SCHEDULING ALGORITHMS FOR NEXT GENERATION CELLULAR NETWORKS

by

MEHMET KARACA

Submitted to the Graduate School of Engineering
and Natural Sciences in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Sabancı University

January 2013

SCHEDULING ALGORITHMS FOR NEXT GENERATION CELLULAR NETWORKS

by Mehmet Karaca

APPROVED BY

Assoc. Prof. Dr. Özgür Erçetin ....
(Thesis Advisor)

Assoc. Prof. Dr. Özgür Gürbüz ....

Assoc. Prof. Dr. Albert Levi ....

Assist. Prof. Dr. Hakan Erdoğan ....

Assist. Prof. Dr. Hazer İnaltekin ....

DATE OF APPROVAL: January 18, 2013

To my dad and mum: Yusuf and Zeynep

SCHEDULING ALGORITHMS FOR NEXT GENERATION CELLULAR NETWORKS

Mehmet Karaca

PhD Thesis, 2013

Thesis Advisor: Assoc. Prