximizing the Minimum Utility of the Network

This problem is formulated as

$$\begin{array}{ll}
\underset{\boldsymbol{\alpha} \succeq \mathbf{0}, \ \mathbf{0} \prec \boldsymbol{R} \preceq \boldsymbol{R}_{\mathbf{0}}}{\text{maximize}} & \min_{i \in \mathcal{I}} (1-w) \sum_{k=1}^{K_{i}} \alpha_{i}^{k} \prod_{e \in \mathcal{E}} \left(1-a_{i}^{ek} P_{e} \right) - w \sum_{k=1}^{K_{i}} \delta_{i}^{k} \\
\text{subject to} & \sum_{e \in Q} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_{i}} a_{i}^{ek} \ \alpha_{i}^{k} \ R_{e}^{-1} \ c_{e}^{-1} \leq \nu, \quad Q \in \mathcal{Q}, \\
& \delta_{i}^{k} \leq \delta_{i}^{max}, \qquad i \in \mathcal{I}, \ k = 1, \dots, K_{i}.
\end{array}$$

$$(3.9)$$

Unlike (3.8), here the design addresses the notion of maxmin fairness among sessions.

3.3 Delay-aware Optimal Data Rate and Coding Rate Allocation

The optimization problem in (3.8) is non-convex and non-separable due to the product forms in the objective function with respect to the effective throughput, the fractional forms in the first set of constraints and in the delay constraints in (3.6), the exponential forms in the objective function with respect to error probabilities, the non-separability of reliability and throughput due to multipath routing, and the coupling across variables because of delay constraints and channel coding. Most of the above properties are due to the fact that we consider multipath routing and wireless interference. For example, if we assume there is no interference, which is true for wired networks, the clique capacity constraints would reduce to *linear* link capacity constraints for any link $e \in \mathcal{E}$:

$$\frac{1}{R_e c_e} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} a_i^{ek} \alpha_i^k \le 1, \quad \Rightarrow \quad \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} a_i^{ek} \alpha_i^k - R_e c_e \le 0.$$
(3.10)

We can also show that the non-convexity due to the product forms in the objective function can be resolved if there is only a *single* routing path for each session [22]. However, all sources of complexity remain in place when multipath routing is used and wireless transmissions are subject to interference. In the following, we use various techniques to overcome the complexity of the problem formulation and convert problem (3.8) into a convex problem.

Consider the exponential form of P_e in (3.2). For notational simplicity, we can rewrite (3.2) as $P_e = X_e \exp(Z R_e)$ for each link $e \in \mathcal{E}$, where $X_e = 2^{-TR_{0e}}$ and $Z = T \ln 2$. We can use *Taylor series expansion* and rewrite the above equation as $P_e = X_e \sum_{n=0}^{\infty} \frac{(Z R_e)^n}{n!}$. Clearly, for some bounded integer $N_e \gg 1$, we can approximate P_e as $X_e \sum_{n=0}^{N_e} \frac{(Z R_e)^n}{n!}$ for each link $e \in \mathcal{E}$. We investigate the value of N_e necessary for obtaining a good approximation through simulation. If the error probabilities P_e are small, we can rewrite the receiving rates in each session as

$$\sum_{k=1}^{K_i} \alpha_i^k \prod_{e \in \mathcal{E}} \left(1 - a_i^{ek} P_e \right) \approx \sum_{k=1}^{K_i} \alpha_i^k \left(1 - \sum_{e \in \mathcal{E}} a_i^{ek} P_e \right).$$
(3.11)

Due to the polynomial forms in the objective function and the constraints, we can solve problem (3.8) by using geometric programming techniques. In this respect, using the approximated value for P_e , we replace (3.11) in the objective function of problem (3.8) and introduce variable t such that t is a lower-bound for the objective function. That is,

$$t + (1 - w) \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \sum_{e \in \mathcal{E}} \sum_{n=0}^{N_e} \frac{\alpha_i^k a_i^{ek} X_e(ZR_e)^n}{n!} + w \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \delta_i^k \le (1 - w) \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \alpha_i^k.$$
(3.12)

Then, we follow the signomial programming techniques [83] to approximate the polynomial in the right-hand side of (3.12), which is only a function of α , as a *monomial*, i.e., a polynomial with only *one* term and *positive* multiplier. This approximation can be done around some initial point $\hat{\alpha}$. For a parameter $f_s > 1$, which is close to 1, we have

$$\sum_{i\in\mathcal{I}}\sum_{k=1}^{K_i}\alpha_i^k \approx \left(\sum_{i\in\mathcal{I}}\sum_{k=1}^{K_i}\hat{\alpha}_i^k\right)\prod_{i\in\mathcal{I}}\prod_{k=1}^{K_i}\left(\frac{\alpha_i^k}{\hat{\alpha}_i^k}\right)^{\hat{\alpha}_i^k/\left(\sum_{i'\in\mathcal{I}}\sum_{k'=1}^{K_{i'}}\hat{\alpha}_{i'}^{k'}\right)}, \forall \ \boldsymbol{\alpha} \in [\hat{\boldsymbol{\alpha}}/f_s, f_s\hat{\boldsymbol{\alpha}}],$$
(3.13)

where $[\hat{\boldsymbol{\alpha}}/f_s, f_s \hat{\boldsymbol{\alpha}}]$ is a small neighborhood around initial point $\hat{\boldsymbol{\alpha}}$. For notational convenience, we define $\hat{\Lambda}$, which only depends on the initial point $\hat{\boldsymbol{\alpha}}$, as $\hat{\Lambda}^{-1} = \left(\sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \hat{\alpha}_i^k\right)$. Then inequality (3.12) can be approximated around the initial point $\hat{\boldsymbol{\alpha}}$ as

$$\frac{\hat{\Lambda}}{1-w} \left(t + (1-w) \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \sum_{e \in \mathcal{E}} \sum_{n=0}^{N_e} \frac{\alpha_i^k a_i^{ek} X_e(Z \ R_e)^n}{n!} + w \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \delta_i^k \right) \prod_{i \in \mathcal{I}} \prod_{k=1}^{K_i} \left(\frac{\alpha_i^k}{\hat{\alpha}_i^k} \right)^{-\hat{\alpha}_i^k \hat{\Lambda}} \leq 1.$$
(3.14)

The above constraint is a *posynomial*, i.e., a polynomial with only *positive* terms. Posynomials are the building blocks in geometric programming [83]. By minimizing t^{-1} , we maximize the objective function in (3.8).

To tackle the fractional forms in the delay constraints, we can write (3.6) in an inequality form

$$\delta_{i}^{k} \geq \frac{L}{2} \sum_{e \in \mathcal{E}} \frac{a_{i}^{ek}}{c_{e}R_{e}} \left(2 + \frac{\sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}}{\left(c_{e}R_{e} - \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'} \right)} \right).$$
(3.15)

for each $i \in \mathcal{I}, k = 1, ..., K_i$. It can be shown that (3.15) is always satisfied with equality for the optimal solution. This can be proved by contradiction. Assume that (3.15) is not satisfied with equality in the *optimal solution* for some *i* and *k*. Note that δ_i^k is not lower bounded in any other set of constraints. Therefore, we can decrease δ_i^k such that the corresponding constraint is satisfied with equality. This leads to further increasing the value of the objective function by choosing a solution different than the optimal solution which is a contradiction. That is, it is an active inequality constraint. For each link $e \in \mathcal{E}$, we introduce new variables Y_e such that

$$\frac{Y_e^{-1}}{c_e} + \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \frac{\alpha_{i'}^{k'} R_e^{-1}}{c_e} \le 1, \quad \forall \ e \in \mathcal{E}.$$
(3.16)

We can show that (3.15) is satisfied if (3.16) holds and we have

$$\delta_i^k \ge \frac{L}{2} \sum_{e \in \mathcal{E}} a_i^{ek} \left(\frac{2}{c_e R_e} + \frac{Y_e \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}}{c_e R_e^2} \right), \qquad \forall i \in \mathcal{I}, \ k = 1, \dots, K_i.$$
(3.17)

Similarly, we can show that (3.16) and (3.17) are always satisfied with equality for the optimal solution.

By introducing t as in (3.14) and adding constraints (3.16), and (3.17) to the constraints of problem (3.8), it is equivalent to the problem (3.18) in which the objective is to minimize t^{-1} which is equal to maximizing t where t is declared to be a lower bound for the network utility function in the first set of constraints. The second set of constraints declare the capacity constraints and the third and fourth set are active constraints (3.17) and (3.16), respectively. The last set of constraints guarantees all end-to-end delays to be bounded:

$$\underset{t>0,\hat{\alpha}/f_{s}\preceq\alpha\preceq f_{s}\hat{\alpha}, \ \mathbf{0}\prec \mathbf{R}\preceq \mathbf{R}_{\mathbf{0}}, \boldsymbol{\delta}\succ\mathbf{0}, \mathbf{Y}\succ\mathbf{0}}{\text{minimize}} t^{-1}$$

subject to

$$\frac{\hat{\Lambda}}{1-w} \left(t + (1-w) \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \sum_{e \in \mathcal{E}} \sum_{n=0}^{N_e} \frac{\alpha_i^k a_i^{ek} X_e(ZR_e)^n}{n!} + w \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} \delta_i^k \right) \left(\prod_{i \in \mathcal{I}} \prod_{k=1}^{K_i} \left(\frac{\alpha_i^k}{\hat{\alpha}_i^k} \right)^{-\hat{\alpha}_i^k \hat{\Lambda}} \right) \le 1,$$

$$\frac{1}{\nu} \sum_{e \in \mathcal{Q}} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K_i} a_i^{ek} \alpha_i^k R_e^{-1} c_e^{-1} \leq 1, \qquad \forall Q \in \mathcal{Q},$$

$$\frac{L}{2} \sum_{e \in \mathcal{E}} \frac{a_i^{ek}}{c_e} (2R_e^{-1} + R_e^{-2}Y_e \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} \alpha_{i'}^{k'}) \delta_i^{k-1} \leq 1,$$

$$\forall i \in \mathcal{I}, k = 1, \dots, K_i,$$

$$\frac{Y_e^{-1}}{c_e} + \sum_{i' \in \mathcal{I}} \sum_{k'=1}^{K_{i'}} a_{i'}^{ek'} R_e^{-1} \alpha_{i'}^{k'} \leq 1, \qquad \forall e \in \mathcal{E},$$

$$\delta_i^k \leq \delta_i^{max}, \qquad \forall i \in \mathcal{I}, k = 1, \dots, K_i.$$
(3.18)

Problem (3.18) is a standard geometric programming problem that can be converted into a convex optimization problem [83] and can be solved around an initial point. It has been shown that iteratively solving (3.18) converges to the optimal solution of problem (3.8) [83]. In each iteration, (3.18) is initialized with the optimal solution of the problem corresponding to the last iteration. As discussed, we may move on the throughput-delay trade-off curve by tuning the delay importance weight w. w must be chosen such that the objective function in problem (3.8) remains positive. Similarly, we can convert problem (3.9) into a convex optimization problem. Compared to solving problem (3.8), there are two differences. First, variable t is introduced such that

$$t \le (1-w) \sum_{k=1}^{K_i} \alpha_i^k \left(1 - \sum_{e \in \mathcal{E}} a_i^{ek} X_e \sum_{n=0}^{N_e} \frac{(Z \ R_e)^n}{n!} \right) - w \sum_{k=1}^{K_i} \delta_i^k, \quad \forall \ i \in \mathcal{I}.$$
(3.19)

As in the earlier case, we use signomial techniques to convert (3.19) into a constraint in the standard form of geometric programming problems. By applying a similar technique in (3.13) this time to $\sum_{k=1}^{K_i} \alpha_i^k$, we can rewrite (3.19) as

$$\frac{\hat{\Lambda}_i}{1-w} \left(t + (1-w) \sum_{k=1}^{K_i} \sum_{e \in \mathcal{E}} \sum_{n=0}^{N_e} \frac{\alpha_i^k a_i^{ek} X_e(Z \ R_e)^n}{n!} + w \sum_{k=1}^{K_i} \delta_i^k \right) \prod_{k=1}^{K_i} \left(\frac{\alpha_i^k}{\hat{\alpha}_i^k} \right)^{-\hat{\alpha}_i^k \hat{\Lambda}_i} \le 1, \quad \forall \ i \in \mathcal{I},$$

$$(3.20)$$

where $\hat{\Lambda}_i^{-1} = (\sum_{k=1}^{K_i} \hat{\alpha}_i^k)$. Second, inequalities (3.16) and (3.17) may not be active anymore and so inequality (3.15) may not be satisfied with equality. In this case, we are in fact dealing with the upper bounds of the average end-to-end delay in the objective function. In this way, the performance of the network is better than what we would expect from the obtained solution in terms of average delay since the upper bounds on the average delay are used in the objective function. The rest of the formulation is the same as the one in (3.18). Again, w must be chosen such that the objective function in problem (3.9) remains positive.

3.4 Numerical Results

In this section, we numerically solve problem (3.8) to determine the optimal sending and coding rates such that the utility of the network (i.e., the trade-off between the aggregate effective throughput and the average delay) is maximized. We show that maximizing the aggregate network utility leads to higher benefits in the throughput-delay trade-off (i.e., we obtain lower delays at the cost of lower decrease in throughput), compared to the case of maximizing the minimum utility in the network. In the former case, no fairness is achieved and some sessions may be starved. Therefore, we also solve problem (3.9) to determine the sending rates and coding rates such that the minimum utility of the network is maximized. We show that at the cost of loosing some gain in the throughput-delay trade-off we can provide fairness among all sessions. In our set of simulations, we use T = 10, L = 8000bits, $c_e = 11$ Mbps, $N_e = 15$, $f_s = 1.1$, $R_{0e} = 1$, and $\nu = \frac{2}{3}$ [87]. The proper values for



Figure 3.2: The average delay decreases as its importance weight in the objective function increases. Delay values are normalized over the corresponding values for which w = 0. We observe a decrease of almost 58% when w = 0.5.

parameters N_e and f_s are obtained from simulations.

We solve problem (3.8) for different values of w in a feasible range across 50 different random topologies. Each random topology is a 5 × 5 grid topology for which 20 nodes are placed in random locations. Five pairs of nodes are randomly selected as the source and destination pairs. We observe that by considering the delay in the objective function, even with a small weight, we can decrease the average delay by 37% compared to the case with no delay consideration (Fig. 3.2). We also note that by further increasing the importance weight of delay in the objective function the average delay decreases by 58%. The more a link is utilized, the higher will be the queuing delays for that link. Therefore, decreasing the average delay leads to the use of the intermediate links at a lower rate that in turn leads to a slight decrease (0.1%) in aggregate throughput. This confirms the delay-throughput trade-off. We can also see that by decreasing the throughput by a small amount, the delay decreases dramatically at the starting point. This is because delay is an exponential function of the utilization rate.

Maximizing the total throughput usually requires sacrificing fairness among sessions. That is, some sessions may starve while some other sessions use the network with a higher throughput. For instance after solving problem (3.8) for a sample topology, we observe



Figure 3.3: The normalized maximum delay among the users in the network decreases as the corresponding importance weight in the objective function increases. Delay values are normalized over the values corresponding to w = 0. We observe almost 37% decrease in the maximum delay in the network when w reaches 0.5.

that sessions 1 and 3 use the network at a rate of 2 Mbps while sessions 2 and 5 starve and session 4 sends data at a rate of 4.5 Mbps. To provide fairness among sessions, we solve problem (3.9) to maximize the minimum utility across sessions for different feasible choices of parameter w and for 50 randomly selected topologies. Maximizing the minimum utility of the network, we expect that there will be no session with starvation. We can see that the normalized maximum end-to-end delay in the network among all routing paths decreases by almost 5% on average when we consider the delay only by a very small weight (Fig. 3.3). We can further decrease the delay by 37% when w increases. Again, there exists a trade-off between the maximum end-to-end delay of the network and the aggregate throughput of the session with the minimum aggregate throughput in Fig. 3.4. By tracing the graph starting from the upper right corner (w = 0) to the lower-left corner (w = 0.5), we gain a 37% improvement in the maximum delay at a cost of a slight decrease at only 11% in the minimum throughput when we are in proper points of the curve.

We use Jain's fairness index to quantitatively measure the fairness of the throughput attained among different unicast sessions. Let Ψ denote Jain's fairness index. We have $\Psi = \frac{(\sum_{i \in \mathcal{I}} x_i)^2}{|\mathcal{I}| \sum_{i \in \mathcal{I}} x_i^2}$, where x_i denotes the total effective throughput of flow $i \in \mathcal{I}$ from (3.3).



Figure 3.4: The trade-off between the maximum delay and the minimum throughput in the network. We may move on the curve by tuning w, the delay importance weight. Delay and throughput are normalized over their corresponding values at w = 0.



Figure 3.5: Fairness index is shown for both problem (3.8) (aggregate utility maximization) and problem (3.9) (maxmin fairness).

We can see in Fig. 3.5 that fairness is improved when the resource allocation is based on the solution of problem (3.9).

By now, we considered the normalized values of delay over their value when w = 0(delay is not considered). It is interesting to see how the average maximum delays in the network change under different delay guarantees. We solve problem (3.9) without considering the last set of constraints (delay guarantee constraints), and also when δ_i^{max} is 10 ms and 20 ms for all $i \in \mathcal{I}$ (Fig. 3.6). We can see that the delay is guaranteed to be less than δ_i^{max} .



Figure 3.6: Average maximum delay is shown when there is no guarantee on the maximum delay, there is a guarantee of 10 ms, and 20 ms. Since the guarantee is imposed on the upper bound of the delay, the delay may not reach the guaranteed value.

As mentioned earlier, the decrease in delay is gained at the cost of decreasing the utility of the links which in turn leads to a decrease in the overall throughput. This throughput degradation can be compensated for by using channel coding. To determine the effect of channel coding, we consider the throughput-delay trade-off in a network in which channel coding is *not performed* and we study how this can affect the network performance. We assume a packet error rate of 30% at each link and solve problem (3.9) without channel coding to show how it affects the performance. By increasing the weight of delay, we can only decrease the maximum delay by around 3% at the cost of 22% decrease in minimum throughput in Table 3.1. w varies in its feasible range. This shows how performance degrades when channel coding is not used and reveals the importance of channel coding.

Table 3.1: The trade-off between the maximum delay and the minimum throughput in the network when channel coding is not being used.

w	Normalized Maximum	Normalized Minimum
	Delay	${f Throughput}$
0	1	1
0.3	0.97	0.78

3.5 Summary

In this chapter, we considered the trade-off between decreasing the end-to-end delay and increasing the aggregate throughput in wireless networks with channel coding and showed that a noticeable enhancement across both design goals is feasible if a combination of multipath routing and adaptive channel coding are employed. We jointly formulated the end-to-end data rate allocation and adaptive channel coding within the general framework of network utility maximization (NUM) under two variations. The first problem is formulated for maximization of the aggregate network utility, i.e., the overall system performance. The second problem is formulated for maximization of the minimum utility among the end-to-end flows to achieve fairness. Due to non-convexities such as in product terms and fractional terms in the objective function and the constraints, the formulated optimization problems are non-convex and non-separable, and difficult to solve. Nevertheless, we introduced an algorithm that can solve the two NUM problems with low computational complexity. Through our simulation studies, we note that, in many cases, significant improvement in end-to-end delay can be obtained with marginal decrease in aggregate throughput, suggesting that satisfying stringent delay requirements can be achieved if multipath routing and adaptive channel coding are employed. The fair resource allocation aspect of our proposed design is of interest in vehicular networks where multiple vehicles share an access point in order to obtain connectivity to the Internet. The centralized solution that we have proposed in this chapter can particularly be used in the case when the stationary access point provides connectivity for all vehicles that it serves. It can also serve as a benchmark for distributed algorithms which are to be developed in future. Nevertheless, a distributed algorithm can support much broader ranges of application types.

Chapter 4

Reliability-based Rate Allocation in Wireless Inter-session Network Coding Systems

4.1 Introduction

In Chapters 2 and 3, we developed rate allocation algorithms with channel coding and multipath routing to improve the performance of the network. In this chapter, we focus on inter-session network coding for multiple unicast sessions in a wireless network. The target applications are ones that require reliable data transfer (e.g., HTTP, FTP, P2P traffic) in multi-hop wireless mesh networks. Our objective is to increase the network utility and the *end-to-end reliability* of data transmissions by proper allocation of routing and inter-session network coding rates for each data source in the network. For this purpose, we use the failure probability of intermediate links to calculate the reliability (i.e., the probability that data is successfully received) of various routing and network coding paths. Given the calculated reliability information, we maximize the effective aggregate network throughput by choosing the optimal rate allocation for network coding paths. We use the *network utility maximization* framework developed by Kelly *et al.* [33]. To the best of our knowledge, there has been no prior work on improving the *end-to-end reliability* in an inter-session network coding system among unicast sessions.

The contributions of this chapter are as follows:

- We develop a recursive algorithm to calculate end-to-end reliability, i.e., the probability of correctly delivering a packet, over each routing and network coding path. This allows us to mathematically model the effective throughput for each unicast session in the network.
- We formulate a network utility maximization problem for unreliable inter-session network coding systems. This problem formulation takes into account the network topology, mutual interference among wireless links, session utility functions, and link reliability information.
- We propose a distributed algorithm to solve the formulated network utility maximization problem using the *proximal method* and *gradient projection*.
- Simulation results show that by taking into account the reliability information, the aggregate network throughput can be increased by up to 100% while the aggregate network utility is also improved significantly.

Unlike intra-session network coding, there is no dominant coding scheme for intersession network coding. Our inter-session network coding scheme is similar to the scheme by Khreishah *et al.* [50] for wired networks. We, however, extend that model [50] to the *wireless* networking case by representing wireless capacity using the *contention graph*. Our proposed approach is based on cooperation among all network users. Game theoretic analysis of inter-session network coding with non-cooperative users is also studied in [98].

The rest of this chapter is organized as follows. The system model is described in Section 4.2. Our algorithm to calculate end-to-end reliability is presented in Section 4.3. We solve the considered network utility maximization problem in Section 4.4 using a distributed

algorithm. Simulation results are presented in Section 4.5. A summary is given in Section 4.6.

4.2 System Model

Consider a wireless network modeled as a directed acyclic graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of all nodes in the network and \mathcal{E} is the set of all wireless links. We denote $e = (u, v) \in \mathcal{E}$ if and only if $d(u, v) \leq d_T$, where d(u, v) is the Euclidean distance between nodes u and v, and d_T denotes the maximum transmission range. Let $\mathcal{S} = \{1, 2, \ldots, S\}$ denote the set of all unicast sessions in the network. Each session $i \in \mathcal{S}$ is denoted by a tuple (s_i, t_i) , where s_i and t_i denote the source node and the destination node of session i, respectively.

4.2.1 Pair-wise Inter-session Network Coding

Following the inter-session network coding model in [45], we model the network graph $G(\mathcal{V}, \mathcal{E})$ as a superposition of S routing subgraphs $G_r^{(i)}(\mathcal{V}_r^{(i)}, \mathcal{E}_r^{(i)})$ for all $i \in S$ and S(S-1) pairwise inter-session network coding subgraphs $G_{nc}^{(i,j)}(\mathcal{V}_{nc}^{(i,j)}, \mathcal{E}_{nc}^{(i,j)})$ for all $i, j \in S$ such that $i \neq j$. For each session $i \in S$, routing subgraph $G_r^{(i)}$ includes all routing paths from source node s_i to destination node t_i . We define $\mathcal{P}_r^{(i)}$ as the set of all routing paths $P_{s_i t_i}$ in graph $G_r^{(i)}$, where $P_{s_i t_i}$ denotes a path from source s_i to destination t_i . We also define $N_r^{(i)} = |\mathcal{P}_r^{(i)}|$. Furthermore, for each $k = 1, \ldots, N_r^{(i)}$, we define $\epsilon_i^{ek} = 1$ if link e belongs to the k^{th} routing path of session i; otherwise, $\epsilon_i^{ek} = 0$.

For any two sessions $i, j \in S$, we define $\mathcal{P}_{nc}^{(i,j)}$ as the set of all triples $\{P_{s_it_i}, P_{s_jt_j}, P_{s_jt_i}\}$ in graph $G_{nc}^{(i,j)}$ such that at most two of the three paths $P_{s_it_i}$, $P_{s_jt_j}$, and $P_{s_jt_i}$ share the same link. We define $N_{nc}^{(i,j)} = |\mathcal{P}_{nc}^{(i,j)}|$. For a pair of sessions *i* and *j*, we construct subgraph $G_{ij}^{lm}(\mathcal{V}_{ij}^{lm}, \mathcal{E}_{ij}^{lm})$ by the union of the *l*th triple path in $\mathcal{P}_{nc}^{(i,j)}$ and the *m*th triple path in $\mathcal{P}_{nc}^{(j,i)}$. The graphs G_{ij}^{lm} can be used to implement inter-session network coding (cf. [45, 50]) as we



Figure 4.1: (a) An example directed acyclic network graph with 6 nodes and 7 links. One data session exists between s_1 and t_1 and another one exists between s_2 and t_2 . (b) The corresponding (undirected) contention graph.

explain in the following example.

Consider the network in Fig. 4.1(a), where $\mathcal{V} = \{s_1, s_2, v_1, v_2, t_1, t_2\}$ and $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. This network topology is sometimes referred to as the *butterfly* network [45, 47–50]. We can see that there is only one routing path from s_1 to t_1 in Fig. 4.1(a), denoted by $(e_1e_4e_7)$. Similarly, there is only one routing path from s_2 to t_2 , denoted by $(e_2e_4e_6)$. Therefore, sets $\mathcal{P}_r^{(1)}$ and $\mathcal{P}_r^{(2)}$ have only one member and $N_r^{(1)} = N_r^{(2)} = 1$. On the other hand, set $\mathcal{P}_{nc}^{(1,2)}$ has one member, denoted by triple $\{e_2e_4e_6, e_1e_4e_7, e_2e_4e_6, e_5\}$. Similarly, set $\mathcal{P}_{nc}^{(2,1)}$ has one member, denoted by triple $\{e_2e_4e_6, e_1e_4e_7, e_3\}$. Therefore, $N_{nc}^{(1,2)} = N_{nc}^{(2,1)} = 1$. Furthermore, graph $G_{12}^{11} = G_{21}^{11} = G$. To implement network coding, node v_1 jointly encodes packets it receives from source nodes s_1 and s_2 . The encoded packets are then transmitted to receiver nodes t_1 and t_2 via node v_2 . Node t_1 can decode the received packets using the remedy data it receives from node s_2 over side link e_5 . Similarly, node t_2 can decode the received packets using the remedy data it receives from node s_1 over side link e_3 . Notice that encoding packets at node v_1 reduces the required bandwidth on links e_4 , e_6 and e_7 , leading to an increase in network throughput.

In a general network, the network coding scheme can be constructed by using the *add-up* and reset scheme [45]. Here, we assume that the network coding graphs are predetermined.

4.2.2 Rate Allocation

For each session $i \in \mathcal{S}$, let α_i^k denote the data rate of source s_i on the k^{th} routing path in subgraph $G_r^{(i)}$ for $k = 1, \ldots, N_r^{(i)}$. Also let α_{ij}^{lm} denote the rate of source s_i on network coding subgraph G_{ij}^{lm} for $l = 1, \ldots, N_{nc}^{(i,j)}$ and $m = 1, \ldots, N_{nc}^{(j,i)}$, where $j \in \mathcal{S} \setminus \{i\}$. Clearly, we must have $\alpha_{ij}^{lm} = \alpha_{ji}^{ml}$. The aggregate *sending* rate of source s_i is obtained as

$$\sum_{k=1}^{N_r^{(i)}} \alpha_i^k + \sum_{j \in \mathcal{S}, j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{ij}^{lm}.$$
(4.1)

Notice that since the wireless links are prone to error, the effective *receiving* rate at the destination node t_i can be different from the above sending rate at source node s_i . We will investigate this issue in detail in Sections 4.2.4 and 4.3.

Given the sending rates, we can also model the aggregate traffic load on each wireless link $e \in \mathcal{E}$ as [50]:

$$u_e = \sum_{i \in \mathcal{S}} \left(\sum_{k=1}^{N_r^{(i)}} \epsilon_i^{ek} \alpha_i^k + \sum_{j>i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(i,j)}} \alpha_{ij}^{lm} \max\{H_{ij}^{el}, H_{ji}^{em}\} \right), \tag{4.2}$$

where $H_{ij}^{el} = 1$ if link *e* belongs to at least one path in the *l*th triple $\{P_{s_it_i}, P_{s_jt_j}, P_{s_jt_i}\}$ of set $\mathcal{P}_{nc}^{(i,j)}$; otherwise, $H_{ij}^{el} = 0$. For the network in Fig. 4.1(a), $u_{e_1} = u_{e_7} = \alpha_1^1 + \alpha_{12}^{11}$, $u_{e_2} = u_{e_6} = \alpha_2^1 + \alpha_{12}^{11}$, $u_{e_3} = u_{e_5} = \alpha_{12}^{11}$, and $u_{e_4} = \alpha_1^1 + \alpha_2^1 + \alpha_{12}^{11}$.

4.2.3 Interference

In a wireless network, where some of the links can interfere with each other, mutual interference can be modeled using a *contention graph* $G'(\mathcal{V}', \mathcal{E}')$. In a contention graph G', the set of vertices \mathcal{V}' represents the set of all wireless links \mathcal{E} in the network graph G. There exists an edge between any two vertices in set \mathcal{V}' if the wireless links corresponding to the two vertices mutually interfere with each other. That is, if the receiver of one link is within the interference range of the sender of the other link. Given the contention graph, each *complete* subgraph, i.e., a subgraph in which all vertices are connected, is called a *clique*. A *maximal clique* is defined as a clique which is *not* a subgraph of any other clique. We denote the set of all maximal cliques in contention graph G' by Q. Each maximal clique is denoted by $Q_n \in Q$ for $n = 1, \ldots, |Q|$. Only one link among all the links corresponding to the vertices of a maximal clique can be active at a time.

Let c_e denote the *nominal* data rate of link $e \in \mathcal{E}$. The ratio $\frac{u_e}{c_e}$ denotes the portion of time that each link $e \in \mathcal{E}$ would be active. It is required that

$$\sum_{e \in Q_n} \frac{u_e}{c_e} \le \gamma, \qquad \forall Q_n \in \mathcal{Q}, \tag{4.3}$$

where $\gamma \in (0, 1]$ is called the clique capacity. It is common practice to select $\gamma = \frac{2}{3}$ [87]. For the network in Fig. 4.1(a), the corresponding contention graph is shown in Fig. 4.1(b). We can see that the contention graph includes three maximal cliques. They impose three constraints. For instance, clique $Q_1 = \{e_1, e_2, e_4, e_6, e_7\}$ requires that $\frac{u_1}{c_1} + \frac{u_2}{c_2} + \frac{u_4}{c_4} + \frac{u_6}{c_6} + \frac{u_7}{c_7} \leq \gamma$.

4.2.4 Link Failure

In practice, since the wireless channel between any two neighboring nodes u and v is not perfect due to environmental obstacles and background noise, each link e = (u, v) may have a probability of failure $p_e \in [0, 1]$ with which *packets* sent by node u are corrupted and *not* received by receiver node v correctly. We model the wireless channels as binary erasure channels (BEC) [99, p. 187] and assume that data packets transmitted on link e are received successfully at the receiver node with probability $1 - p_e$. The *failure probability vector* of all links in the network $\mathbf{p} = (p_e, e \in \mathcal{E})$ is assumed to be known through measurements (e.g., by probe transmissions).

Given the link failure probability vector \mathbf{p} , we can further obtain the *end-to-end* failure probability for each routed or network coded packet. For each session $i \in S$, let P_i^k denote the end-to-end reliability (i.e., 1 minus failure probability) for a packet that is *routed* over the k^{th} routing path, where $k = 1, \ldots, N_r^{(i)}$. Furthermore, for each pair of sessions $i, j \in S$, let P_{ij}^{lm} denote the end-to-end reliability for a *network coded* packet that is transmitted over network coding subgraph G_{ij}^{lm} . The aggregate *receiving* rate at destination t_i is obtained as

$$\sum_{k=1}^{N_r^{(i)}} \alpha_i^k P_i^k + \sum_{j \in \mathcal{S}, j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{lj}^{lm} P_{lj}^{lm}.$$
(4.4)

We will discuss in detail how the end-to-end reliability can be calculated in Section 4.3.

4.2.5 Network Utility Maximization Formulation

Let us concatenate all sending rates α_i^k and α_{ij}^{lm} for all $i, j \in S$, all $k = 1, \ldots, N_r^{(i)}$, all $l = 1, \ldots, N_{nc}^{(i,j)}$, and all $m = 1, \ldots, N_{nc}^{(j,i)}$ and denote the resulting vector as $\boldsymbol{\alpha}$. The network utility maximization problem for unreliable wireless networks with inter-session network coding among multiple unicast sessions can be formulated as

where u_e for each $e \in \mathcal{E}$ is as in (4.2), and for each session $i \in \mathcal{S}$, $U_i(\cdot)$ denotes a *strictly* concave and increasing utility function. For example, utility functions can be logarithmic. In that case, the utility maximization problem (4.5) becomes a proportionally fair resource allocation problem [100]. Also notice that if $U_i(x) = x$, then problem (4.5) reduces to a *throughput maximization* problem.

4.3 End-to-End Reliability

Recall from Section 4.2.4 that the aggregate receiving rate of data at receiver node t_i for each $i \in \mathcal{S}$ is as in (4.4), where P_i^k and P_{ij}^{lm} are end-to-end reliabilities for all routing and network coding paths for session *i*. In this section, we develop an algorithm to obtain these end-to-end reliability measures.

For each session $i \in S$, consider the k^{th} routing path in graph $G_r^{(i)}$ for $k = 1, \ldots, N_r^{(i)}$. The probability that data is transmitted successfully along this path can be obtained as

$$P_i^k = \prod_{e \in \mathcal{E}_r} \epsilon_i^{ek} \left(1 - p_e \right). \tag{4.6}$$

For the network in Fig. 4.1(a), we have $P_1^1 = (1 - p_{e_1})(1 - p_{e_4})(1 - p_{e_7})$ and $P_2^1 = (1 - p_{e_2})(1 - p_{e_4})(1 - p_{e_6})$.

Obtaining the end-to-end reliability of pairwise inter-session network coding paths is more complex due to the overlapping among different paths and the fact that an encoded packet is successfully received only if the corresponding remedy data is also received successfully. To explain our model, let us consider the example network in Fig. 4.1(a). Node t_1 can successfully receive some data in a network coding setting if and only if *all* of the following three conditions hold: (a) Intermediate node v_1 successfully receives the data packets from both source nodes s_1 and s_2 over links e_1 and e_2 , respectively. This happens with probability $(1 - p_{e_1})(1 - p_{e_2})$. (b) The encoded packet is successfully received by node t_1 over links e_4 and e_7 . This happens with probability $(1 - p_{e_4})(1 - p_{e_7})$. (c) The remedy packet, corresponding to the data packet, is successfully received by node t_1 over link e_5 . This happens with probability $(1 - p_{e_5})$. Therefore, we have

$$P_{12}^{11} = (1 - p_{e_5})(1 - p_{e_4})(1 - p_{e_7})(1 - p_{e_1})(1 - p_{e_2}).$$
(4.7)

Similarly, we can show that

$$P_{21}^{11} = (1 - p_{e_3})(1 - p_{e_4})(1 - p_{e_6})(1 - p_{e_1})(1 - p_{e_2}).$$
(4.8)

To generalize the idea to an arbitrary network, recall that for each pair $i, j \in S$, any $m = 1, \ldots, N_{nc}^{(i,j)}$, and any $l = 1, \ldots, N_{nc}^{(j,i)}$, inter-session network coding subgraph G_{ij}^{lm} is constructed as the union of the m^{th} triple path in $\mathcal{P}_{nc}^{(i,j)}$ and the l^{th} triple path in $\mathcal{P}_{nc}^{(j,i)}$. Given G_{ij}^{lm} , let φ_v denote the probability that node $v \in \mathcal{V}_{ij}^{lm}$ receives an original/encoded/remedy packet correctly. For the simplicity of exposition, we define $\varphi_{s_i} = \varphi_{s_j} = 1$, since the source nodes s_i and s_j have the correct data with probability one. For the network in Fig. 4.1(a), $\varphi_{v_1} = (1 - p_{e_1}) \times \varphi_{s_1} (1 - p_{e_2}) \varphi_{s_2}, \varphi_{v_2} = (1 - p_{e_4}) \varphi_{v_1}$, and $\varphi_{t_1} = (1 - p_{e_7}) \varphi_{v_2} \times (1 - p_{e_5}) \varphi_{s_2}$. We can see that there is a recursive relationship on failure probabilities of neighboring nodes. This observation motivates our end-to-end reliability calculation for inter-session network coding algorithm in Algorithm 4.1. For each node $v \in \mathcal{V}_{ij}^{lm}$, let in(v) denote the set of in-neighbors (i.e., neighbors with incoming links) of node v in graph G_{ij}^{lm} . Notice that $in(v) \subseteq \mathcal{V}_{ij}^{lm} \setminus \{v\}$. For each node $u \in in(v)$, the directed wireless link from node u to node v is denoted by e = (u, v). In this case,

$$\varphi_v = \prod_{u \in in(v)} (1 - p_{e=(u,v)})\varphi_u.$$
(4.9)

Algorithm 4.1 can be used to obtain end-to-end reliability for *any* arbitrary inter-session network coding subgraph.

Algorithm 4.1 End-to-end reliability calculation for inter-session network coding between sessions $i \in \mathcal{S}$ and $j \in \mathcal{S} \setminus \{i\}$, executed for any $l = 1, \ldots, N_{nc}^{(i,j)}$ and $m = 1, \ldots, N_{nc}^{(j,i)}$. 1: Set $\varphi_{s_i} = \varphi_{s_j} = 1$, $\varphi_v = -1$ for all $v \in \mathcal{V}_{ij}^{lm} \setminus \{s_i, s_j\}$.

- 2: while $\varphi_{t_i} = -1$ do
- Find node $v \in \mathcal{V}_{ij}^{lm} \setminus \{s_i, s_j\}$ such that $\varphi_v = -1$ and $\varphi_u \neq -1$ for all neighboring 3: nodes $u \in in(v)$.
- Set $\varphi_v = \prod_{u \in in(v)} (1 p_{e=(u,v)}) \varphi_u$. 4:
- 5: end while
- 6: $P_{ij}^{lm} = \varphi_{t_i}$

Reliability-based Rate Allocation 4.4

In this section, we propose a distributed rate allocation algorithm to solve the network utility maximization problem in (4.5). We define $F_{ij}^{elm} = \frac{1}{2} \max\{H_{ij}^{el}, H_{ji}^{em}\}$ for notational simplicity. In that case, for each clique $Q_n \in \mathcal{Q}$, where $n = 1, \ldots, |\mathcal{Q}|$, the clique constraint in (4.3) becomes

$$\sum_{e \in Q_n} \sum_{i \in \mathcal{S}} \left(\sum_{k=1}^{N_r^{(i)}} \frac{\epsilon_i^{ek}}{c_e} \alpha_i^k + \sum_{j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \frac{F_{ij}^{elm}}{c_e} \alpha_{ij}^{lm} \right) \le \gamma.$$

$$(4.10)$$

Next, we notice that although the objective function in problem (4.5) is concave, it is not strictly concave due to the linear terms inside each utility function. Thus, problem (4.5) may have *multiple* optimal solutions. This can pose some difficulties if we require a *distributed* scheme to solve the optimization problem at hand. To overcome this problem, we use the *proximal method* [101]: we add some extra terms to the objective function to

make it strictly concave. Then problem (4.5) becomes

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}\succeq\mathbf{0}} \sum_{i\in\mathcal{S}} U_{i} \left(\sum_{k=1}^{N_{r}^{(i)}} \alpha_{i}^{k} P_{i}^{k} + \sum_{j\neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{ij}^{lm} P_{ij}^{lm} \right) \\
- \sum_{i\in\mathcal{S}} \sum_{k=1}^{N_{r}^{(i)}} \frac{a_{i}}{2} (\alpha_{i}^{k} - \beta_{i}^{k})^{2} - \sum_{i\in\mathcal{S}} \sum_{j\neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \frac{b_{i}}{2} (\alpha_{ij}^{lm} - \beta_{ij}^{lm})^{2} \\
\text{subject to} \quad \sum_{e\in\mathcal{Q}_{n}} \sum_{i\in\mathcal{S}} \left(\sum_{k=1}^{N_{r}^{(i)}} \frac{\epsilon_{i}^{ek}}{c_{e}} \alpha_{i}^{k} + \sum_{j>i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \frac{F_{ij}^{elm}}{c_{e}} \alpha_{ij}^{lm} \right) \leq \gamma, \quad \forall \ Q_{n} \in \mathcal{Q} \\
\alpha_{ij}^{lm} = \alpha_{ji}^{ml}, \quad \forall \ i, j \in \mathcal{S}, i \neq j \\
l = 1, \dots, N_{nc}^{(i,j)}, \ m = 1, \dots, N_{nc}^{(j,i)}, \\
\end{cases}$$
(4.11)

where β_i^k and β_{ij}^{lm} are *auxiliary* variables introduced for all $i, j \in S, l = 1, \ldots, N_{nc}^{(i,j)}$, and any $m = 1, \ldots, N_{nc}^{(j,i)}$. On the other hand, a_i and b_i are arbitrary positive coefficients. Notice that if $\beta_i^k = \alpha_i^k$ and $\beta_{ij}^{lm} = \alpha_{ij}^{lm}$, then the objective function in problem (4.11) reduces to the original objective function in problem (4.5). For notational simplicity, we concatenate all β_i^k and β_{ij}^{lm} for all $i, j \in S, k = 1, \ldots, N_r^{(i)}, l = 1, \ldots, N_{nc}^{(i,j)}$, and $m = 1, \ldots, N_{nc}^{(j,i)}$, and denote the resulting vector by $\boldsymbol{\beta}$.

To solve the *modified* problem (4.11) via a *distributed* scheme, we use duality theory [81] and obtain the dual Lagrangian function as

$$L(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\beta}) = \sum_{i \in \mathcal{S}} B_i(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\beta}) + \sum_{Q_n \in \mathcal{Q}} \lambda_n \gamma$$
(4.12)

where

$$B_{i}(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\beta}) = U_{i} \left(\sum_{k=1}^{N_{r}^{(i)}} \alpha_{i}^{k} P_{i}^{k} + \sum_{j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{ij}^{lm} P_{ij}^{lm} \right) - \frac{a_{i}}{2} \sum_{k=1}^{N_{r}^{(i)}} (\alpha_{i}^{k} - \beta_{i}^{k})^{2} - \frac{b_{i}}{2} \sum_{j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} (\alpha_{ij}^{lm} - \beta_{ij}^{lm})^{2} - \sum_{Q_{n} \in \mathcal{Q}} \sum_{e \in Q_{n}} \sum_{k=1}^{N_{r}^{(i)}} \frac{\epsilon_{i}^{ek}}{c_{e}} \alpha_{i}^{k} \lambda_{n} - \sum_{Q_{n} \in \mathcal{Q}} \sum_{e \in Q_{n}} \sum_{j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{ij}^{lm} \frac{F_{ij}^{elm}}{c_{e}} \lambda_{n} - \sum_{j > i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \nu_{ij}^{lm} \alpha_{ij}^{lm} + \sum_{j < i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \nu_{ji}^{ml} \alpha_{ij}^{lm},$$

and λ and ν are vectors of Lagrange multipliers corresponding to clique capacity constraints and equality constraints, respectively. Notice that for each clique constraint $\sum_{e \in Q_n} \frac{u_e}{c_e} \leq \gamma$, where $n = 1, \ldots, |\mathcal{Q}|$, the corresponding Lagrange multiplier is denoted by λ_n . On the other hand, for each equality constraint $\alpha_{ij}^{lm} = \alpha_{ji}^{ml}$, where $i, j \in \mathcal{S}, l = 1, \ldots, N_{nc}^{(i,j)}$, and $m = 1, \ldots, N_{nc}^{(j,i)}$, the corresponding Lagrange multiplier is denoted by ν_{ij}^{lm} . The dual problem of the primal problem in (4.11) can be obtained as

$$\min_{\boldsymbol{\lambda} \succeq \mathbf{0}, \boldsymbol{\nu}} \Gamma(\boldsymbol{\lambda}, \boldsymbol{\nu}), \tag{4.13}$$

where $\Gamma(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \max_{\boldsymbol{\alpha} \succeq 0} L(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\beta})$. In dual problem (4.13), the variables are the Lagrange multipliers. Problem (4.13) can be solved using the gradient projection method [81]. Notice that since primal problem (4.11) is convex and its constraints are linear, strong duality holds [81, p. 226]. Thus, by solving the dual problem (4.13), the optimal solution of the primal problem (4.11) is readily obtained. However, we want to solve the original network utility maximization problem (4.5). We now explain how the solution to the problem

(4.5) can be obtained.

Our proposed distributed algorithm to identify the optimal data rates includes *two* sub-algorithms which are executed *iteratively* and *alternatively*. The first iterative sub-algorithm is based on the gradient method which is executed in shorter intervals. This sub-algorithm is used to update the dual variables λ and ν and the primal variables α . On the other hand, the second iterative sub-algorithm is based on the *proximal method* which is executed in *larger* intervals. This sub-algorithm is used to update the *auxiliary* variables β . In our iterative distributed algorithm, the first sub-algorithm forms the *inner loop*, while the second sub-algorithm forms the *outer loop*.

First Sub-algorithm

At each time t = 1, ..., T, where T denotes the network operation time, source node s_i for each session $i \in S$ updates its data rates individually as

$$\boldsymbol{\alpha}_{i}(t+1) = \operatorname{argmax}_{\boldsymbol{\alpha}_{i} \succ \mathbf{0}} B_{i}(\boldsymbol{\alpha}_{i}, \boldsymbol{\lambda}(t), \boldsymbol{\nu}(t), \boldsymbol{\beta}_{i}(t))$$
(4.14)

where α_i denotes the vector of all transmission rates of session *i*. Given the new data rates, then for each *n* such that $Q_n \in \mathcal{Q}$, for $n = 1, \ldots, |\mathcal{Q}|$, dual variable λ_n is updated as

$$\lambda_{n}(t+1) = \left[\lambda_{n}(t) + \delta_{n} \left(\sum_{i \in \mathcal{S}} \sum_{e \in Q_{n}} \left(\sum_{k=1}^{N_{r}^{(i)}} \frac{\epsilon_{i}^{ek}}{c_{e}} \alpha_{i}^{k}(t+1) + \sum_{j \neq i} \sum_{l=1}^{N_{nc}^{(i,j)}} \sum_{m=1}^{N_{nc}^{(j,i)}} \alpha_{ij}^{lm}(t+1) \frac{F_{ij}^{elm}}{c_{e}} \right) - \gamma \right]^{+},$$
(4.15)

where δ_n is a small constant step size. Furthermore, for each $i, j \in S$, such that j > i, and for any $l = 1, \ldots, N_{nc}^{(i,j)}$ and any $m = 1, \ldots, N_{nc}^{(j,i)}$, the *dual* variable ν_{ij}^{lm} is updated as

$$\nu_{ij}^{lm}(t+1) = \nu_{ij}^{lm}(t) + \sigma_{ij}^{lm}(\alpha_{ij}^{lm}(t+1) - \alpha_{ji}^{ml}(t+1)), \qquad (4.16)$$

where σ_{ij}^{lm} is a small constant step size. Notice that the update equations in (4.15) and (4.16) are based on applying the gradient method to convex problem (4.13). The convergence of (4.14) in our first sub-algorithm follows from the *descent lemma* [102, p. 639]. In particular, we can show that the first sub-algorithm converges if the following *sufficient* condition holds:

$$\left[\sum_{n:Q_n\in\mathcal{Q}}\sum_{e\in\mathcal{Q}_n} \left(\sum_{i\in\mathcal{S}}\sum_{k=1}^{N_r^{(i)}} \epsilon_i^{ek} + \sum_{i\in\mathcal{S}}\sum_{j\neq i}\sum_{l=1}^{N_{nc}^{(i,j)}}\sum_{m=1}^{N_{nc}^{(j,i)}} (F_{ij}^{elm})^2\right)\right] \max_{n:Q_n\in\mathcal{Q}} \delta_n + \max_{i,j>i,l,m} \sigma_{ij}^{lm}$$

$$< 2\min\{a_1,\ldots,a_S,b_1,\ldots,b_S\}.$$
(4.17)

The key idea is to show that our proposed (joint) distributed algorithm forms a *pseudo*contraction mapping [102, p. 182]. Details are omitted here for brevity. We notice that for any arbitrary network topology and any arbitrary choice of system parameters, we can always select step sizes δ_n and σ_{ij}^{lm} small enough such that the strict inequality in (4.17) holds.

Second Sub-algorithm

At larger intervals, i.e., large enough such that the first sub-algorithm converges within each interval, the second sub-algorithm simply sets

$$\boldsymbol{\beta}(t+1) = \boldsymbol{\alpha}(t). \tag{4.18}$$

After that, the first sub-algorithm is again triggered, but this time based on the new values of β . We continue alternate operation of the two sub-algorithms until the joint algorithm converges. The convergence is always guaranteed [102, p.232].

Next, we show that the joint algorithm formed by update equations (4.14)-(4.18) results in *optimal* rate allocation. In this regard, we notice that after convergence of the algorithm, we would have $\beta = \alpha$. In that case, the objective value in problem (4.11) reduces to the objective value in problem (4.5) as all the additional terms will become zero. Thus, the obtained data rates, which form the optimal solution to the primal problem (4.11), are also the optimal data rates with respect to the original network utility maximization problem (4.5). Similar to other distributed algorithms in which the problem is decomposed and solved by different nodes, our approach also needs message passing among different nodes.

4.5 Performance Evaluation

In this section, we assess the performance of our proposed reliability-based rate allocation algorithm and compare it with the rate allocation algorithm in [50] which does *not* consider link reliability. In particular, we evaluate how the performance improves if we take into account reliability information. In this regard, we simulate ten different randomly generated wireless topologies. Each topology is randomly selected to include between 10 to 15 wireless nodes. Since, our proposed scheme involves solving a *convex* optimization problem, it has polynomial complexity and can be implemented in practical systems and larger topologies.

In the first experiment, we assume that the utility functions are selected to maximize the network throughput. That is, $U_i(x) = x$ for all $i \in S$. In each topology, one or two links are selected randomly as *unreliable links*. The failure probability of unreliable links is chosen to be 0.5. This implies that *half* of the packets transmitted over the unreliable



Figure 4.2: Aggregate network throughput in all 10 simulated topologies with and without reliability consideration when the utility functions are selected as $U_i(x) = x$ for all $i \in S$. Taking reliability information into account increases the throughput by 36.2% on average, among all ten scenarios.

links experience transmission errors and are not received correctly. Simulation results are shown in Fig. 4.2. We can see that reliability considerations can significantly increase the network throughput. Notice that the exact performance gain differs among the topologies. This particularly depends on which link is selected as unreliable link in each topology. For example, the performance is improved by more than 100% for the case of topology 8 while the performance gain is negligible for topology 7. For the latter case, the low performance gain is due to the fact that the selected unreliable link does *not* carry any significant traffic load in an optimal design, even if it is *assumed* to be reliable. Therefore, its unreliability does not affect the network performance significantly. On average, among all 10 simulated topologies, accounting for reliability information increases the throughput by 36.2%.

Next, we study the impact of changes in the link failure probability. We assume that the failure probability of the unreliable links vary from 0 to 0.5. The former case indicates having reliable links, while the latter case indicates having unreliable links which lose *half* of the packets. Regarding the choice of utility functions, we consider the two important cases of maximizing the throughput and achieving proportional fairness. For the latter



Figure 4.3: Comparison between our algorithm (with reliability consideration) and the rate allocation algorithm which does *not* use link reliability information: (a) Maximizing the aggregate network throughput. (b) Maximizing the aggregate network utility when the utility functions are logarithmic.

case, the utility functions are selected to be *logarithmic*. Simulation results for the case of topology number 1 are shown in Fig. 4.3. Results for other topologies are similar. We can see that as the link failure probability increases, it becomes more important to take into account the reliability information. From the results in Fig. 4.3(a), if the link failure probability is 0.5, then reliability consideration can increase the throughput by 50%. From the results in Fig. 4.3(b), if the link failure probability is 0.5, then reliability consideration can increase the throughput by 50%. From the results in Fig. 4.3(b), if the link failure probability is 0.5, then reliability consideration can increase the network utility by 18.4%.

Finally, we assess the impact of the choice of utility function on the achieved performance gain. To this end, we consider the following important class of utility functions



Figure 4.4: The impact of changing the utility parameter ψ on the achieved aggregate network utility with and without reliability consideration.

[100]:

$$U_{i}(x) = \begin{cases} \log x, & \psi = 1, \\ \frac{x^{1-\psi}}{1-\psi}, & \psi \neq 1, \end{cases}$$
(4.19)

where $\psi > 0$ is a utility parameter. Simulation results for the case of the first topology when ψ varies from 0 to 10, and the link failure probability is chosen to be 0.5, are shown in Fig. 4.4. Results for the remaining topologies are similar. We can see that reliability consideration always improves the performance for *any* choice of utility parameter ψ .

4.6 Summary

In this chapter, we considered the problem of allocating data rates for all sources in a wireless inter-session network coding system. We focused on a scenario where some of the wireless links are *not* reliable, i.e., have *non-zero* failure probability. Using a simple algorithm, we calculated the *end-to-end* reliability measures of all network coding paths in the network. We then formulated a network utility maximization problem, where the objective function is the sum of the utility functions of all data sessions. We also proposed a distributed iterative algorithm to solve the formulated optimization problem. We proved

the optimality and convergence of our algorithm. Simulation results showed that it is essential to consider link reliability to achieve high throughput. In our evaluations, the network throughput is increased by 36.2% on average with up to 100% performance gain for some scenarios. The proposed scheme also significantly improves aggregate network utility for various choices of utility functions.

Chapter 5

Distributed Scheduling in Multihop Wireless Networks with Maxmin Fairness Provisioning

5.1 Introduction

In Chapters 2, 3, and 4, we studied rate allocation problems with the goal of improving the network performance in terms of achieving both high throughput and fairness. While the previous results provide an upper bound for performance of the network, they do not address the achievability of the proposed rate allocation solutions through transmission scheduling policies. In this chapter, we consider transmission scheduling in unreliable wireless networks. The network is said to be *stable* if every node only has a finite number of packets queued for transmission. Stability is subject to the condition that the data transmission rates lie within the network capacity region, i.e., it is feasible to transmit all packets with bounded delay. However, wireless links have limited capacity and may interfere with each other. The variation of the link capacity and network traffic can have an impact on the stability of the network. Another aspect is that wireless links are not as reliable as wired connections, and data packets may be corrupted during transmission. Moreover, without careful resource allocation strategy, certain users may be starved for network access whereas others may receive an unfairly large share of the available network bandwidth (e.g., see [1]). This latter aspect relates to *fairness*. We propose to address such problems through a stable and decentralized scheduling mechanism that allocates resources such that wireless links do not interfere with each other and fairness is provided while maintaining a high network throughput. We shall begin with a discussion of the key ideas and highlight our main contributions.

Using a slotted notion of time, we consider link scheduling to determine the active links in each time slot. Lyapunov stability theory has been used to construct stable and optimal decentralized scheduling policies [103]. Lyapunov techniques have also been extended to provide delay guarantee [104]. They are also applicable to throughput maximization [105] and energy minimization in single hop [106] and multihop [107] networks. A utility optimal algorithm with delay consideration using the shortest paths is also developed in [108]. Throughput optimal scheduling in ad hoc networks, which is an NP-hard problem [109], has often been reduced to a rate allocation problem, which only provides an upper bound on the rates that a network can support. Near optimal scheduling algorithms for mobile ad hoc networks have been proposed in [110]. However, fairness is not considered in the above mentioned work.

Fairness provisioning has been considered in literature (see Section 1.1.1). However, fairness is provided with maximizing the summation of utility functions corresponding to the individual flows such as in [8]. Our approach is different from all of the above in that the minimum throughput of the network is directly maximized to provide maxmin notion of fairness instead of considering utility functions for different users. We achieve this by using Lyapunov stability theory and constructing virtual queues.

In this chapter, we take the advantage of two opportunities offered by multihop networks. First, we employ multiple paths for data flows from the source to the destination. Second, we utilize different channel code rates at different links to compensate for variations in link reliability.

Our work differs from the previous work in the literature in several aspects. While most of the NUM problems determine the average sending rates [38, 42] at which the objective function is maximized, we propose an efficient scheduling and link activation policy. Our work is also novel because in the related work on fair scheduling [3–8], fairness is improved by maximizing the utility function while we directly maximize the minimum throughput in the network. We will show how this improves the network performance. Maxmin fair scheduling is considered in one hop networks with multi-radio receivers [10]. Optimal maxmin transmission and forwarding rates for sensor networks are studied in [111]. A maxmin fair scheduling policy for one hop multiple-input and multiple-output (MIMO) networks is also considered [112]. Our work is different from the above as we propose a distributed scheduling policy and consider multihop networks with single radio and therefore interference effects are incorporated. In addition, the optimality is analytically proved. While some papers considered channel coding to improve the network throughput [22], no previous work has considered both multipath routing and channel coding in the joint problem of code rate assignment, flow control and decentralized scheduling to achieve maxmin fairness in wireless networks.

The main contributions of this chapter are as follows:

• We propose a decentralized online scheduling and flow control algorithm, which we call DisF, that aims to provide fairness for each flow by maximizing the minimum throughput in a multihop wireless network. We integrate the use of multipath routing and link-dependent channel code rates in our solution approach to utilize resources better and improve reliability. While we do improve link reliability by using appropriate channel coding, we also assume the use of link-level acknowledgments and
retransmissions when a node is unable to decode a packet. We demonstrate stability of the proposed algorithm by applying the Lyapunov stability theory.

- We develop an optimal centralized rate allocation method using geometric programming, which provides an upper bound on the performance of any decentralized scheduling policy.
- We study the performance of DisF and optimal centralized algorithms through simulations over multiple random topologies. We show that the DisF algorithm ensures stability, whenever feasible, and that its performance is comparable to that of the centralized optimal solution. We also compare DisF with Lyapunov stability-based algorithms, which do not consider fairness in their design. Next, we show that the use of different channel code rates can improve the performance of the network. We also compare DisF with a class of existing approaches which use utility functions to provide maxmin fairness through simulations [8].

This chapter is organized as follows. The system model is described in Section 5.2. The decentralized stable algorithm is developed in Section 5.3. The centralized approach is formulated as a geometric programming problem and it is solved in Section 5.4. In Section 5.5, the algorithm is evaluated through simulations and the chapter is summarized in Section 5.6.

5.2 System Model

We model the wireless network with a graph $G(\mathcal{N}, \mathcal{E})$, where \mathcal{N} represents the set of $N = |\mathcal{N}|$ wireless nodes and \mathcal{E} denotes the set of directed wireless links. Link $e = (m, n) \in \mathcal{E}$ connects two nodes $m, n \in \mathcal{N}$ if and only if node n is in the transmission range of node m. We use the notations e and (m, n) interchangeably. The set of data flows is denoted by \mathcal{F} and the number of data flows is denoted by $F = |\mathcal{F}|$. The set of source nodes is denoted by \mathcal{S} . Data transmission between a source $s_f \in \mathcal{S}$ and the destination d_f of flow $f \in \mathcal{F}$ can be relayed through multiple hops.

We use multipath routing for data transmission. The set \mathcal{K}_f contains $K_f = |\mathcal{K}_f|$ routing paths for flow $f \in \mathcal{F}$. For each link $e \in \mathcal{E}$, path $k \in \mathcal{K}_f$, and flow $f \in \mathcal{F}$, we define $a_e^{fk} = 1$ if link e belongs to the k^{th} routing path for flow f, and $a_e^{fk} = 0$, otherwise. For any node $n \in \mathcal{N}$, each data flow $f \in \mathcal{F}$, and any path $k \in \mathcal{K}_f$, let i_n^{fk} and $o_n^{fk} \in \mathcal{E}$ be the input and output links to and from node n on path k of flow f, respectively. Whenever the context is clear, we remove the indices n, f, k and denote the input and output links with i and o, respectively (see Fig. 5.1).

A slotted notion of time is used with time slots $t \in \{1, 2, ...\}$. We denote the value of time-varying parameters at the beginning of each time slot t with the index t. We use the same parameter without the index t to denote its average value over all time slots. At each intermediate node $n \in \mathcal{N}$, we assume a separate queue for any path $k \in \mathcal{K}_f$ of flow $f \in \mathcal{F}$. The number of data bits corresponding to path k for flow f, stored in node n is denoted as $Q_n^{fk}(t)$. We assume $Q_{d_f}^{fk}(t) = 0, \forall t, k \in \mathcal{K}_f, f \in \mathcal{F}$ since the received bits are transferred to the upper layers at the destination node d_f . We incorporate all the queue backlogs in the vector $\mathbf{Q}(t) = (Q_n^{fk}(t), \forall n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F})$.

We use link-dependent channel code rates to counter channel variations and improve network reliability. Each source node or intermediate node $n \in \mathcal{N}$ for any flow encodes data bits by adding redundant bits and transmitting the resultant codeword of length g. Hereafter, we assume each packet consists of one codeword and we use the terms packet and codeword interchangeably. We define the code rate $R_e(t)$ as the ratio of the data bits to the total transmitted bits (data plus redundant bits) on link $e \in \mathcal{E}$. We concatenate code rates for all links $e \in \mathcal{E}$ in vector $\mathbf{R}(t) = (R_e(t), \forall e \in \mathcal{E})$. The smaller the code rate



Figure 5.1: (a) Path k of flow f from node s_f to node d_f which uses node n as a relay node is shown. It is shown that solid link 1 is active $(\mu_1^{fk}(t) = 1)$ while dotted links 2 and 3 are not active $(\mu_2^{fk}(t) = \mu_3^{fk}(t) = 0)$ in that particular time slot t. Note that links 1 and 2 belong to the mentioned path $(a_1^{fk} = a_2^{fk} = 1)$ but link 3 does not $(a_3^{fk} = 0)$. (b) A relay node n is shown with its input link i and its output link o corresponding to the k^{th} path of flow f. The corresponding packets are stored in Q_n^{fk} before they are sent.

 $R_e(t)$, the greater number of redundant bits is added, and the higher the reliability is. The reliability is gained at the cost of increased network traffic. When $R_e(t)$ is equal to one, channel coding is not used on link e.

We use $R_{0e} \leq 1$ to denote the *cut-off rate* of wireless link $e \in \mathcal{E}$. The cut-off rate is a channel parameter to which the rate of the adopted coding scheme is always limited [85]. In general, R_{0e} depends on the particular modulation scheme which is being used and also the signal-to-noise ratio (SNR) in the receiver node. For example, for a binary phase shift keying (BPSK) waveform [85], we have $R_{0e} = 1 - \log_2(1 + e^{-\gamma_e})$, where γ_e denotes the SNR at the receiver node of wireless link $e \in \mathcal{E}$. When γ_e is relatively large, R_{0e} is close to 1. Given $R_e(t) \leq R_{0e}$ for $e \in \mathcal{E}$, we have

$$P_e(t) \ge 1 - 2^{-g(R_{0e} - R_e(t))},\tag{5.1}$$

where $P_e(t)$ is the probability that a codeword of length g is received correctly on link e with rate $R_e(t)$ [85, pp. 392-397]. The vector $\mathbf{P}(t) = (P_e(t), \forall e \in \mathcal{E})$ represents the successful probabilities on all links $e \in \mathcal{E}$. For the rest of this chapter, we consider the worst case in which inequality (5.1) is satisfied with equality. For each transmission on link $e \in \mathcal{E}$, we define $\rho_e(t) = 1$ if the packet is transmitted successfully and $\rho_e(t) = 0$ otherwise. We have $\rho_e(t) = 1$ with the probability of $P_e(t)$. We define $\boldsymbol{\rho}(t) = (\rho_e(t), e \in \mathcal{E})$ as the channel state at time slot t.

As mentioned above, a codeword may be corrupted with probability $1 - P_e(t)$ through a transmission on link $e \in \mathcal{E}$. The receiver at link e sends a link-level acknowledgement (ACK) to the transmitter if the packet is received correctly. The transmitter retransmits the packet if no ACK is received within a predefined time period. Retransmissions ensure that packets admitted to the network will be received at their corresponding destination nodes. This is at the cost of increased network load.

We denote the number of data bits which are admitted to the path $k \in \mathcal{K}_f$ of flow $f \in \mathcal{F}$ at the beginning of time slot t as $\alpha_f^k(t)$. The vector $\boldsymbol{\alpha}(t) = (\alpha_f^k(t), \forall k \in \mathcal{K}_f, f \in \mathcal{F})$. Suppose all admissions are upper bounded (i.e., $\alpha_f^k(t) \leq \alpha_{max}$). We assume that all source nodes are backlogged (i.e., each source node has at least α_{max} data bits available to send over each of its routing paths at any time slot). We define the capacity region $\boldsymbol{\Lambda}$ as the closure of the set of all sending rate vectors $\boldsymbol{\alpha}$ (considering all possible routing and scheduling policies), for which the network is stable, that is

$$\mathbf{\Lambda} = \left\{ \boldsymbol{\alpha} \mid \boldsymbol{\alpha} \succeq \mathbf{0}, \lim_{t \to \infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{k \in \mathcal{K}_f, f \in \mathcal{F}, n \in \mathcal{N}} E\{Q_n^{fk}(\tau)\} < M \right\},$$
(5.2)

where M is a finite number. Note that $\boldsymbol{\alpha} = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \alpha(\tau)$ is the time average value of $\boldsymbol{\alpha}(t)$.

Two links $e_1, e_2 \in \mathcal{E}$ mutually interfere with each other if and only if the receiver of one link is in the transmission range of the sender of the other. At each time slot t, only one wireless link may be active among those wireless links which are in mutual interference with each other. We define $\mu_e^{fk}(t) = 1$ if link e is active in data transmission for the k^{th} routing path of flow f at time slot t, and $\mu_e^{fk}(t) = 0$ otherwise.

We define c_e as the number of bits that can be transmitted by link $e \in \mathcal{E}$ in each time slot t. c_e contains data bits as well as redundant bits due to channel coding. A small section of the modeled network is depicted in Fig. 5.1.

5.3 Decentralized and Stable Scheduling

In this section, we tackle the problem of online flow control and scheduling for wireless links. We address maxmin fairness provisioning by maximizing the minimum throughput in the network. Our goal is to solve the following problem

maximize
$$\min_{f \in \mathcal{F}} \alpha_f$$
 (5.3)
subject to $\boldsymbol{\alpha} \in \boldsymbol{\Lambda}$.

The goal in problem (5.3) is to admit new packets and schedule the transmissions such that the minimum sending rate $\alpha_f = \sum_{k \in \mathcal{K}_f} \alpha_f^k$ over all flows $f \in \mathcal{F}$ is maximized and all queues in the network remain stable, that is the number of bits stored in any queue is bounded. Note that data bits are removed from the queue of the sender node only after it has received an ACK from the receiver. Therefore, if the queues are stable, the sending rate of each flow is the same as its throughput at the corresponding destination.

To enhance the minimum throughput of the network, we need to introduce a decision parameter $\lambda(t)$ and a set of *virtual queues* $Z_f(t), \forall f \in \mathcal{F}$. We denote $\mathbf{Z}(t) = (Z_f(t), \forall f \in \mathcal{F})$ \mathcal{F}). For each virtual queue $Z_f(t)$ for flow f at each time slot t, we set $\sum_{k \in \mathcal{K}_f} \alpha_f^k(t)$ as the service rate and $\lambda(t)$ as the input rate. Then, we have the following update equation:

$$Z_f(t+1) \le \max\left[Z_f(t) - \sum_{k \in \mathcal{K}_f} \alpha_f^k(t), 0\right] + \lambda(t).$$
(5.4)

Suppose $\lambda(t)$ is upper bounded (i.e., $\lambda(t) \leq \lambda_{max}$ for any time slot t) and its time average $\lambda = \lim_{t \to \infty} \frac{\sum_{\tau=0}^{t-1} E\{\lambda(\tau)\}}{t}$ exists. We will show later that burstiness in the network increases when λ_{max} increases. The stability of each virtual queue Z_f implies that the time average of its input rate is less than or equal to that of its service rate. That is

$$\lambda \le \sum_{k \in \mathcal{K}_f} \alpha_f^k, \tag{5.5}$$

where $\alpha_f^k = \lim_{t\to\infty} \frac{\sum_{\tau=0}^{t-1} E\{\alpha_f^k(\tau)\}}{t}$ is the time average value of $\alpha_f^k(t)$. Therefore, if all virtual queues are stable, maximizing the time average value of $\lambda(t)$ is equivalent to maximizing the minimum throughput among all data flows in the network. The goal is to maximize the time average value of $\lambda(t)$ such that both real queues (which store the data bits) and virtual queues remain stable.

We now present some aspects of Lyapunov stability theory [103] that are useful for developing our scheduling algorithm. Let Lyapunov function $L(\Theta(t))$ be a non-negative function of any queue vector $\Theta(t)$. We define the Lyapunov drift $\Delta(\Theta(t)) \triangleq E\{L(\Theta(t + 1)) - L(\Theta(t)) \mid \Theta(t)\}$.

Proposition 5.1 (Lyapunov Optimization [103]) Let u(t) be a utility function and B > 0, $\epsilon > 0$, and V > 0 be constants such that for all time slots t and queue vector $\Theta(t) =$ $(\Theta_q(t) \mid q \in \Pi = \{1, 2, \dots, |\Pi|\}), we have$

$$\Delta(\boldsymbol{\Theta}(t)) - VE\{u(t) \mid \boldsymbol{\Theta}(t)\} \le B - \epsilon \sum_{q \in \Pi} \Theta_q(t) - Vu^*, \tag{5.6}$$

where u^* is a target value for utility function u(t), then we have

$$u_{inf} \ge u^* - B/V,$$

$$\lim_{t \to \infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{q \in \Pi} E\{\Theta_q(\tau)\} \leq \frac{B + V(u_{sup} - u^*)}{\epsilon},$$

where $u_{inf} = \lim_{t \to \infty} \inf \frac{1}{t} \sum_{\tau=0}^{t-1} E\{u(\tau)\}$ and $u_{sup} = \lim_{t \to \infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} E\{u(\tau)\}.$

The proof of the proposition can be found in [103, pp. 82-84]. Note that the expectation is over random parameters such as channel states and possibly randomized scheduling policies.

Let $u(t) = \lambda(t)$. We concatenate the backlog queues and virtual queues in the vector $\Theta(t) = (\mathbf{Q}(t), \mathbf{Z}(t))$. Proposition 5.1 states that if condition (5.6) holds under a scheduling algorithm, then all the queues in $\Theta(t)$ are stable and λ will be at most B/V away from the target value λ^* . Stability of virtual queues \mathbf{Z} ensures that λ is always less than or equal to the minimum throughput of the network. By increasing V, we can get closer to the target value at the cost of a linear increase in the congestion in the network. Next, we obtain $\Delta(\Theta(t))$ for any time slot t. We define

$$L(\boldsymbol{\Theta}(t)) = \sum_{n \in \mathcal{N}, \ k \in \mathcal{K}_f, \ f \in \mathcal{F}} \frac{Q_n^{fk}(t)^2}{2} + \sum_{f \in \mathcal{F}} \frac{Z_f^2(t)}{2}.$$
(5.7)

We assume that scheduled transmissions occur at the beginning of each time slot. For an

intermediate relay node $n \in \mathcal{N}, n \neq s_f$, any path $k \in \mathcal{K}_f$ and flow $f \in \mathcal{F}$, we have

$$Q_{n}^{fk}(t+1) \leq Q_{n}^{fk}(t) - \min[Q_{n}^{fk}(t), c_{o}R_{o}(t)]\mu_{o}^{fk}(t)\rho_{o}(t) + c_{i}R_{i}(t)\mu_{i}^{fk}(t)\rho_{i}(t)$$

$$= \max[Q_{n}^{fk}(t) - c_{o}R_{o}(t)\mu_{o}^{fk}(t)\rho_{o}(t), Q_{n}^{fk}(t)(1-\mu_{o}^{fk}(t)\rho_{o}(t))] + c_{i}R_{i}(t)\mu_{i}^{fk}(t)\rho_{i}(t).$$
(5.8)

For source node $s_f \in \mathcal{S}, f \in \mathcal{F}$ and $k \in \mathcal{K}_f$, we have

$$Q_{s_{f}}^{fk}(t+1) \leq Q_{s_{f}}^{fk}(t) - \min[Q_{s_{f}}^{fk}(t), c_{o}R_{o}(t)]\mu_{o}^{fk}(t)\rho_{o}(t) + \alpha_{f}^{k}(t) = \max[Q_{s_{f}}^{fk}(t) - c_{o}R_{o}(t)\mu_{o}^{fk}(t)\rho_{o}(t), Q_{s_{f}}^{fk}(t)(1-\mu_{o}^{fk}(t)\rho_{o}(t))] + \alpha_{f}^{k}(t).$$
(5.9)

Lemma 5.1 For any $\rho \in \{0, 1\}$, U, R, and $\mu \in \{0, 1\}$ we have

$$\max[U - R\mu\rho, U(1 - \mu\rho)] \leq \max[U - R\mu, 0] + R\mu(1 - \rho).$$
(5.10)

Proof Let μ be equal to one. We verify the inequality in both cases when $U \ge R$ and when U < R separately. If $U \ge R$, then we have $U(1 - \rho) < U - R\rho$ and both sides of (5.10) are equal to $U - R\rho$. On the other hand, if we have U < R, the left hand side of (5.10) is $U(1 - \rho)$ and the right hand side is $R(1 - \rho)$ and the inequality is verified. In the case where $\mu = 0$, inequality (5.10) states that $U \le \max[U, 0]$, which is true.

Considering (5.8) and using Lemma 5.1, for a relay node $n \in \mathcal{N}$, $n \neq s_f$, $k \in \mathcal{K}_f$, and $f \in \mathcal{F}$, we have

$$Q_n^{fk}(t+1) \le \max[Q_n^{fk}(t) - c_o R_o(t)\mu_o^{fk}(t), 0] + c_o R_o(t)\mu_o^{fk}(t)(1 - \rho_o(t)) + c_i R_i(t)\mu_i^{fk}(t)\rho_i(t).$$
(5.11)

Considering (5.9) and Lemma 5.1, for any source node $s_f \in \mathcal{S}, k \in \mathcal{K}_f, f \in \mathcal{F}$, we have

$$Q_{s_f}^{fk}(t+1) \leq \max[Q_{s_f}^{fk}(t) - c_o R_o(t) \mu_o^{fk}(t), 0] + c_o R_o(t) \mu_o^{fk}(t) (1 - \rho_o(t)) + \alpha_f^k(t).$$
(5.12)

We now introduce two lemmas to simplify (5.11) and (5.12). Lemma 5.2 can be found in [103].

Lemma 5.2 ([103]) For any positive U_1 , U_2 , η , and ν , if we have $U_1 \leq \max[U_2 - \eta, 0] + \nu$, then

$$U_1^2 \le U_2^2 + \eta^2 + \nu^2 - 2U_2(\eta - \nu).$$
(5.13)

Proof If $U_2 > \eta$, then we have $U_1 \le U_2 - \eta + \nu$. By squaring both sides of this inequality and adding the positive value of $2\nu\eta$ to the right hand side of the inequality, (5.13) is verified. If $U_2 < \eta$, then we have $U_1^2 \le \nu^2$. Since $\eta^2 + U_2^2 - 2U_2(\eta - \nu) \ge 0$, we can add it to the right hand side of the above and inequality (5.13) is verified.

Lemma 5.3 For positive U_1 , U_2 , O, I, ρ , and $\rho' \leq 1$ if $U_1 \leq \max[U_2 - O, 0] + O(1 - \rho') + I\rho$, then

$$U_1^2 \le U_2^2 + B - 2U_2(O\rho' - I\rho),$$

where $B = O^2 + \rho^2 I^2 + O^2 (1 - \rho')^2 + 2\rho (1 - \rho') OI$.

Proof We proved in Lemma 5.2 that for positive values of U_1 , U_2 , η , and ν if $U_1 \leq \max[U_2 - \eta, 0] + \nu$, then we have $U_1^2 \leq U_2^2 + \eta^2 + \nu^2 - 2U_2(\eta - \nu)$. By substituting $\eta = O$ and $\nu = O(1 - \rho') + I\rho$, Lemma 5.3 is proven.

Using Lemma 5.3 and inequalities (5.11) and (5.12), for any $k \in \mathcal{K}_f$ and $f \in \mathcal{F}$, we

have

$$Q_n^{fk}(t+1)^2 \le Q_n^{fk}(t)^2 + B_n^{fk}(t) - 2Q_n^{fk}(t)(c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - c_i R_i(t)\mu_i^{fk}(t)\rho_i(t)),$$
(5.14)

for any intermediate relay node $n \in \mathcal{N}$ $(n \neq s_f)$ where

$$B_n^{fk}(t) = c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 + \rho_i(t)^2 c_i^2 R_i(t)^2 \mu_i^{fk}(t)^2 + c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 (1 - \rho_o(t))^2 + 2\rho_i(t)(1 - \rho_o(t)) c_o R_o(t) \mu_o^{fk}(t) c_i R_i(t) \mu_i^{fk}(t).$$

Similarly, for source node s_f , we have

$$Q_{s_f}^{fk}(t+1)^2 \le Q_{s_f}^{fk}(t)^2 + B_{s_f}^{fk} - 2Q_{s_f}^{fk}(t)(c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - \alpha_f^k(t)),$$
(5.15)

where

$$B_{s_f}^{fk}(t) = c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 + \alpha_f^k(t)^2 + c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 (1 - \rho_o(t))^2 + 2(1 - \rho_o(t)) c_o R_o(t) \mu_o^{fk}(t) \alpha_f^k(t).$$

From Lemma 5.2 and inequality (5.4), for each virtual queue Z_f , $f \in \mathcal{F}$, we have

$$Z_f^2(t+1) \leq Z_f^2(t) + \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t)\right)^2 + \lambda^2(t) - 2Z_f(t) \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) - \lambda(t)\right).$$
(5.16)

Now, we can write $\triangle \Theta(t) - VE\{\lambda(t) \mid \Theta(t)\}$ as

$$\triangle(\boldsymbol{\Theta}(t)) - VE\{\lambda(t) \mid \boldsymbol{\Theta}(t)\} = E\{L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) \mid \boldsymbol{\Theta}(t)\} - VE\{\lambda(t) \mid \boldsymbol{\Theta}(t)\}$$

$$\leq B - \sum_{k \in \mathcal{K}_{f}, f \in \mathcal{F}} \left(Q_{s_{f}}^{fk}(t) E\{c_{o}R_{o}(t)\mu_{o}^{fk}(t)\rho_{o}(t) - \alpha_{f}^{k}(t) \mid \boldsymbol{\Theta}(t) \} + \sum_{n \neq s_{f}} Q_{n}^{fk}(t) E\{c_{o}R_{o}(t)\mu_{o}^{fk}(t)\rho_{o}(t) - c_{i}R_{i}(t)\mu_{i}^{fk}(t)\rho_{i}(t) \mid \boldsymbol{\Theta}(t) \} \right)$$

$$-\sum_{f\in\mathcal{F}} Z_f(t) E\left\{\sum_{k\in\mathcal{K}_f} \alpha_f^k(t) - \lambda(t) \mid \boldsymbol{\Theta}(t)\right\} - VE\{\lambda(t) \mid \boldsymbol{\Theta}(t)\},\tag{5.17}$$

where $B = B_Q + B_Z$, and

$$B_Q = 5NF\left(\max_{f\in\mathcal{F}} K_f\right) \max\left[\alpha_{max}^2, \max_{e\in\mathcal{E}} c_e^2\right]/2,\tag{5.18}$$

$$B_Z = F\left(\left(\alpha_{max} \max_{f \in \mathcal{F}} K_f\right)^2 + \lambda_{max}^2\right)/2.$$

Inequality (5.17) holds under any adopted scheduling algorithm. Note that the expected values are taken over the random channel state probabilities.

Let λ^* be the optimal value for λ which can be achieved by an algorithm such that all backlog and virtual queues remain stable. Since the corresponding sending rates are stably supported by the network, there exist link data rates for all wireless links that support data transmission in the network. These data rates can be achieved with a possibly randomized *channel state-only* algorithm \mathcal{X} . This is proved by projection of link data rates in different time-varying channel states and then expressing each projection as the convex combination of corresponding independent sets. Further details can be found in [105]. Assume that Algorithm \mathcal{X} determines the decision parameters ($\lambda(t)$, $R_e(t)$, $\mu_e^{fk}(t)$, and $\alpha_f^k(t)$ for all $e \in \mathcal{E}, k \in \mathcal{K}_f$, and $f \in \mathcal{F}$) at the beginning of each time slot t such that λ achieves the optimal value λ^* , virtual queues are stable (i.e., $\lambda < \sum_{k \in \mathcal{K}_f} \alpha_f^k$ for any $f \in \mathcal{F}$) and backlog queues are also stable. Note that Algorithm \mathcal{X} is a *channel state-only* algorithm which makes the decisions only based on the observed channel states at each time slot. Therefore, it needs a *priori* knowledge on channel states. The stability of backlog queues implies that under Algorithm \mathcal{X} , for an intermediate relay node $n \in \mathcal{N}$ and $n \neq s_f$, we have

$$E\{c_o R_o(t) \ \mu_o^{fk}(t)\rho_o(t) \ | \ \Theta(t)\} > E\{c_i R_i(t)\mu_i^{fk}(t)\rho_i(t) \ | \ \Theta(t)\}.$$
(5.19)

For any source node $s_f, f \in \mathcal{F}$, we have

$$E\{c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) \mid \boldsymbol{\Theta}(t)\} > E\{\alpha_f^k(t) \mid \boldsymbol{\Theta}(t)\}.$$
(5.20)

The stability of virtual queues implies that

$$E\{\lambda(t) \mid \boldsymbol{\Theta}(t)\} < E\left\{\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) \mid \boldsymbol{\Theta}(t)\right\},\tag{5.21}$$

for all $f \in \mathcal{F}$. Then, from (5.17), there exists $\epsilon > 0$ such that

$$\Delta(\boldsymbol{\Theta}(t)) - VE\{\lambda(t) \mid \boldsymbol{\Theta}(t)\} \leq B - \epsilon \left(\sum_{n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F}} Q_n^{fk}(t) + \sum_{f \in \mathcal{F}} Z_f(t)\right) - V\lambda^*,$$
(5.22)

under Algorithm \mathcal{X} , where $B = B_Q + B_Z$ as in (5.18). Note that here the expectation is taken over different random channel states and different randomized decisions.

We now present the distributed fair (DisF) algorithm for maximizing the minimum throughput in a multihop network with channel coding and multipath routing. The goal of the algorithm is to select the decision parameters $\lambda(t)$, $R_e(t)$, $\alpha_f^k(t)$ and $\mu_e^{fk}(t)$ for all $e = (n_e, n'_e) \in \mathcal{E}, \ k \in \mathcal{K}_f$, and $f \in \mathcal{F}$, such that

$$\sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \alpha_f^k(t) \left(Z_f(t) - Q_{s_f}^{fk}(t) \right) + \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \sum_{e \in \mathcal{E}} a_e^{fk} c_e R_e(t) \mu_e^{fk}(t) \left(Q_{n_e}^{fk}(t) - Q_{n'_e}^{fk}(t) \right) P_e(t) + \lambda(t) \left(V - \sum_{f \in \mathcal{F}} Z_f(t) \right),$$
(5.23)

is maximized over all available decision parameters at each time slot t. Before we present the algorithm in detail, we analyze the performance of the algorithm in Theorem 5.1.

Theorem 5.1 Let Algorithm DisF be an algorithm that maximizes (5.23) over all available decision parameters at each time slot t. Then, it is throughput-optimal. That is, it stabilizes the network if any other algorithm can do so and the minimum throughput in the network is at most B/V away from the optimal value.

Proof Algorithm DisF maximizes (5.23), which can be rewritten as

$$\sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \left(Q_{s_f}^{fk}(t) \left(c_o R_o(t) \mu_o^{fk}(t) P_o(t) - \alpha_f^k(t) \right) + \sum_{n \neq s_f} Q_n^{fk}(t) \left(c_o R_o(t) \mu_o^{fk}(t) P_o(t) - c_i R_i(t) \mu_i^{fk}(t) P_i(t) \right) \right)$$

$$+ \sum_{f \in \mathcal{F}} Z_f(t) \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) - \lambda(t) \right) + V\lambda(t).$$
(5.24)

Recall from Section 5.2 that for any node $n \in \mathcal{N}$, each data flow $f \in \mathcal{F}$, and any path $k \in \mathcal{K}_f$, i_n^{fk} and $o_n^{fk} \in \mathcal{E}$ denote the input and output links to and from node n on path k of flow f, respectively. We remove the indices n, f, k and denote the input and output links with i and o in (5.24), respectively. Recall that inequality (5.17) holds under any algorithm including the DisF algorithm. By maximizing (5.24) at each time slot t, we minimize the upper bound on $\Delta(\Theta(t)) - VE\{\lambda(t) \mid \Theta(t)\}$ (right hand side of (5.17)) over

its value under any other algorithm including Algorithm \mathcal{X} , that is on the right hand side of (5.22). Then, under the DisF algorithm, we have

$$\Delta(\Theta(t)) - VE\{\lambda(t) \mid \Theta(t)\} \le B - \epsilon \sum_{n \in \mathcal{N}, \ k \in \mathcal{K}_f, \ f \in \mathcal{F}} Q_n^{fk}(t) - \epsilon \sum_{f \in \mathcal{F}} Z_f(t) - V\lambda^*.$$
(5.25)

This is the condition of Proposition 5.1.

We must note that while maximizing (5.23) over all possibilities leads to an optimal solution, it is an NP-hard problem in wireless networks due to interference constraints [113]. In order to solve this problem in a distributed manner and in reasonable time (compared to the introduced delay and channel variation time due to channel fluctuations), we use the greedy maximal scheduling (GMS) policy. GMS is sub-optimal and may be implemented in a distributed manner. It has been shown in [114] that its efficiency ratio is at least 1/2 in the 1-hop interference model. In other words, GMS can achieve at least half of the throughput achieved by the optimal policy. Algorithm 5.1 shows the DisF algorithm which aims at maximizing (5.23) at any time slot t in a decentralized manner. This algorithm has several phases that are performed at the beginning of each time slot t.

Flow Control

Each source node s_f checks the backlog queue $Q_{s_f}^{fk}(t)$ for each path $k \in \mathcal{K}_f$ of flow f. If $Q_{s_f}^{fk}(t) \leq Z_f(t)$, s_f schedules α_{max} new data bits for flow f to be admitted to the corresponding path (i.e., sets $\alpha_f^k(t)$ to be equal to α_{max}) (Lines 2-9).

Scheduling

The candidate set is initialized with all links that have data to send. Each link $e = (n, n') \in \mathcal{E}$, sets its weight w_e equal to the maximum value of $Q_n^{fk}(t) - Q_{n'}^{fk}(t)$ over all paths $k \in \mathcal{K}_f$

Algorithm 5.1 Distributed fair (DisF) algorithm for maximizing the minimum throughput among the entire users in the network, executed at the beginning of each time slot t. The algorithm is initiated with $\mathbf{Z}(0) = \mathbf{0}$ at t = 0.

1: Initialization: Assign values for α_{max} , λ_{max} , V. 2: Set $\boldsymbol{\alpha}(t) = \mathbf{0}$. 3: for each source $s_f \in \mathcal{S}, f \in \mathcal{F}$ do for each $k \in \mathcal{K}_f$ do 4: if $Q_{s_f}^{fk}(t) \leq Z_f(t)$ then s_f sets $\alpha_f^k(t) = \alpha_{max}$. 5: 6:7: end if 8: end for 9: end for 10: Initiate *candidate set* C of all links that have data to send. 11: for each link $e = (n, n') \in C$ do Set $w_e = \max_{f,k} (Q_n^{fk}(t) - Q_{n'}^{fk}(t)).$ 12:Set $(f_e^*, k_e^*) = \underset{f,k}{\arg \max}(Q_n^{fk}(t) - Q_{n'}^{fk}(t))$ 13:14: **end for** 15: while C is not empty do Set $e^* = \arg \max w_e$. 16:Set $\mu_{e^*}^{f^*_{e^*}k^*_{e^*}}(t) = 1.$ 17:Find $R_{e^*}(t)$ by solving $2^{-g(R_{0e^*}-R_{e^*}(t))} = \frac{1}{1+R_{e^*}(t)g\ln 2}$. 18:Remove e^* and all links that interfere with e^* from the candidate set C. 19: 20: end while 21: for each source $s_f \in \mathcal{S}, f \in \mathcal{F}$ do if $\sum_{f \in \mathcal{F}} Z_f(t) \leq V$ then 22:Set $\lambda(t) = \lambda_{max}$. 23: 24:else 25:Set $\lambda(t) = 0$. end if 26:27: end for 28: for each source $s_f \in \mathcal{S}, f \in \mathcal{F}$ do $Z_f(t+1) = \max \left[Z_f(t) - \sum_{k \in \mathcal{K}_f} \alpha_f^k(t), 0 \right] + \lambda(t).$ Update the other sources with $Z_f(t+1)$ through control messages. 29:30: 31: end for

and flows $f \in \mathcal{F}$ that use that link. Then the link with the maximum weight is selected to transmit data for the corresponding path and flow, it is removed from the candidate set and all links which interfere with that link are also removed from the set. This process continues until no link remains in the candidate set (Lines 10-20). The scheduling process (lines 15-20) can be implemented in a distributed manner by making modifications to the MAC parameters. This is done in [113] by varying contention window parameters CW_{min} and CW_{max} for links to transmit more or less aggressively according to their weights w_e .

Code Rate Allocation

For each scheduled link e, the optimal code rate that maximizes $R_e(t)(1 - 2^{-g(R_{0e} - R_e(t))})$ is determined by solving $2^{-g(R_{0e} - R_e(t))} = \frac{1}{1 + R_e(t)g \ln 2}$ (Line 18). In case of any change in the wireless environment, $R_e(t)$ is determined according to the new available updates for R_{0e} .

Fairness Provisioning

The source node of each data flow $f \in \mathcal{F}$ sets $\lambda(t) = \lambda_{max}$ if $\sum_{f \in \mathcal{F}} Z_f(t) \leq V$, and sets $\lambda(t) = 0$ otherwise (Lines 21-27). Virtual queues are updated according to (5.4) (Line 28-31). The value of virtual queues at each time slot is transmitted between the source nodes through control messages. Next, the scheduled links transmit their packets and new data bits are admitted in the source node queues.

5.4 Geometric Programming Formulation

In this section, we provide a benchmark for performance of the decentralized scheduling algorithm for evaluation purposes. We formulate the problem of code rate and sending rate allocation as a NUM problem and describe a centralized solution approach. We assume multiple paths are available for data flows. We also allow channel coding to improve the reliability and use retransmissions in case of any data loss. In this section, we do not consider scheduling. The slotted notion of time is not considered and the variables are time average values.

Recall that the sending rate for the k^{th} path of flow $f \in \mathcal{F}$ is α_f^k . Channel coding is performed on link $e \in \mathcal{E}$ at a rate of R_e . Thus, link e must transmit packets for path $k \in \mathcal{K}_f$ of flow $f \in \mathcal{F}$ with a rate of α_f^k/R_e if all transmissions are successful. Since the probability of a successful transmission is P_e , each transmission is completed within $1/P_e$ attempts on average where $P_e = 1 - 2^{-g(R_{0e} - R_e)}$ in the worst case. We have

$$u_e^{fk} = a_e^{fk} \frac{\alpha_f^k}{R_e P_e},\tag{5.26}$$

where u_e^{fk} is the data rate at which link e is being used to transmit data for the k^{th} path of flow $f \in \mathcal{F}$. We can express the usage of wireless link $e \in \mathcal{E}$ as

$$u_e = \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} a_e^{fk} \frac{\alpha_f^k}{R_e P_e^{fk}}.$$
(5.27)

To model the interference in the network, we use a contention graph $G_C(\mathcal{N}_C, \mathcal{E}_C)$. The set of vertices \mathcal{N}_C represents the set of wireless links in graph G. There is a link between each two vertices if and only if the corresponding links in graph G interfere with each other. Each *complete* subgraph in graph G_C is called a clique and a maximal clique ω is one that is not a subset of a larger clique. We define Ω as the set of all maximal cliques in the network. It is *necessary* for successful transmissions that the summation of link usages over all links in each maximal clique be less than the capacity of the clique ζ_{ω} . This is a necessary condition for successful transmission and leads to an *upper bound* on the network performance.

Now, we can write the problem of fair sending rate and code rate allocation in a

multihop wireless network to maximize the minimum throughput as

$$\begin{array}{ll} \underset{\alpha, \mathbf{R}, \sigma}{\operatorname{maximize}} & \sigma \\ \text{subject to} & \sigma \leq \sum_{k \in \mathcal{K}_f} \alpha_f^k, & \forall f \in \mathcal{F}, \\ & \sum_{e \in \omega} \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} a_e^{fk} \frac{\alpha_f^k}{R_e P_e} \leq \zeta_\omega, & \forall \omega \in \Omega, \\ & P_e \leq 1 - 2^{-g(R_{0e} - R_e)}, & \forall e \in \mathcal{E}, \\ & \boldsymbol{\alpha} \succ \mathbf{0}, \ \mathbf{0} \prec \mathbf{R} \prec \mathbf{R}_{\mathbf{0}}. \end{array}$$
(5.28)

In problem (5.28), the objective is to maximize the throughput of the flow with the minimum value. The second set of constraints satisfies the *necessary* condition for successful transmissions and leads to an upper bound on the performance. Since retransmissions due to packet loss are taken into account in the second set of constraints, the total number of admitted packets will be received at the destination. Therefore, by maximizing the sending rate of any flow, the received throughput is also maximized.

The objective function in problem (5.28) and the left hand side of the second set of constraints are posynomials (i.e., polynomials with positive terms). The first set of constraints has *signomials* (i.e., polynomials with *both* positive and negative terms) on the left hand side. Therefore, we can apply *signomial programming* techniques [82] to solve this problem. In this regard, we need to approximate the right hand side of the first set of constraints with a monomial around an initial point $\hat{\alpha}$. For a parameter b > 1 very close to 1, we have

$$\sum_{k \in \mathcal{K}_f} \alpha_f^k \approx \hat{\Lambda}_f^{-1} \prod_{k \in \mathcal{K}_f} \left(\frac{\alpha_f^k}{\hat{\alpha}_f^k} \right)^{\hat{\alpha}_f^k \Lambda_f}, \ \forall \ \boldsymbol{\alpha} \in \left[\hat{\boldsymbol{\alpha}} / b, b \hat{\boldsymbol{\alpha}} \right],$$
(5.29)

where $\hat{\Lambda}_{f}^{-1} = \sum_{k \in \mathcal{K}_{f}} \hat{\alpha}_{f}^{k}$. Finally, we tackle the third set of constraints in (5.28). We can approximate the exponential term using the *Taylor series expansion* and rewrite the

constraint as

$$P_e \le 1 - X_{1e} \sum_{n=0}^{\infty} \frac{(X_2 R_e)^n}{n!},$$
(5.30)

where $X_{1e} = 2^{-gR_{0e}}$ and $X_2 = g \ln 2$. For large M_e , we have

$$P_e \le 1 - X_{1e} \sum_{n=0}^{M_e - 1} \frac{(X_2 R_e)^n}{n!}.$$
(5.31)

 M_e must be large enough such that $(X_2R_e)^{M_e} \ll M_e!$ and can be found through simulations. Now, we can rewrite problem (5.28) in the standard form of geometric programming problems as

$$\begin{array}{ll} \underset{\sigma,\alpha,\mathbf{R},\mathbf{P}}{\operatorname{minimize}} & \sigma^{-1} \\ \text{subject to} & \hat{\Lambda}_{f}\sigma \prod_{k\in\mathcal{K}_{f}} (\hat{\alpha}_{f}^{k}\alpha_{f}^{k-1})^{\hat{\alpha}_{f}^{k}\hat{\Lambda}_{f}} \leq 1, & \forall f \in \mathcal{F}, \\ & \sum_{e\in\omega} \sum_{f\in\mathcal{F}} \sum_{k\in\mathcal{K}_{f}} a_{e}^{fk}\alpha_{f}^{k}R_{e}^{-1}P_{e}^{-1} \leq \zeta_{\omega}, & \forall \omega \in \Omega, \\ & \frac{P_{e}}{1-X_{1e}} + \frac{X_{1e}}{1-X_{1e}} \sum_{n=1}^{M_{e}-1} \frac{(X_{2}R_{e})^{n}}{n!} \leq 1, & \forall e \in \mathcal{E}, \\ & \hat{\alpha}/b \leq \alpha \leq b\hat{\alpha}, \\ & \sigma > 0, \ \mathbf{0} \prec \mathbf{R} \prec \mathbf{R}_{\mathbf{0}}, \ \mathbf{P} \succ \mathbf{0}. \end{array} \right.$$

$$(5.32)$$

The above problem is a geometric programming problem that can be solved iteratively using the interior-point method [82]. In each iteration, we use the solution obtained in the previous iteration as the new initial point $\hat{\alpha}$ and use the approximation in (5.29) around that point. We use the solution of this problem as a benchmark for evaluating the DisF algorithm.

5.5 Performance Evaluation

In this section, we evaluate our proposed algorithm through simulations. For simulations, we use Matlab. First, we compare the performance of the DisF algorithm with the centralized approach obtained through solving geometric programming problem. Then, we study fairness provisioning in the network. For that purpose, we quantitatively measure fairness under the DisF algorithm in several random topologies and compare it with a Lyapunovbased algorithm in which the fairness is not considered. Here, we call that algorithm as DisA algorithm. The goal in DisA algorithm is to maximize the aggregate throughput of the network (i.e., $\sum_{f,k} \alpha_f^k$) while DisF algorithm maximizes the minimum throughput in the network (i.e., $\min_f \sum_k \alpha_f^k$). We also show how fairness is provided at the cost of degrading the aggregate throughput in the network. We show the effect of channel coding on the network performance by comparing with the case when channel coding is not used. Moreover, we study the effect of algorithm parameter V on the obtained performance of the DisF algorithm. Finally, we compare the proposed approach with the distributed utility-based approach presented in [8].

We compare the minimum throughput under the DisF algorithm with the minimum throughput obtained through solving the geometric programming problem (5.28). We run the simulations for both approaches in topologies with 30 nodes. The number of flows varies between 2 and 10. In this set of simulations, we set $\alpha_{max} = 2$, $\lambda_{max} = 10$, V = 50, and $c_e = 10$ bits for all links $e \in \mathcal{E}$. This is because the MOSEK software [89] that we used to solve the geometric programming problem cannot solve the problem when c_e and consequently M_e (see (5.31)) grows. We run the simulations on 50 random topologies. The DisF algorithm follows the optimal solution as the number of flows in the network grows (Fig. 5.2). Increasing the number of flows leads to higher load on the network and causes a degradation in the minimum throughput in the network for both approaches.



Figure 5.2: The performance of the DisF algorithm is compared with the solution of centralized geometric programming problem. Each point represents the average performance obtained over 50 random topologies. The DisF algorithm follows the centralized approach as the number of flows increases.

Hereafter, we set parameters $\alpha_{max} = 1000$, $\lambda_{max} = 5000$, V = 250000, and $c_e = 5000$ bits for all links $e \in \mathcal{E}$. Next, we study the max-min fair (DisF) algorithm in regards of fairness provisioning. We run both DisF and DisA algorithms in a sample network topology (Fig. 2.2) with 20 nodes and 5 data flows. We observe that the achieved throughput, for the sample topology, is distributed fairly under the DisF algorithm while this is not the case for the DisA algorithm (see Fig. 5.3).

We also study the fairness provisioning quantitatively on several random topologies. We use the Jain's fairness index [2], to measure the fairness among network users. The fairness index $\psi = \frac{(\sum_{f \in \mathcal{F}} \alpha_f)^2}{|\mathcal{F}| \sum_{f \in \mathcal{F}} \alpha_f^2}$, where $\alpha_f = \sum_{k \in \mathcal{K}_f} \alpha_f^k$ denotes the throughput of flow $f \in \mathcal{F}$. For the results in Fig. 5.3, we have $\psi = 0.97$ under DisF algorithm while ψ decreases to 0.41 under DisA algorithm. Using the DisF algorithm in several random topologies, the fairness index is always higher than 0.95 while that of DisA algorithm degrades to 0.55 when the number of flows is equal to 10 (see Fig. 5.4).

Improving the minimum throughput in the network is at the cost of degrading the



Figure 5.3: In a sample topology, fairness is provided under the DisF algorithm while some flows starve under the DisA algorithm that does not consider fairness.



Figure 5.4: The performance of DisF and DisA algorithms in regard of fairness provisioning. Each point represents the average value of the fairness index over 50 random topologies.



Figure 5.5: The tradeoff between minimum throughput in the network (a) and aggregate throughput in the network (b) is shown under both DisF and DisA algorithms.

aggregate throughput which is achievable with the entire users in the network. Fig. 5.5 shows the tradeoff between the minimum throughput and aggregate throughput of the network. It is shown that the minimum throughput (and also the fairness) is improved via DisF algorithm (Fig. 5.5 (a)). However, that is gained at the cost of degrading the aggregate throughput of the network (Fig. 5.5 (b)).

Next, we study the effect of channel coding on the performance of the DisF algorithm. The average performance of the algorithm is shown in Fig. 5.6 when channel coding is used in the network and it is compared with the case that channel coding is not used. Each point in Fig. 5.6 represents the average value over 50 random topologies. In this set of simulations, we assume the probability at which a packet is transmitted successfully over a wireless link to be 0.8 if channel coding is not used. We observe that the minimum throughput increases by 28% when the number of flows is 10.

In Fig. 5.7, we study the effect of varying parameter V on both the minimum throughput and the delay in the network under the DisF algorithm. We used the total backlog in



Figure 5.6: Performance of the network with channel coding is compared with the case in which channel coding is not being used.

the network as a measure of the delay. Each point in Fig. 5.7 represents the average performance of the algorithm for 50 random topologies with 20 nodes and 5 data flows. We vary V from 0 to 50000 and it is shown that the minimum throughput in the network increases with increasing V at the expense of a linear increase of delay in the network.

Finally, we compare our approach with the one introduced in [8] as an example of a class of approaches that use utility functions to provide different notions of fairness including maxmin fairness. This is comparing to our proposed method which directly pushes the minimum throughput up instead of using utility functions. In [8], the problem of maximizing the aggregate network utility for all flows in the network was considered as maximizing $\sum_f u_f^{\beta}(\alpha_f)$ subject to $\boldsymbol{\alpha} \in \boldsymbol{\Lambda}$, where $u_f^{\beta}(x)$ is the utility function (i.e., $u_f^{\beta}(x) = x^{1-\beta}/(1-\beta), \beta \neq 1$, and $u_f^{\beta}(x) = \log x, \beta = 1$). It was shown that by increasing β in the above utility functions, the performance of the network converges to maxmin fairness provisioning. The two algorithms are compared in Figs. 5.8 and 5.9 regarding the minimum throughput they provide for network users and fairness provisioning. Each point represents the average of simulation results in 50 random topologies. To make the comparison fair,



Figure 5.7: The effect of increasing parameter V is shown on the minimum throughput (a). This is at the expense of increasing the congestion in the network (b).

we assume wireless links are completely reliable. We do not employ multipath routing and channel coding in the network. Parameter β for the utility function is chosen to be 0, 1, and 100 which is for aggregate throughput maximization, proportional fairness, and maxmin fairness, respectively. It is verified (Fig. 5.8) that the fairness index is improved under DisF algorithm compared with the utility-based algorithm when $\beta = 0$ and 1. The fairness index of the utility-based algorithm improves when β increases but at the cost of the dramatic decrease in the minimum achieved throughput (Fig. 5.9).

5.6 Summary

In this chapter, we studied fairness provisioning in multihop wireless networks. We developed an online decentralized algorithm to schedule new data packet admission and packet transmissions such that the minimum throughput of the network is maximized. We consid-



Figure 5.8: DisF algorithm is compared with the utility-based approach in terms of fairness provisioning using Jain's fairness index. It is shown that for large values of β , the utility-based approach has a better performance comparing with DisF algorithm. This is at the cost of a decrease in the minimum throughput. (Fig. 5.9)



Figure 5.9: DisF algorithm is compared with the utility-based approach when different number of flows are using the network. It is shown that increasing β is at the cost of decreasing the minimum throughput.

ered networks with multipath routing and channel coding. We studied the performance of the algorithm analytically. Through simulations, we showed that the proposed algorithm followed the optimal centralized approach with under control degree of sub-optimality. We also showed that the proposed algorithm improves the performance of the network regarding fairness comparing to the other approaches which ignore fairness provisioning. Moreover, we showed the effectiveness of channel coding on the performance of the network. Finally, we showed through simulations that our proposed approach has a better performance in terms of fairness provisioning comparing to the class of utility-based approaches. Since the proposed algorithm determines the scheduling at each time slot, it can adapt to the dynamic changes of the wireless environment.

Chapter 6

Throughput-Optimal Scheduling and Interference Alignment for MIMO Wireless Systems

6.1 Introduction

In this chapter, we study the use of interference alignment techniques for transmission scheduling in wireless MIMO systems with the goal of improving the network performance. Multiple antennas can be used for data transmission between the mesh nodes and the end users and also between the mesh nodes and the gateway. The current work in interference alignment (see Section 1.2.5) provides signal design for interference wireless networks with K users, when K is relatively small, by minimizing the interference leakage at each receiver. Therefore, in each transmission time all users are scheduled to transmit. However, while in a network with many users it may be impractical for the entire users to transmit simultaneously even employing interference alignment techniques, there is not enough attention to scheduling the data transmissions in such network settings. Packet admission control is also not considered in that body of work.

The main issue that needs to be considered in designing any network scheduling algorithm is stability and to this end we apply Lyapunov stability theory. The framework of Lyapunov stability theory has been used to establish stable, distributed, scheduling policies [103] for throughput maximization [105, 115] and energy minimization in single hop [106] and multihop [107] networks. However, to the best of our knowledge, there is no prior work on using interference alignment to improve the effective network capacity via Lyapunov stability theory.

For interference alignment to be practical, one would need to determine a *scheduling policy* and an appropriate *signal design* (i.e., encoding and decoding matrices) such that the interference caused by undesired signals at each receiver is minimized. Signal design depends on the channel conditions and the particular set of users scheduled for transmission at that time instant. In addition, new data packets must be admitted from the upper layer by the users subject to network stability considerations. The contributions of this chapter are as follows:

- We formulate a joint scheduling, signal design, and packet admission control problem with the goal of maximizing the aggregate network throughput and ensuring stability. We also propose a centralized scheduling and interference alignment (SIA) algorithm to solve the problem.
- Using Lyapunov stability theory, we transform the problem into a nonlinear mixedinteger programming (MIP) problem with non-convex constraints for each time slot. In the problem formulation, we incorporate interference alignment to construct necessary conditions for minimizing the interference.
- We transform the problem with non-convex constraints into a nonlinear MIP with convex constraints using the coordinate ascent method and semidefinite programming (SDP) techniques. For solving the MIP problem, we propose an algorithm that is based on the generalized Benders decomposition (GBD) method.

- We propose a semi-distributed scheduling and interference alignment (SDSIA) algorithm, which is heuristic and has lower computational complexity than SIA.
- Through simulation, we show that SIA converges to the solution after a few iterations. We also show that SDSIA, although suboptimal, has a similar performance as SIA. In addition, we determine the impact of the joint use of interference alignment and scheduling on the network throughput by comparing the performance of the proposed algorithm with two other approaches: (a) an interference alignment (IA) algorithm which employs interference alignment techniques without scheduling and (b) greedy maximal scheduling (GMS) which uses scheduling techniques without interference alignment.

This chapter is organized as follows: The system model is presented in Section 6.2. The joint scheduling, packet admission control, and signal design problem is formulated in Section 6.3. In Section 6.4, we solve the formulated problem using the GBD method. A semi-distributed heuristic is provided in Section 6.5. Simulation results are presented in Section 6.6, and the chapter is concluded in Section 6.7.

6.2 System Model

Consider a single-hop MIMO wireless network. Each link, together with its dedicated transmitter and receiver nodes, is called a *user*. Let $\mathcal{K} = \{1, \ldots, K\}$ denote the set of users. We assume that each user's receiver node can hear every other user's transmissions. Time is divided into equal-length slots. Let $\mathcal{T} = \{0, 1, \ldots, T-1\}$ denote the set of time slots. For each user $k \in \mathcal{K}$, we introduce a scheduling variable $\rho_k(t) \in \{0, 1\}$ such that $\rho_k(t)=1$ if user k transmits data in time slot t, and $\rho_k(t)=0$ otherwise. We assume that a user can send at most one data packet in each time slot.



Figure 6.1: Network topology for wireless MIMO system with K users.

Consider the network topology shown in Fig. 6.1. User k has M_k antennas at the transmitter node and N_k antennas at the receiver node. At each time slot t, if user k is scheduled to transmit, it prepares a data packet $\mathbf{x}_k(t)$ as a vector of symbols of size d_k . Then, the transmitter node of user k encodes the data packet with an encoding matrix $\mathbf{V}_k(t) \in \mathbb{C}^{M_k \times d_k}$, where \mathbb{C} denotes the set of complex numbers, and transmits the encoded $M_k \times 1$ vector over its M_k antennas.

For two users $k, l \in \mathcal{K}$, the wireless channel between the transmitter node of user k and the receiver node of user l is modeled by matrix $\mathbf{H}_{lk}(t)$ of size $N_l \times M_k$. At the receiver node of user l, the packet is received as an $N_l \times 1$ vector and is decoded using decoding matrix $\mathbf{U}_l(t) \in \mathbb{C}^{N_l \times d_l}$. The decoded data packet $\mathbf{y}_l(t)$ at time slot t can be represented as

$$\mathbf{y}_{l}(t) = \sum_{k \in \mathcal{K}} \rho_{k}(t) \mathbf{U}_{l}^{*}(t) \mathbf{H}_{lk}(t) \mathbf{V}_{k}(t) \mathbf{x}_{k}(t) + \mathbf{U}_{l}^{*}(t) \mathbf{n}_{l}(t), \qquad (6.1)$$

where matrix $\mathbf{U}_{l}^{*}(t)$ is the conjugate transpose of $\mathbf{U}_{l}(t)$ and $\mathbf{n}_{l}(t)$ is the additive white Gaussian noise (AWGN) at the receiver node of user l.

Interference alignment techniques aim at minimizing the projection of the interfering signal within the interference-free subspace of the receiver. For *ideal* interference alignment, we need to determine the encoding matrices $\mathbf{V}_1(t), \ldots, \mathbf{V}_K(t)$ and decoding matrices $\mathbf{U}_1(t), \ldots, \mathbf{U}_K(t)$ such that for each user l there is zero interference from other users $k \in \mathcal{K} \ (k \neq l)$ projected into the interference-free subspace of the receiver node of user l, and the desired signal is received through a full rank channel matrix, i.e.,

$$\mathbf{U}_{l}^{*}(t)\mathbf{H}_{lk}(t)\mathbf{V}_{k}(t) = \mathbf{0}, \qquad \forall k, l \in \mathcal{K}, \ k \neq l, \qquad (6.2)$$

$$\operatorname{rank}(\mathbf{U}_{k}^{*}(t)\mathbf{H}_{kk}(t)\mathbf{V}_{k}(t)) = d_{k}, \qquad \forall k \in \mathcal{K}.$$
(6.3)

Note that (6.2) and (6.3) are the interference alignment feasibility conditions [70]. However, *complete suppression* of interference at the receiver may not be practical.

Let $I_l(t) = \sum_{k \in \mathcal{K}, k \neq l} I_{lk}(t)$ denote the total interference leakage for any user $l \in \mathcal{K}$, where

$$I_{lk}(t) = \frac{1}{d_k} \operatorname{tr} \left(\mathbf{U}_l^*(t) \mathbf{H}_{lk}(t) \mathbf{V}_k(t) \mathbf{V}_k^*(t) \mathbf{H}_{lk}^*(t) \mathbf{U}_l(t) \right)$$
(6.4)

denotes the interference leaked by the transmitter of user k at the receiver of user l [70], and tr(·) denotes the trace of a matrix. Thus, at each time slot t, we aim to keep $I_l(t)$ for any scheduled user $l \in \mathcal{K}$ below a user specified threshold ϵ . That is,

$$\rho_l(t)I_l(t) \le \epsilon. \tag{6.5}$$

For scheduled user k, the received signal should be larger than the *receiver threshold* P_{th} . That is,

$$I_{kk}(t) \ge P_{th}\rho_k(t), \quad \forall \ k \in \mathcal{K}.$$
(6.6)

Equations (6.5) and (6.6) ensure that the interference from undesired signals is *sufficiently* suppressed at each receiver node l. On the other hand, if the corresponding transmitter

k is scheduled to transmit, the desired signal is sufficiently strong. To ensure error-free decoding at the receivers, P_{th} and ϵ have to be properly chosen. Note that $I_l(t)$ in (6.5) is the summation of the interference received at receiver l from all transmitters including those not scheduled at time slot t ($\rho_k(t) = 0$). Therefore, for (6.5) to be satisfied, we require elements of the encoding matrices for not scheduled users as well. Including non-scheduled users has no effect on the final solution.¹

Each user's transmitter node performs admission control and maintains a backlog queue. Let $Q_k(t)$ denote the number of packets that are waiting to be sent in the backlog queue of user k at time slot t. Let $\alpha_k(t)$ denote the number of packets that are admitted into the queue backlog of user k by the upper layer at time slot t. We assume that the number of admitted packets in each time slot is bounded by a constant α_{max} . That is, $\alpha_k(t) \leq \alpha_{max}$, for all $k \in \mathcal{K}$.

The backlog at the transmitter node of user k, $Q_k(t)$, can be modeled as a queue with arrival process $\alpha_k(t)$ and service process $\rho_k(t)$. That is,

$$Q_k(t+1) \le \max\{Q_k(t) - \rho_k(t), 0\} + \alpha_k(t), \quad \forall \ k \in \mathcal{K}.$$
(6.7)

We use the notion of strong stability [116]. The network is strongly stable if

$$\lim_{T \to \infty} \sup \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \mathbb{E}\{Q_k(t)\} < \infty.$$
(6.8)

The expectation is taken over all possible channel states. If the network is stable, then the admission rate, $\alpha_k(t)$ at each transmitter node of user $k \in \mathcal{K}$ is also the throughput at the

¹Note that we can avoid the involvement of the encoding matrices of not scheduled users by adding the term $\rho_k(t)$ in the expression of $I_l(t)$. However, this formulation does not lead to a tractable problem. On the other hand, it can be easily shown that the optimal solutions for both formulations are the same regarding the variables of scheduled users. The obtained solution for the variables of users that are not scheduled can be ignored.

corresponding receiver node. The average throughput for user $k \in \mathcal{K}$ in T time slots is

$$\bar{\alpha}_k = \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha_k(t)\}.$$
(6.9)

The aggregate network throughput is

$$\bar{\alpha} = \sum_{k \in \mathcal{K}} \bar{\alpha}_k = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \mathbb{E}\{\alpha_k(t)\} = \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha(t)\},$$
(6.10)

where $\alpha(t) = \sum_{k \in \mathcal{K}} \alpha_k(t)$ denotes the total number of packets that are admitted by the upper layer at time slot t. Finally, we define Π as the set of all admission rates $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_K)$ that satisfy the inequality in (6.8). That is, the network is stable when $\bar{\boldsymbol{\alpha}} \in \Pi$.

6.3 Problem Formulation

We now present the joint scheduling, admission control, and signal design problem formulation. The goal is to maximize the aggregate throughput of the network such that all queues remain stable. The optimization problem can be formulated as follows:

$$\begin{array}{ll}
\begin{array}{l} \underset{\boldsymbol{\alpha}(t), \, \boldsymbol{\rho}(t), \, \mathbf{U}(t), \, \mathbf{V}(t), \, t \in \mathcal{T}}{\text{maximize}} & \bar{\boldsymbol{\alpha}} \\
\text{subject to} & \bar{\boldsymbol{\alpha}} \in \Pi, & k \in \mathcal{K}, \\
& I_{kk}(t) \geq P_{th} \boldsymbol{\rho}_k(t), & k \in \mathcal{K}, \, t \in \mathcal{T} \\
& \rho_k(t) I_k(t) \leq \epsilon, & k \in \mathcal{K}, \, t \in \mathcal{T} \\
& \rho_k(t) \in \{0, 1\}, & k \in \mathcal{K}, \, t \in \mathcal{T} \\
& 0 \leq \alpha_k(t) \leq \alpha_{max}, \, k \in \mathcal{K}, \, t \in \mathcal{T}
\end{array} \tag{6.11}$$

where $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_K(t))$ denotes the vector of admitted packets at time slot t,

 $\boldsymbol{\rho}(t) = (\rho_1(t), \dots, \rho_K(t))$ denotes the scheduling vector, and $\mathbf{U}(t)$ and $\mathbf{V}(t)$ denote the set of all matrices $\mathbf{U}_k(t)$ and $\mathbf{V}_k(t)$ for $k \in \mathcal{K}$, respectively. In problem (6.11), the objective function is the aggregate network throughput over all time slots. The first constraint is the network stability constraint and ensures that the obtained solution leads to a stable network. The second and third constraints ensure the scheduling variables as well as the encoding and decoding matrices are selected such that the admitted data packets can be transmitted successfully. Note that P_{th} and ϵ are not independent. They must be chosen such that the signal to interference plus noise ratio is greater than a threshold Γ_{th} , that is $P_{th}/(\epsilon + \sigma_n^2) \geq \Gamma_{th}$. Instead of solving problem (6.11) to obtain the solutions for all time slots, we decompose this problem into multiple problems, one for each time slot. The solution to each problem gives suitable values for the variables in that particular time slot. We formulate the problems such that their solutions lead to the solution of problem (6.11). For this purpose, we first present some preliminaries.

We begin by summarizing some aspects of Lyapunov stability theory. Let the Lyapunov function $L(\mathbf{Q}(t))$ be a non-negative function of a vector $\mathbf{Q}(t) = (Q_1(t), \dots, Q_K(t))$. The Lyapunov drift is defined as $\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\}.$

Proposition 6.1 (Lyapunov Optimization [103]) Let $\alpha(t)$ be the utility function at time t, and A > 0, $\varepsilon > 0$, and Z > 0 be constants such that for all time slots t and queue vectors Q(t), we have

$$\Delta(\boldsymbol{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \boldsymbol{Q}(t)\} \le A - \varepsilon \sum_{k \in \mathcal{K}} Q_k(t) - Z\alpha^*,$$
(6.12)

where α^* can be any target value for utility function $\alpha(t)$. Then, we have

$$\alpha_{inf} \ge \alpha^* - A/Z,\tag{6.13}$$

$$\lim_{T \to \infty} \sup \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \mathbb{E}\{Q_k(t)\} \le \frac{A + Z(\alpha_{sup} - \alpha^*)}{\varepsilon},$$

where $\alpha_{inf} = \lim_{T \to \infty} \inf \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha(t)\}$ and $\alpha_{sup} = \lim_{T \to \infty} \sup \frac{1}{T} \sum_{t \in \mathcal{T}} \mathbb{E}\{\alpha(t)\}$

The proof of the proposition can be found in [103, pp. 82-84]. Proposition 6.1 implies that by satisfying inequality (6.12) at each time slot t, we can approach the target point α^* while the queue backlogs remain stable. Note that the larger the parameter Z, the closer we can get to α^* . This is at the expense of a linear increase in the aggregate queue backlog. Consider Lyapunov function $L(\mathbf{Q}(t)) = (1/2) \sum_{k \in \mathcal{K}} Q_k^2(t)$. Before calculating the Lyapunov drift, we state the following lemma.

Lemma 6.1 For any positive Q_1 , Q_2 , ρ , and α , if $Q_1 \leq \max[Q_2 - \rho, 0] + \alpha$, then

$$Q_1^2 \le Q_2^2 + \rho^2 + \alpha^2 - 2Q_2(\rho - \alpha).$$
(6.14)

The proof can be found in [103, 116]. According to Lemma 6.1 and inequality (6.7), we have

$$Q_k^2(t+1) \leq Q_k^2(t) + \rho_k^2(t) + \alpha_k^2(t) - 2Q_k(t)(\rho_k(t) - \alpha_k(t)), \qquad (6.15)$$

for all $k \in \mathcal{K}$. Thus, we can write

$$\Delta(\mathbf{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\} = \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\} - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\}$$

$$\leq A_{max} - \sum_{k \in \mathcal{K}} \mathbb{E}\{Q_k(t)(\rho_k(t) - \alpha_k(t)) \mid \mathbf{Q}(t)\} - Z\mathbb{E}\{\sum_{k \in \mathcal{K}} \alpha_k(t) \mid \mathbf{Q}(t)\}, \quad (6.16)$$

where $A_{max} = K(1 + \alpha_{max}^2)/2$. Note that inequality (6.16) holds for any algorithm.

Now, we present the SIA algorithm. In each time slot t, SIA selects the admission vector $\boldsymbol{\alpha}(t)$, scheduling vector $\boldsymbol{\rho}(t)$, encoding and decoding matrices $\mathbf{V}(t)$ and $\mathbf{U}(t)$, such
that the following problem is solved:

Maximizing the objective function of problem (6.17) is equivalent to minimizing the righ hand side of (6.16). The first set of constraints ensures that the desired signal is received at the receiver if the corresponding user is scheduled for transmission. The second set of constraints implies that the interference produced by the other users in the interference free subspace of any receiver which is scheduled for receiving data is suppressed. In Theorem 6.1, we explain why solving problem (6.17) leads to the optimal solution of problem (6.11).

Theorem 6.1 Let the SIA algorithm solve problem (6.17) in each time slot t. Then, SIA is throughput-optimal.² The throughput $\bar{\alpha}$ is at most A_{max}/Z away from the optimal value α^* .

Before proceeding to the proof, we note that a channel state-only (CSO) algorithm is an algorithm which makes (possibly random) decisions based on only the observed state of the channels. CSO algorithms require global knowledge of the channel state.

Proof Suppose there exists a channel state-only (CSO) algorithm which determines $\rho(t)$, $\mathbf{V}_k(t)$ and $\mathbf{U}_k(t)$ for $k \in \mathcal{K}$, and $\alpha(t)$ for all $t \in \mathcal{T}$, such that the network is stable and the

²Throughput optimality means that the algorithm can provide a larger aggregate throughput than any other algorithm while maintaining stability [117].

network throughput is equal to the optimal value α^* . Since the network is stable, we have stable queues at all transmitters and therefore an $\varepsilon > 0$ can be found such that

$$\mathbb{E}\{\rho_k(t) \mid \mathbf{Q}(t)\} > \mathbb{E}\{\alpha_k(t) \mid \mathbf{Q}(t)\} + \varepsilon, \quad \forall k \in \mathcal{K}.$$
(6.18)

From (6.16) and (6.18), for the CSO algorithm, we have

$$\Delta(\mathbf{Q}(t)) - Z\mathbb{E}\{\alpha(t) \mid \mathbf{Q}(t)\} \le A_{max} - \varepsilon \sum_{k \in \mathcal{K}} Q_k(t) - Z\alpha^*.$$
(6.19)

Recall that (6.16) is true for any algorithm including SIA. The SIA algorithm selects the variables which solve problem (6.17). Maximizing the objective function in problem (6.17) in all time slots is equivalent to minimizing the right hand side of (6.16). Thus, for SIA, the right hand side of inequality (6.16) is smaller than its value for any other algorithm including the CSO algorithm which in turn is smaller than the right hand side of (6.19). Therefore, we obtain (6.19) also for the SIA algorithm. This is the necessary condition (6.12) in Proposition 6.1 and leads to (6.13). Therefore, SIA can support any target value for the aggregate throughput α^* that can be achieved with any CSO algorithm.

Note that Theorem 6.1 assumes that α^* is an achievable throughput, which implies that there exists a CSO algorithm that achieves throughput α^* . In fact, the theorem states that the SIA algorithm is able to achieve any target throughput α^* that is *feasible*. Clearly, the maximum feasible throughput is the optimal value. Note that we can maximize the second term in the objective function in problem (6.17) independent of the first term by setting $\alpha_k(t) = \alpha_{max}$ whenever $Q_k(t) \leq Z$, and $\alpha_k(t) = 0$ otherwise ($\forall k \in \mathcal{K}$). Thus, we have the following problem:

$$\begin{array}{ll}
\underset{\boldsymbol{\rho}(t), \boldsymbol{U}(t), \boldsymbol{V}(t)}{\text{maximize}} & \sum_{k \in \mathcal{K}} Q_k(t) \rho_k(t) \\
\text{subject to} & I_{kk}(t) \ge P_{th} \rho_k(t), \quad k \in \mathcal{K}, \\
& \rho_k(t) I_k(t) \le \epsilon, \qquad k \in \mathcal{K}, \\
& \rho_k(t) \in \{0, 1\}, \qquad k \in \mathcal{K}.
\end{array}$$
(6.20)

In problem (6.20), the objective is to maximize the number of scheduled users where each user is weighted with its corresponding queue backlog.

6.4 Scheduling and Interference Alignment (SIA) Algorithm

Problem (6.20) is a nonlinear mixed integer programming (MIP) problem with nonlinear constraints. The multiplicative terms in $I_{lk}(t)$ make the problem hard to solve. We use several techniques to convert problem (6.20) into simpler problems that can be solved efficiently. Using those techniques may result in a sub-optimality in the obtained solution. First, to deal with the multiplicative terms $\mathbf{U}_l(t)$ and $\mathbf{V}_k(t)$ in $I_{lk}(t)$, we use the coordinate ascent method [118] and solve problem (6.20) *iteratively* by solving two separate problems at the transmitter and receiver sides. The new problems are still non-convex. Then, we use semidefinite programming (SDP) techniques to convert each problem into a linear MIP problem. Finally, we use generalized Benders decomposition (GBD) to solve the formulated MIPs. For the reminder of the discussion, we assume data packet $\mathbf{x}_k(t)$ to be a scalar, that is $d_k = 1$ for all users $k \in \mathcal{K}$. This means at each time slot, each scheduled transmitter sends a single symbol stream.

Using the coordinate ascent method [118], problem (6.20) can be separated into problems at the transmitter and receiver side, respectively. At each side, the signal design parameters related to the other side are considered as given input parameters. Since the problem is non-convex, this may lead to a sub-optimal solution. The problem at the transmitter side is

$$\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\mbox{maximize} & \sum_{k \in \mathcal{K}} Q_k(t) \rho_k(t) \\
\mbox{subject to} & \mathbf{V}_k^*(t) \mathbf{F}_{kk}(t) \mathbf{V}_k(t) \geq P_{th} \rho_k(t), & \forall \ k \in \mathcal{K}, \\
\end{array} \\
\begin{array}{ll}
\rho_l(t) & \sum_{k \in \mathcal{K}, \ k \neq l} \mathbf{V}_k^*(t) \mathbf{F}_{lk}(t) \mathbf{V}_k(t) \leq \epsilon, \quad \forall \ l \in \mathcal{K}, \\
\rho_k(t) \in \{0, 1\}, & \forall \ k \in \mathcal{K}, \\
\end{array}$$
(6.21)

where $\mathbf{F}_{lk}(t) = \mathbf{H}_{lk}^*(t)\mathbf{U}_l(t)\mathbf{U}_l^*(t)\mathbf{H}_{lk}(t)$. The objective function in problem (6.21) is linear in $\rho_k(t)$. It can be shown that $\mathbf{V}_k^*(t)\mathbf{F}_{kk}(t)\mathbf{V}_k(t)$ is convex in $\mathbf{V}_k(t)$ for each $k \in \mathcal{K}$.³ Therefore, the second set of constraints is convex while the first set is non-convex. To deal with the non-convexity in the first set of constraints, we use Lemma 6.2.

Lemma 6.2 For any vector $\mathbf{a} \in \mathbb{C}^N$ and matrix $\mathbf{B} \in \mathbb{C}^{N \times N}$, we have $\mathbf{a}^*\mathbf{B}\mathbf{a} = tr(\mathbf{B}\mathbf{A})$, where $\mathbf{A} = \mathbf{a}\mathbf{a}^*$.

Proof The lemma is proved by simply expanding both sides of the equality.

We rewrite $\mathbf{V}_{k}^{*}(t)\mathbf{F}_{lk}(t)\mathbf{V}_{k}(t)$ as $\operatorname{tr}(\mathbf{F}_{lk}(t)\mathbf{W}_{k}(t))$, where $\mathbf{W}_{k}(t) = \mathbf{V}_{k}(t)\mathbf{V}_{k}^{*}(t)$. Note that for this to be true, we need $\mathbf{W}_{k}(t)$ to be a rank one matrix, $\forall k \in \mathcal{K}$. Let $\mathbf{W}(t)$ denote the set of all $\mathbf{W}_{k}(t), \forall k \in \mathcal{K}$.

We also modify the second set of constraints in problem (6.21) to separate admission

³The convexity can be proved by verifying that the corresponding Hessian is positive semidefinite.

variables $\rho(t)$ from the other variables. Problem (6.21) can be transformed as

$$\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\mbox{maximize}\\ \rho(t), \mathbf{W}(t) \end{array} & \sum_{k \in \mathcal{K}} Q_k(t) \rho_k(t) \\
\mbox{subject to} & \operatorname{tr}(\mathbf{F}_{kk}(t) \mathbf{W}_k(t)) \geq P_{th} \rho_k(t), & \forall \ k \in \mathcal{K}, \\
\end{array} & \\
\begin{array}{ll}
\mbox{} \sum_{k \in \mathcal{K}, \ k \neq l} \operatorname{tr}(\mathbf{F}_{lk}(t) \mathbf{W}_k(t)) \leq \epsilon + B(1 - \rho_l(t)), & \forall \ l \in \mathcal{K}, \\
\end{array} & \\
\begin{array}{ll}
\mbox{} \rho_k(t) \in \{0, 1\}, & \forall \ k \in \mathcal{K}, \\
\end{array} & \\
\begin{array}{ll}
\mbox{} \operatorname{rank}(\mathbf{W}_k(t)) = 1, & \forall \ k \in \mathcal{K}. \end{array}
\end{array}$$
(6.22)

In problem (6.22), the objective function and the first two constraints are linear. However, the rank constraint makes the problem non-convex. The positive constant B is chosen large enough, and therefore, the second set of constraints has to be satisfied only if $\rho_l(t) = 1$. Nevertheless, (6.22) is a computationally-demanding nonlinear mixed integer optimization problem.

Similarly, the receiver-side problem can be formulated as

$$\begin{array}{ll}
 \text{maximize} & \sum_{k \in \mathcal{K}} Q_k(t) \rho_k(t) \\
 \text{subject to} & \operatorname{tr}(\mathbf{G}_{kk}(t) \mathbf{X}_k(t)) \geq P_{th} \rho_k(t), & \forall k \in \mathcal{K}, \\
 \sum_{k \in \mathcal{K}, \ k \neq l} \operatorname{tr}(\mathbf{G}_{lk}(t) \mathbf{X}_l(t)) \leq \epsilon + B(1 - \rho_l(t)), & \forall l \in \mathcal{K}, \\
 \rho_k(t) \in \{0, 1\}, & \forall k \in \mathcal{K}, \\
 \operatorname{rank}(\mathbf{X}_k(t)) = 1, & \forall k \in \mathcal{K}.
\end{array}$$
(6.23)

where $\mathbf{G}_{lk}(t) = \mathbf{H}_{lk}(t)\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t)$ and $\mathbf{X}_k(t) = \mathbf{U}_k(t)\mathbf{U}_k^*(t)$ for $\forall k, l \in \mathcal{K}$.

To solve the nonlinear mixed-integer optimization problem (6.22), we use the GBD method [84]. We decompose the problem into two problems: a *primal problem* and a *master problem*. The primal problem is a relaxed SDP problem which is a convex optimization problem with the encoding vectors $\mathbf{V}(t)$ as variables when the other variables are fixed

and it yields an upper bound for the final solution. The master problem is an MIP with binary variables $\rho(t)$ when the other variables are fixed and it yields a lower bound for the solution. We iteratively solve primal and master problems until their solutions converge. In this subsection, since we are discussing the solution of problem (6.22) in one particular time slot t, for ease of notation, we drop the time index t.

Primal problem $(m^{\text{th}} \text{ iteration})$

The input parameters (i.e., constants) include $\rho^{(m)}$ (obtained from the master problem in the m^{th} iteration). The primal problem is as follows:

$$\begin{array}{ll}
\text{minimize} & -\sum_{k \in \mathcal{K}} Q_k \rho_k^{(m)} \\
\text{subject to} & \operatorname{tr}(\mathbf{F}_{kk} \mathbf{W}_k) \ge P_{th} \rho_k^{(m)}, & \forall k \in \mathcal{K}, \\
& \sum_{k \in \mathcal{K}, \ k \neq l} \operatorname{tr}(\mathbf{F}_{lk} \mathbf{W}_k) \le \epsilon + B(1 - \rho_l^{(m)}), & \forall l \in \mathcal{K}, \\
& \mathbf{W}_k \succ \mathbf{0}, & \forall k \in \mathcal{K}.
\end{array}$$
(6.24)

In problem (6.24), the objective function is a constant. The two sets of constraints are linear in \mathbf{W}_k . Problem (6.24) is a standard form SDP and it can be solved by using a convex optimization solver such as CVX [119]. Note that in problem (6.24), the rank constraint rank(\mathbf{W}_k) = 1, $\forall k \in \mathcal{K}$, is relaxed. Having solved the primal problem, we use eigendecomposition to obtain a rank-one approximation of the obtained solutions \mathbf{W}_k . Thus, $\mathbf{V}_k^{(m)} = \sqrt{\gamma_k} \mathbf{q}_k$, where γ_k is the largest eigenvalue of matrix \mathbf{W}_k and \mathbf{q}_k is the corresponding eigenvector. Note that the rank-one approximation leads to a sub-optimal solution. From the solver, the corresponding Lagrange multipliers, $\boldsymbol{\lambda}^{(m)} = \{\lambda_1^{k(m)}, \lambda_2^{l(m)}, \forall k, l \in \mathcal{K}\}$, for the first and second set of constraints in problem (6.24) can also be obtained. The solution to primal problem $\mathbf{W}^{(m)}$ is used as an input to formulate the master problems for the next iterations. Given the input parameters $\rho^{(m)}$, if problem (6.24) is *infeasible*, then we formulate an l_1 -minimization problem (6.25) as in [84] and use its corresponding solution $\mathbf{V}^{(m)}$ to continue to the master problems in the next iterations.

$$\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\begin{array}{ll}
\end{array} \\ & \mathbf{W} \succeq 0, \, \beta^1, \, \beta^2 \end{array} & \sum_{k \in \mathcal{K}} \left(\beta_k^1 + \beta_k^2 \right) \\
\end{array} \\
\begin{array}{ll}
\begin{array}{ll}
\end{array} \\ & \text{subject to} & \operatorname{tr}(\mathbf{F}_{kk} \mathbf{W}_k) + \beta_k^1 \ge P_{th} \rho_k^{(m)}, & \forall k \in \mathcal{K}, \\
\end{array} \\ & \sum_{k \in \mathcal{K}, \ k \neq l} \operatorname{tr}(\mathbf{F}_{lk} \mathbf{W}_k) \le \beta_l^2 + \epsilon + B(1 - \rho_l^{(m)}), & \forall l \in \mathcal{K}, \\
\end{array} \\ & \beta_k^1, \beta_k^2 \ge 0, & \forall k \in \mathcal{K}. \end{array}$$

$$(6.25)$$

Problem (6.25) is an SDP problem and is always feasible. Similar to (6.24), the corresponding Lagrange multipliers $\lambda_1^{k(m)}$, $\lambda_2^{l(m)}$, $\forall k, l \in \mathcal{K}$, can be obtained. We define \mathcal{M} and \mathcal{M}' as the set of all iteration numbers at which the primal problem is feasible and infeasible, respectively. Note that similar to (6.24), in (6.25) the rank constraint rank(\mathbf{W}_k) = 1, $\forall k \in \mathcal{K}$, is relaxed. Therefore, we use a similar technique to find a rank-one approximation on the obtained solutions \mathbf{W}_k .

Master problem $(m^{\text{th}} \text{ iteration})$

The input parameters are $\mathbf{V}^{(n)}$ and $\boldsymbol{\lambda}^{(n)}$ (obtained from the primal problem), where vector $\boldsymbol{\lambda}^{(n)}$ is a concatenation of $\lambda_1^{k(n)}$, $\lambda_2^{k(n)}$, for $k \in \mathcal{K}$, $n \in \mathcal{M} \cup \mathcal{M}'$. The master problem is

$$\begin{array}{ll} \underset{\mu, \rho}{\operatorname{minimize}} & \mu \\ \text{subject to} & \mu \geq \Lambda \left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)} \right), & n \in \mathcal{M}, \\ & 0 \geq \Lambda' \left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)} \right), & n \in \mathcal{M}', \end{array}$$
(6.26)

where

$$\Lambda'\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right) = \sum_{k \in \mathcal{K}} \lambda_1^{k(n)} \left(P_{th} \rho_k - \mathbf{V}_k^{(n)*} \mathbf{F}_{kk} \mathbf{V}_k^{(n)}\right) + \sum_{l \in \mathcal{K}} \lambda_2^{l(n)} \left(\sum_{k \in \mathcal{K}, \ k \neq l} \mathbf{V}_k^{(n)*} \mathbf{F}_{lk} \mathbf{V}_k^{(n)} - (\epsilon + B(1 - \rho_l))\right), \ (6.27)$$

for all $n = 1, \ldots, m - 1$, and

$$\Lambda\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right) = -\sum_{k \in \mathcal{K}} Q_k \rho_k + \Lambda'\left(\boldsymbol{\rho}, \mathbf{V}^{(n)}, \boldsymbol{\lambda}^{(n)}\right), \qquad (6.28)$$

for all $n \in \mathcal{M}$. Problem (6.26) is an MIP that can be solved by an integer program solver such as MOSEK [89].

GBD algorithm

By weak duality [81, p. 225], in each iteration m, the solution of the master problem (6.26), $\mu^{(m)}$, is a lower bound for the optimum of problem (6.22). Moreover, in each iteration, the master problem has one additional constraint compared to the one formulated in the previous iteration and therefore, its optimum is equal to or greater than that of the previous iteration. Thus, the lower bounds on problem (6.22) achieved through solving the master problem in each iteration are non-decreasing. Since the integer variables are fixed in primal problem (6.24), its optimal value is always equal or worse (greater) than the optimal value of problem (6.22). Therefore, it provides an upper bound for the optimal value of problem (6.22). However, the order of the obtained upper bounds may be non-decreasing. We set the upper bound in each iteration equal to the minimum of all upper bounds achieved by that iteration. We solve master problem (6.26) in each iteration and then solve primal problem (6.24) given the optimal solution of the master problem. Since problem (6.22) is always feasible, monotonicity of the obtained upper bounds and lower bounds causes the GBD algorithm to converge to the solution. The GBD method solves problem (6.22)

Algorithm 6.1 Generalized Benders decomposition (GBD) method.

1: Initialization: $\rho^{(1)}$, $\mathcal{M} := \emptyset$, $\mathcal{M}' := \emptyset$, and m := 1. 2: Obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$ by solving primal problem (6.24). 3: $\mathcal{M} := \mathcal{M} \cup \{m\}.$ 4: flag := 1. 5: while $f lag \neq 0$ do Set m := m + 1. 6: Solve master problem (6.26) and obtain $\rho^{(m)}$, and the m^{th} lower bound $(LB^{(m)})$. 7: Solve primal problem (6.24) and obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$, and the m^{th} upper bound $(UB^{(m)})$. 8: if problem (6.24) is infeasible then 9: Solve problem (6.25) and obtain $\mathbf{V}^{(m)}, \boldsymbol{\lambda}^{(m)}$, and $UB^{(m)}$. 10: $\mathcal{M}' := \mathcal{M}' \cup \{m\}.$ 11: 12:else $\mathcal{M} := \mathcal{M} \cup \{m\}.$ 13:end if 14: if $|LB^{(m)} - UB^{(m)}| < \xi$ then 15:16: flag := 0.end if 17:18: end while

as shown in Algorithm 6.1. After initialization, in the first iteration, the primal problem (6.24) is solved given the initial $\rho^{(1)}$ (lines 1-2). The only condition for $\rho^{(1)}$ is that problem (6.24) must be feasible at the initial point. Since scheduling only one user to transmit is always possible, the corresponding binary point creates a feasible primal problem and can be used as an initial point. In the m^{th} iteration (m > 1), master problem (6.26) is formulated using $\mathbf{V}^{(n)}$, $\boldsymbol{\lambda}^{(n)}$ for $n \in \mathcal{M} \cup \mathcal{M}'$ (line 7) and the m^{th} lower bound $\mu^{(m)}$ is obtained. Then, the optimal solution of master problem (6.26), $\rho^{(m)}$, is used to formulate the primal problem (6.24) and $\mathbf{V}^{(m)}$ is obtained as well as the m^{th} upper bound (line 8). If problem (6.24) is not feasible, l_1 -minimization problem (6.25) is solved, $\mathbf{V}^{(m)}$ and the m^{th} upper bound are obtained and the iteration number is stored in \mathcal{M}' (line 13). In iteration mwhen the difference between the m^{th} lower bound and the m^{th} upper bound is less than a threshold ξ , the solution is obtained and is equal to $\mathbf{V}^{(m)}$, $\boldsymbol{\rho}^{(m)}$ (lines 15-17).

The SIA algorithm is presented in Algorithm 6.2. It is initialized with encoding and decoding matrices $\mathbf{V}^{(0)}(t)$, $\mathbf{U}^{(0)}(t)$ (line 1). For each user $k \in \mathcal{K}$, if the queue backlog $Q_k(t)$ is less than Z, α_{max} packets are admitted (lines 2-6). In iteration n > 0, Problem (6.22) is formulated using $\mathbf{U}^{(n-1)}(t)$ as input and the optimal solution $\mathbf{V}^{(n)}(t)$ and $\boldsymbol{\rho}^{(n)}(t)$ is obtained (line 11). Then, using $\mathbf{V}^{(n)}(t)$ as given, problem (6.23) is formulated and the optimal solution $\mathbf{U}^{(n)}(t)$ is obtained (line 12). If the difference between the current solution and the previous solution is less than η , then the obtained solution is equal to $\mathbf{V}^{(n)}(t)$, $\mathbf{U}^{(n)}(t)$, and $\boldsymbol{\rho}^{(n)}(t)$ (lines 13-15).

Algorithm 6.2 Efficient scheduling and interference alignment (SIA) algorithm. SIA is run at each time slot t, and takes the queue backlogs $\mathbf{Q}(t)$ and the channel state information as inputs. It is initialized with Z, η , α_{max} , P_{th} , and ϵ .

1: Initialization $\mathbf{U}^{(0)}(t)$, $\mathbf{V}^{(0)}(t)$, and $\boldsymbol{\alpha}(t) := \mathbf{0}$. 2: for each $k \in \mathcal{K}$ do if $Q_k(t) \leq Z$ then 3: $\alpha_k(t) := \alpha_{max}.$ 4: end if 5:6: end for 7: n := 0. 8: Set flag := 1. 9: while $flag \neq 0$ do Set n := n + 1. 10:Formulate problem (6.22) using $\mathbf{U}^{(n-1)}(t)$ and solve it with GBD (Algorithm 6.1) to obtain 11: $\mathbf{V}^{(n)}(t), \, \boldsymbol{\rho}^{(n)}(t).$ Formulate problem (6.23) using $\mathbf{V}^{(n)}(t)$ and solve it with GBD (similar to Algorithm 6.1) 12:to obtain $\mathbf{U}^{(n)}(t)$, $\boldsymbol{\rho}^{(n)}(t)$. 13:14: 15:end if 16: end while

6.5 Semi-Distributed Scheduling and Interference Alignment (SDSIA) Algorithm

The SIA algorithm presented in the previous section can determine an efficient solution, but it has a high computational complexity. In this section, we propose a semi-distributed scheduling and interference alignment algorithm. The proposed SDSIA algorithm has two parts. The first part is shown in Algorithm 6.3. It is executed at each time slot and has three phases.

1) Transmission scheduling (lines 2-6, 24-27): The candidate set S is the set of all users which have at least one packet to send. Then, the user k' with the largest number of packets in its queue backlog is considered as a scheduled user ($\rho_{k'}(t) = 1$). The chosen user is removed from the candidate set. The optimum signal design is obtained and its feasibility is checked regarding the first two sets of constraints in problem (6.20) through signal design and feasibility check phases. If the scheduled set is not feasible, then the most recently added user is removed from the scheduled set (i.e., $\rho_{k'}(t) = 0$). The above process is repeated until the candidate set is empty.

2) Signal design (lines 7-17): When there is only one user to be scheduled, its corresponding matrix $\mathbf{U}_l(t)$ is set equal to a preset value $\mathbf{U}^0(t)$. For the case when more than one user is scheduled, the signal design is obtained based on interference alignment techniques. The goal is to minimize the interference leakage $I_l(t)$ at all receivers whose corresponding transmitters are scheduled to transmit in time slot t. Therefore, we need to set the columns of matrix $\mathbf{U}_l(t)$ equal to the vectors spanning the subspace with the least interference [70]. At each receiver l with $\rho_l(t) = 1$, we determine

$$\mathbf{E}_{l}(t) = \sum_{k \in \mathcal{K}, k \neq l, \ \rho_{k}(t)=1} \mathbf{H}_{lk}(t) \mathbf{V}_{k}(t) \mathbf{V}_{k}^{*}(t) \mathbf{H}_{lk}^{*}(t).$$
(6.29)

137

We set $\mathbf{U}_l(t) = \langle \mathbf{E}_l(t) \rangle_{d_l}$, where $\langle \mathbf{E}_l(t) \rangle_{d_l}$ is a matrix consisting of the eigenvectors of matrix $\mathbf{E}_l(t)$ corresponding to its d_l smallest eigenvalues. Those vectors span the subspace with the least interference.

Algorithm 6.3 Semi-distributed scheduling and interference alignment (SDSIA) algorithm. SDSIA is run at each time slot t, and takes the queue backlogs $\mathbf{Q}(t)$, and channel state information as input. It is initialized with α_{max} , Z, P_{th} , ϵ , and preset decoding matrix $\mathbf{U}^{0}(t)$.

1: Initialization: Set $\rho(t) = 0$, $\mathbf{V}(t)$. 2: Initialize *candidate set* S of all users that have data to send. 3: while $S \neq \emptyset$ do Set $k' := \arg \max Q_k(t)$. 4: $\tilde{k} \in S$ Set $\rho_{k'}(t) := 1$. 5: $\mathcal{S} := \mathcal{S} \setminus \{k'\}.$ 6:for each $\{l \in \mathcal{K} \mid \rho_l(t) = 1\}$ do 7: 8: $\mathbf{E}_l(t) := [\mathbf{0}]_{N_l \times N_l}.$ for each $\{k \in \mathcal{K} \mid k \neq l, \rho_k(t) = 1\}$ do 9: $\mathbf{E}_{l}(t) := \mathbf{E}_{l}(t) + \mathbf{H}_{lk}(t)\mathbf{V}_{k}(t)\mathbf{V}_{k}^{*}(t)\mathbf{H}_{lk}^{*}(t).$ 10:end for 11: 12:if $\sum_{k \in \mathcal{K}} \rho_k(t) = 1$ then $\mathbf{U}_l(t) := \mathbf{U}^0(t).$ 13:else 14: $\mathbf{U}_l(t) := \langle \mathbf{E}_l(t) \rangle_{d_l}.$ 15:end if 16:end for 17:18:feasibility := 1. for each $\{l \in \mathcal{K} \mid \rho_l(t) = 1\}$ do 19:20: if $(I_{ll}(t) < P_{th}) \parallel (I_l(t) > \epsilon)$ then Set feasibility := 0. 21: 22:end if 23: end for if feasibility $\neq 1$ then 24: $\rho_{k'}(t) := 0.$ 25:26:end if 27: end while

3) Feasibility check (lines 18-23): In each iteration, having obtained the scheduled set of users and the signal design $\mathbf{V}(t)$ and $\mathbf{U}(t)$, the feasibility of the design is checked based on the interference alignment requirements. If the desired signal strength is higher than

 P_{th} at all receivers (i.e., the first set of constraints in problem (6.20) is satisfied), and the interference strength is also less than threshold ϵ (the second set of constraints in (6.20) is satisfied), then the design is feasible.

To be able to implement the signal design and feasibility check phases in a semidistributed manner, we have to run the algorithm in both transmitters and receivers in a receiver-based manner. That is, at the transmitters a similar algorithm as at the receivers is executed using channel reciprocity. At each time slot t, Algorithm 6.3 is first run at the receivers where the encoding matrices $\mathbf{V}(t)$ are set equal to an initial value and decoding matrices $\mathbf{U}(t)$ as well as $\boldsymbol{\rho}(t)$ are obtained. At the transmitters, using channel reciprocity and the obtained results for $\mathbf{U}(t)$, we set

$$\mathbf{\widetilde{H}}_{kl}(t) = \mathbf{H}_{lk}^{*}(t), \qquad \forall \ k, l \in \mathcal{K},$$
(6.30)

$$\overleftarrow{\mathbf{V}}_{k}(t) = \mathbf{U}_{k}(t), \qquad \forall \ k \in \mathcal{K}.$$
(6.31)

We use (.) to denote the corresponding variables when the algorithm is run at the transmitter. Then, the algorithm is run at the transmitters in a similar way as at the receivers and encoding matrices $\mathbf{V}(t)$ are obtained by finding $\overleftarrow{\mathbf{U}}(t)$. Signal design matrices $\mathbf{U}(t)$ and $\mathbf{V}(t)$ and the obtained schedule $\overleftarrow{\boldsymbol{\rho}}(t)$ are then used for data transmission.

In the transmission scheduling phase, we use the greedy maximal scheduling (GMS) policy [114] to maximize the first term in the objective function of (6.17) in a semi-distributed manner. GMS is sub-optimal and may be implemented in a distributed manner. It has been shown in [114] that in the 1-hop interference model, GMS can achieve at least half of the throughput achieved by the optimal policy. We refer to SDSIA as semi-distributed because the information related to the queue backlogs, encoding matrices, and design feasibility must be passed between users via control messages. Then, each phase will be performed at each user in a distributed manner.

The second part of SDSIA (i.e., admission control) is performed in a distributed manner in each transmitter. Each user $k \in \mathcal{K}$ checks if the number of packets waiting to be sent in its queue $Q_k(t)$ is less than design parameter Z. In this case, it admits α_{max} packets into the queue.

6.6 Performance Evaluation

We now present the simulation results of the SIA algorithm as well as the heuristic SDSIA algorithm. First, we show the convergence of the SIA algorithm in one time slot. Then, we show how the SDSIA algorithm follows the SIA algorithm when the algorithm parameters P_{th} and ϵ and the number of users K change. Finally, we compare our heuristic SDSIA algorithm with two other approaches: (i) an approach that is based only on interference alignment but without greedy scheduling and (ii) an approach that uses GMS but not interference alignment. We run the simulations for the network topology shown in Fig. 6.1. We assume that the channel coefficients in matrices $\mathbf{H}_{lk}(t)$, $\forall k, l \in \mathcal{K}$, follow a complex Gaussian distribution. In our simulations, we set the number of antennas at both transmitters and receivers to be equal to two. d_k is set to be one.

We verify the convergence of the SIA algorithm in one particular time slot and for one particular channel realization. Fig. 6.2 shows the optimum of primal problem (6.24), (6.25), and master problem (6.26) in a network with 5 and 10 users. The algorithm is run in a time slot at which all users have 50 packets in their backlog queues. As shown in Fig. 6.2, for the case of K = 5, after five iterations the lower bound and upper bound converge to the value of -150 which corresponds to the solution allowing users 1, 3, and 5 to send their packets. The number of iterations increases with the number of users in the



Figure 6.2: Convergence of SIA algorithm is shown in one time slot when there are 5 and 10 users in the network.

network. For K = 10, the algorithm converges to the value of -300 after 62 iterations and schedules users 1, 3, 5, 6, 8, and 10. Note that if the primal problem (6.24) is infeasible in one iteration, the upper bound gets the value of the last feasible primal problem in that iteration. We observe that the optimum for the master problem is non-decreasing.

Fig. 6.3 shows the optimum of problem (6.20) in one time slot for the SIA and SD-SIA algorithms when interference leakage threshold ϵ increases from 0 to 100. During the increase of ϵ , we also increase the receiver threshold P_{th} to keep the signal to interference ratio constant. We set the number of users to be five and, in that particular time slot, each user has 50 data packets to send. We run the SDSIA algorithm 2000 times for the same channel realization to make the results independent of the random behaviour of the SDSIA algorithm. We also run the simulations for both algorithms for 100 different channel realizations. We know that when ϵ increases, it is easier to satisfy the second set of constraints in problem (6.20) leading to an increase in the optimum value. However, the increase in the interference leakage threshold increases the interference in the receivers. Therefore,



Figure 6.3: The optimal value of problem (6.20) changes for both the SIA and SDSIA algorithms when interference leakage threshold ϵ and receiver threshold P_{th} change.

we need to increase the receiver threshold P_{th} to maintain the signal to interference ratio constant which makes it harder to satisfy the first constraint in problem (6.20). This causes a decrease in the optimum value. The results in Fig. 6.3 show that the optimal value of problem (6.20) first increases when ϵ increases to an optimal value after which the effect of increasing P_{th} dominates and the optimal value of problem (6.20) decreases. It is shown that the SIA algorithm achieves up to 0.77 transmissions per user (the objective function is 194) when $\epsilon = 5$. We can also see that the heuristic SDSIA algorithm finds solutions that are within 85% of optimality.

Fig. 6.4 shows the average number of transmissions per user in one particular time slot when the number of users changes. The complexity of the SIA algorithm increases with the number of users. Thus, we consider no more than 10 users for this part of our study. Again, consider a time slot in which each user has 50 data packets to send. We choose $P_{th} = 10$ and $\epsilon = 1$ for this set of simulations. We see that for both approaches, the number of transmissions per user decreases when the number of users increases. We also observe that the SDSIA algorithm is at least within 70% (90% on average) of optimality.



Figure 6.4: The number of transmissions per user is shown when the number of users K increases.

To highlight the effect of the two separate techniques employed in the SDSIA algorithm (interference alignment and greedy maximal scheduling), in Fig. 6.5 we compare the heuristic with two other algorithms in a wireless MIMO system. The IA algorithm uses interference alignment and all users transmit in each time slot (no GMS). The GMS algorithm uses greedy maximal scheduling (no interference alignment). We set $P_{th} = 10$, $\epsilon = 1$, and all backlog queues are initially empty. We vary the number of users from 1 to 15. Each simulation run is for 1000 time slots. Each data point in Fig. 6.5 shows the average data rate achievable for each user $R = \frac{1}{K} \sum_{l \in \mathcal{K}} R_l$ using each approach and is the average over several simulation runs. $R_l = \frac{1}{T} \sum_{t \in \mathcal{T}} R_l(t)$, where $R_l(t) = \log_2 \det(\mathbf{1}_{d_l \times d_l} + \mathbf{U}_l^*(t)\mathbf{H}_{ll}(t)\mathbf{U}_l(t)(\mathbf{1}_{d_l \times d_l} + \sum_{k \neq l} \mathbf{U}_l^*(t)\mathbf{H}_{lk}\mathbf{V}_k(t)\mathbf{V}_k^*(t)\mathbf{H}_{lk}^*(t)\mathbf{U}_l(t))^{-1})$, is the data rate of user l in time slot t. We can see that for all approaches, the average achievable data rate decreases when the number of users increases. Clearly, for one user we have the same performance for all algorithms. When the number of users is two, GMS's performance is decreased to half. This is because the data rate $R_l(t)$ is constant in GMS and



Figure 6.5: The average per user achievable data rate is shown when the number of users increases from 1 to 20.

therefore, it cannot schedule both users at the same time using the same rate. However, interference alignment techniques are able to allow both users to transmit with lower rate and therefore, SDSIA and IA's performances are the same and better than that of GMS. When the number of users increases, the IA algorithm which schedules all users to transmit in each time slot, has the highest number of transmitted packets per user. However, the interference created by the transmitting users decreases the average data rate. The SDSIA algorithm which employs both interference alignment and scheduling outperforms the other two approaches.

6.7 Summary

In this chapter, we considered efficient joint scheduling, packet admission control, and interference alignment in wireless MIMO systems with many users. We formulated the corresponding optimization problem as a nonlinear mixed integer programming problem with non-convex constraints and used a sequence of mathematical tools to solve the problem. We also developed a heuristic algorithm, which is computationally efficient. In addition, in our simulation studies, we observed that the heuristic is within 90% of optimality on average. We showed through simulations that our approach can dramatically improve the network performance when compared with systems that employ either only interference alignment or only scheduling. The presented work suggests, in essence, that network performance can be improved by considering interference alignment and scheduling decisions in a common framework.

Chapter 7

Conclusions and Future Work

In this chapter, we summarize the results and highlight the contributions of this thesis. We also suggest several topics for future work.

7.1 Research Contributions

- In Chapter 2, we considered data rate and channel code rate allocation in wireless networks with multipath routing and channel coding. We formulated a network utility maximization problem with the goal of improving network fairness with unreliable links. We considered both adaptive and non-adaptive coding schemes. We also developed our rate allocation algorithm such that it can adapt with fast changes in the network due to varying channel conditions and mobility. We studied the effects of fast fading on the performance of our algorithm.
- In Chapter 3, we considered the end-to-end delay for network data flows. For this purpose, we incorporated the end-to-end delay in the network utility function and solved the problem of data rate and channel code rate allocation with the goal of decreasing the delay in the network. We considered both maximizing the aggregate network utility and maximizing the minimum network utility through which the concept of fairness is considered. We showed through simulations that we can decrease the average delay in the network at the cost of a slight decrease in network throughput.

- In Chapter 4, we formulated the problem of data rate allocation for routing paths as well as network coding paths for a wireless network with unreliable links. We showed via simulations that considering the reliability of wireless links in our rate allocation problem leads to better performance compared to the case when reliability is not considered. This becomes more important when fairness is important to be provided.
- In Chapter 5, we considered the achievability of our data rate allocation schemes by providing a transmission scheduling and packet admission algorithm for wireless mesh networks. We verified that our algorithm follows the performance of a centralized data rate allocation algorithm. We also compared fairness provisioning under our proposed algorithm with that under similar algorithms that ignore fairness to observe the effects of fairness provisioning in network performance. Finally, we compared the performance of the algorithm with a class of fair algorithms that are based on maximizing the network utility function.
- In Chapter 6, we formulated the joint problem of interference alignment, admission control, and transmission scheduling in MIMO systems. The formulated problem is a non-linear mixed-integer programming problem and therefore, hard to solve. We used several techniques to break this problem into one convex optimization problem and one mixed-integer programming problem and solved it through Benders decomposition method. We also developed a heuristic approach with much less degrees of complexity to solve the problem in a semi-distributed manner. We showed through simulations that the heuristic follows the optimal central solution with some degree of sub-optimality. We also showed that using both scheduling techniques and interference alignment techniques, we gained better performance comparing with the case when one of them is being used.

7.2 Suggestions for Future Work

In the following, we consider several interesting possibilities for extension of the current work.

- 1. Delay-optimal transmission scheduling: In Chapter 3, we studied optimal rate allocation problems with the goal of maximizing the network utility when both throughput and end-to-end delay were incorporated in the utility function of each flow. The remaining question would be: "How can we achieve those optimal data rates in a distributed manner?" An interesting future work is to use stochastic optimization to design transmission scheduling, admission control algorithms, and channel code rate allocations to improve the performance of the network in terms of both the network throughput and the end-to-end delay.
- 2. Transmission scheduling with network coding: The performance of the network is improved via network coding. An interesting future work is to use Lyapunov techniques to develop transmission scheduling and admission control algorithms in wireless networks in which intersession network coding is enabled. In such network settings, the weight of each wireless link for data transmission would depend not only on the data backlog of that particular link, but also on the backlog of the other link which is going to participate in the network coding scheme. This extra dependency creates new challenges to be solved.
- 3. Interference alignment in multihop networks: Interference alignment techniques can also be used in multihop wireless networks with multiple antennas. Another future work is to use stability theory and Lyapunov techniques for transmission scheduling and encoding and decoding matrices design. In such network settings, the goal is to improve the number of simultaneous transmissions and thus the network

throughput.

4. Heuristic approaches for interference alignment: The main challenge of using interference alignment in MIMO systems is that interference alignment is computationally hard. It requires significant processing capability to compute schedules and encoding and decoding matrices at each time slot. One direction could be the investigation of effective heuristics.

Bibliography

- J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. W. Knightly, "Measurement and modeling of the origins of starvation in congestion controlled mesh networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [2] R. Jain, W. Hawe, and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report DEC-TR-301, Sept. 1984.
- [3] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. on Networking*, vol. 14, no. 2, pp. 302–315, Apr. 2006.
- [4] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *Proc. of IEEE Infocom*, Miami, FL, Mar. 2005.
- [5] J. Liu, A. Stolyar, M. Chiang, and H. V. Poor, "Queue back-pressure random access in multihop wireless networks: Optimality and stability," *IEEE Trans. on Information Theory*, vol. 55, no. 9, pp. 4087–4098, Sept. 2009.
- [6] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Trans. on Networking*, vol. 15, no. 6, pp. 1333–1344, Dec. 2007.
- [7] —, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.
- [8] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. on Networking*, vol. 16, no. 2, pp. 396– 409, Apr. 2008.
- [9] S. Sarkar and L. Tassiulas, "Fair distributed congestion control in multirate multicast networks," *IEEE/ACM Trans. on Networking*, vol. 13, no. 1, pp. 121–133, Feb. 2005.
- [10] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 163–173, Jan. 2005.

- [11] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. on Networking*, vol. 18, no. 3, pp. 960–972, June 2010.
- [12] —, "Approaching throughput-optimality in distributed CSMA scheduling algorithms with collisions," *IEEE/ACM Trans. on Networking*, vol. 1, no. 1, p. 1, Apr. 2010.
- [13] D. Wang, X. Wang, and X. Cai, "Optimal power control for multi-user relay networks over fading channels," *IEEE Trans. on Wireless Communications*, vol. 10, no. 1, pp. 199–207, Jan. 2011.
- [14] C. B. R. Estrello, G. H. Valdez, and F. A. C. Perez, "System-level analysis of mobile cellular networks considering link unreliability," *IEEE Trans. on Vehicular Technol*ogy, vol. 58, no. 2, pp. 926–940, Feb. 2009.
- [15] B. Wang, Z. He, and Y. Sun, "Distributed rate allocation and performance optimization for video communication over mesh networks," in *Proc. of ICIP*, San Antonio, TX, Sept. 2007.
- [16] A. Beljadid, A. S. Hafid, and A. Gendreau, "Design of wireless mesh networks: Expansion and reliability studies," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
- [17] A. P. Snow, U. Varshney, and A. D. Malloy, "Reliability and survivability of wireless and mobile networks," *IEEE Computer Mag.*, vol. 33, pp. 49–55, July 2000.
- [18] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [19] T. Issariyakul, E. Hossain, and A. S. Alfa, "End-to-end batch transmission in a multihop and multirate wireless network: Latency, reliability, and throughput analysis," *IEEE Trans. on Mobile Computing*, vol. 5, pp. 1143–1155, Sept. 2006.
- [20] A. Willig and H. Karl, "Data transport reliability in wireless sensor networks. A survey of issues and solutions," *PIK Journal*, vol. 28, pp. 86–92, June 2005.
- [21] U. Varshney, A. P. Snow, and A. D. Malloy, "Measuring the reliability and survivability of infrastructure-oriented wireless networks," in *Proc. of IEEE Local Computer Networks*, Tampa, FL, Nov. 2001.
- [22] J. W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithms for rate-reliability based tradeoff in network utility maximization," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 962–976, May 2006.

- [23] Q. Gao, J. Zhang, and S. V. Hanly, "Cross-layer rate control in wireless networks with lossy links: Leaky-pipe flow, effective network utility maximization and hopby-hop algorithms," *IEEE Trans. on Wireless Communications*, vol. 8, no. 6, pp. 3068–3076, June 2009.
- [24] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, Mar. 2008.
- [25] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *Proc. of IEEE Infocom*, San Francisco, CA, Apr. 2003.
- [26] Y. Fan, J. Zhang, and X. Shen, "Mobility-aware multi-path forwarding scheme for wireless mesh networks," in *Proc. of IEEE WCNC*, Las Vegas, NV, Apr. 2008.
- [27] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *Proc. of IEEE WCNC*, New Orleans, LA, Mar. 2003.
- [28] J. Huang, Z. Li, M. Chiang, and A. K. Katsaggelos, "Joint source adaptation and resource allocation for multi-user wireless video streaming," *IEEE Trans. on Circuits* and Systems for Video Technology, vol. 18, no. 5, pp. 582–595, May 2008.
- [29] D. O'Neill, A. Goldsmith, and S. Boyd, "Wireless network utility maximization," in Proc. of MILCOM, San Diego, CA, Nov. 2008.
- [30] Y. Lin and V. W. S. Wong, "Adaptive tuning of MIMO-enabled 802.11e WLANs with network utility maximization," in *Proc. of IEEE WCNC*, Las Vegas, NV, Apr. 2008.
- [31] D. O'Neill, A. Goldsmith, and S. Boyd, "Optimizing adaptive modulation in wireless networks via utility maximization," in *Proc. of IEEE ICC*, Beijing, China, May 2008.
- [32] D. O'Neill, E. Akuiyibo, S. Boyd, and A. Goldsmith, "Optimizing adaptive modulation in wireless networks via multi-period network utility maximization," in *Proc. of IEEE ICC*, Cape Town, South Africa, May 2010.
- [33] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, pp. 237–252, Mar. 1998.
- [34] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 104–116, Jan. 2005.

- [35] L. Chen, S. H. Low, and J. C. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *Proc. of IEEE Infocom*, Miami, FL, Mar. 2005.
- [36] D. O'Neill, B. S. Thian, A. Goldsmith, and S. Boyd, "Wireless NUM: Rate and reliability tradeoffs in random environment," in *Proc. of IEEE WCNC*, Budapest, Hungary, Apr. 2009.
- [37] M. Andrews, A. F. Anta, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *Proc. of IEEE Infocom*, San Diego, CA, Mar. 2010.
- [38] Y. Li, M. Chiang, and A. R. Calderbank, "Congestion control in networks with delay sensitive traffic," in *Proc. of IEEE Globecom*, Washington, DC, Nov. 2007.
- [39] E. Jorswieck, H. Boche, and S. Naik, "Energy-aware utility regions: Multiple access pareto boundary," *IEEE Trans. on Wireless Communications*, vol. 9, no. 7, pp. 2216–2226, July 2008.
- [40] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in ALOHA networks," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1858–1862, Oct. 2004.
- [41] J. W. Lee, M. Chiang, and A. R. Calderbank, "Utility-optimal medium access control: Reverse and forward engineering," in *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.
- [42] A. H. Mohsenian-Rad and V. W. S. Wong, "Cross-layer fair bandwidth sharing for multi-channel wireless mesh networks," *IEEE Trans. on Wireless Communications*, vol. 7, no. 9, pp. 3436–3445, Sept. 2008.
- [43] T. Voice and G. Raina, "Stability analysis of a max-min fair rate control protocol (RCP) in a small buffer regime," *IEEE Trans. on Automatic Control*, vol. 54, no. 8, pp. 1908–1913, Aug. 2009.
- [44] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204–1216, July 2000.
- [45] C. C. Wang and N. B. Shroff, "Beyond the butterfly- A graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proc. of IEEE ISIT*, Nice, France, June 2007.
- [46] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. on Information Theory*, vol. 52, pp. 2608–2623, June 2006.

- [47] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. of ACM SIGCOMM*, Pisa, Italy, Sept. 2006.
- [48] V. Shah-Mansouri and V. W. S. Wong, "Maximum-lifetime coding subgraph for multicast traffic in wireless sensor networks," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
- [49] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [50] A. Khreishah, C. C. Wang, and N. B. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [51] Y. Li, M. Chiang, A. R. Calderbank, and S. N. Diggavi, "Optimal rate-reliabilitydelay tradeoff in networks with composite links," *IEEE Trans. on Communications*, vol. 57, pp. 1390–1401, May 2009.
- [52] J. W. Lee, M. Chiang, and A. R. Calderbank, "Distributed algorithms for optimal rate-reliability tradeoff in networks," in *Proc. of IEEE ISIT*, Adelaide, Australia, Sept. 2005.
- [53] S. Huang and B. Mukherjee, "Adaptive reliable multi-path provisioning in WDM mesh networks," in *Proc. of IEEE ICC*, Beijing, China, May 2008.
- [54] H. Ma, D. Fayek, and P. H. Ho, "Availability-constrained multipath protection in backbone networks with double-link failure," in *Proc. of IEEE ICC*, Beiging, China, May 2008.
- [55] V. Bui, W. Zhu, A. Botta, and A. Pescape, "An MDP-based approach for multipath data transmission over wireless networks," in *Proc. of IEEE ICC*, Beijing, China, May 2008.
- [56] B. Yahya and J. B. Othman, "REER: Robust and energy efficient multipath routing protocol for wireless sensor networks," in *Proc. of IEEE Globecom*, Honolulu, HI, Dec. 2009.
- [57] D. Quintas and V. Friderikos, "Energy efficient relaying within a cell: Multi path versus shortest path routing," in *Proc. of IEEE Infocom*, San Diego, CA, Mar. 2010.
- [58] S. Misra, G. Xue, and D. Yang, "Polynomial time approximations for multi-path routing with bandwidth and delay constraints," in *Proc. of IEEE Infocom*, Rio De Janeiro, Brazil, Apr. 2009.

- [59] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," IEEE/ACM Trans. on Networking, vol. 6, pp. 885–896, Dec. 1999.
- [60] A. Nasipuri and R. Castaneda, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," ACM Journal on Mobile Networks and Applications., vol. 6, pp. 339–349, Aug. 2001.
- [61] S. J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in Proc. of IEEE WCNC, Chicago, IL, Sept. 2000.
- [62] L. Wang, L. Zhang, Y. Shu, and M. Dong, "Multipath source routing in wireless ad hoc networks," in Proc. of Canadian Conference on Electrical and Computer Engineering, Halifax, NS, May 2000.
- [63] R. Leung, J. Liu, E. Poon, A. L. C. Chan, and B. Li, "MP-DSR: A QoS-aware multipath dynamic source routing protocol for wireless ad-hoc networks," in *Proc. of LCN*, Tampa, FL, Nov. 2001.
- [64] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. of ICC*, Beijing, China, May 2001.
- [65] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. of ICNP*, Riverside, CA, Nov. 2001.
- [66] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energyefficient multipath routing in wireless sensor networks," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 4, pp. 10–24, Oct. 2001.
- [67] S. A. Jafar and M. J. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Trans. on Information Theory*, vol. 53, no. 7, pp. 2637 –2642, July 2007.
- [68] 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.
- [69] B. Nazer, M. Gastpar, S. A. Jafar, and S. Vishwanath, "Ergodic interference alignment," in *Proc. of IEEE ISIT*, Seoul, Korea, July 2009.
- [70] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
- [71] H. Maleki, S. A. Jafar, and S. Shamai, "Retrospective interference alignment," in *Proc. of IEEE ISIT*, Saint-Petersburg, Russia, Aug. 2011.

- [72] C. M. Yetis, G. Tiangao, 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.
- [73] —, "Feasibility conditions for interference alignment," in *Proc. of IEEE Globecom*, Honolulu, HI, Dec. 2009.
- [74] K. R. Kumar and X. Feng, "An iterative algorithm for joint signal and interference alignment," in *Proc. of IEEE ISIT*, Austin, TX, June 2010.
- [75] S. Gollakota, A. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proc. of ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.
- [76] I. Santamaria, O. Gonzalez, R. W. Heath, and S. W. Peters, "Maximum sum-rate interference alignment algorithms for MIMO channels," in *Proc. of IEEE Globecom*, Miami, FL, Dec. 2010.
- [77] H. S. Dhillon and R. M. Buehrer, "On the maximum sum-rate of cognitive MIMO interference channels," in *Proc. of MILCOM*, San Jose, CA, Nov. 2010.
- [78] C. Suh, M. Ho, and D. Tse, "Downlink interference alignment," *IEEE Trans. on Communications*, vol. 59, no. 9, pp. 2616–2626, Sept. 2011.
- [79] K. A. Das, S. Vishwanath, S. A. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," in *Proc. of IEEE ISIT*, Austin, TX, June 2010.
- [80] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. A. Jafar, and S. Vishwanath, "Network coding for three unicast sessions: Interference alignment approaches," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Oct. 2010.
- [81] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [82] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, A Tutorial on Geometric Programming. Springer Science and Business Media, 2007.
- [83] M. Chiang, *Geometric Programming for Communication Systems*. Foundations and Trends in Networking, 2005.
- [84] C. A. Floudas, Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications. Oxford University Press, 1995.
- [85] J. G. Proakis, *Digital Communications*. 4th edition, New York: McGraw-Hill, 2001.

- [86] R. Gupta and J. Walrand, "Approximating maximal cliques in ad-hoc networks," in *Proc. of IEEE PIMRC*, Barcelona, Spain, Sept. 2004.
- [87] C. Shannon, "A theorem on coloring the lines in the network," J. Math. Phys., vol. 28, pp. 148–151, Sept. 1949.
- [88] Y. Nesterov and A. Nemirovsky, Interior point polynomial algorithms in convex programming. SIAM, 1994.
- [89] "MOSEK software." [Online]. Available: http://www.mosek.com
- [90] D. Ren, Y. T. H. Li, and S. H. G. Chan, "Fast-mesh: A low-delay high-bandwidth mesh for peer-to-peer live streaming," *IEEE Trans. on Multimedia*, vol. 11, no. 8, pp. 1446 –1456, Dec. 2009.
- [91] K. Ronasi, A. H. Mohsenian-Rad, V. W. S. Wong, S. Gopalakrishnan, and R. Schober, "Optimal data and channel code rate allocation in wireless networks with multi-path routing," in *Proc. of IEEE ICC*, Kyoto, Japan, June 2011.
- [92] D. ONeill, A. J. Goldsmith, and S. Boyd, "Cross-layer design with adaptive modulation: Delay, rate, and energy tradeoffs," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
- [93] M. Saad, A. Leon-Garcia, and W. Yu, "Optimal network rate allocation under endto-end quality-of-service requirements," *IEEE Trans. on Network and Service Man*agement, vol. 4, pp. 40–49, Dec. 2007.
- [94] M. Saad, A. L. Garcia, and W. Yu, "Delay constrained optimal resource utilization of wireless networks for distributed control systems," *IEEE Communications Letter*, vol. 12, pp. 289–291, Apr. 2008.
- [95] M. G. Kallitsis, R. D. Callaway, M. Devetsikiotis, and G. Michailidis, "Distributed and dynamic resource allocation for delay sensitive network services," in *Proc. of IEEE Globecom*, New Orleans, LA, Dec. 2008.
- [96] S. Kalyanasundaram, E. K. P. Chong, and N. B. Shroff, "Optimal resource allocation in multi-class networks with user-specified utility functions," *Comput. Networks*, vol. 38, pp. 613–630, Apr. 2002.
- [97] D. Bertsekas and R. Gallager, *Data Networks, 2nd Edition*. Prentice Hall, 1992.
- [98] A. Mohsenian-Rad, J. Huang, V. Wong, S. Jaggi, and R. Schober, "Game-theoretical analysis of inter-session network coding," in *Proc. of IEEE ICC*, Dresden, Germany, June 2009.
- [99] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1999.

- [100] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," IEEE/ACM Trans. on Net., vol. 8, pp. 556–567, Oct. 2000.
- [101] D. P. Bertsekas, Nonlinear Programming, second edition. Athena Scientific, 1999.
- [102] D. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical methods. Athena Scientific, 1997.
- [103] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross Layer Control in Wireless Networks*,. Foundations and Trends in Networking, 2006.
- [104] M. J. Neely, "Delay-based network utility maximization," in Proc. of IEEE Infocom, San Diego, CA, Mar. 2010.
- [105] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
- [106] C. P. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Trans. on Mobile Computing*, vol. 9, no. 4, pp. 527–539, Apr. 2010.
- [107] L. Lin, X. Lin, and N. B. Shroff, "Low-complexity and distributed energy minimization in multihop wireless networks," *IEEE/ACM Trans. on Networking*, vol. 18, no. 2, pp. 501–514, Apr. 2010.
- [108] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proc. of IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.
- [109] L. Chen, S. H. Low, J. C. Doyle, and M. Chiang, "Cross layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.
- [110] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," in *Proc. of IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.
- [111] S. Chen, Y. Fang, and Y. Xia, "Lexicographic maxmin fairness for data collection in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 7, pp. 762–776, July 2007.
- [112] H. Shirani-Mehr, G. Caire, and M. J. Neely, "MIMO downlink scheduling with nonperfect channel state knowledge," *IEEE Trans. on Communications*, vol. 58, no. 7, pp. 2055–2066, July 2010.

- [113] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, "Back-pressure routing and rate control for ICNs," in *Proc. of ACM MobiCom*, Chicago, IL, Sept. 2010.
- [114] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [115] K. Ronasi, V. W. S. Wong, S. Gopalakrishnan, and R. Schober, "Distributed scheduling in multihop wireless networks with maxmin fairness provisioning," *IEEE Trans. on Wireless Communications*, vol. 11, pp. 1753–1763, May 2012.
- [116] M. J. Neely, Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan and Claypool, 2010.
- [117] K. Jagannathan, M. Markakis, E. Modiano, and J. N. Tsitsiklis, "Throughput optimal scheduling in the presence of heavy-tailed traffic," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Oct. 2010.
- [118] D. P. Bertsekas and J. N. Tsitsiklis, Parallel And Distributed Computation: Numerical Methods. Prentice Hall, 1989.
- [119] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.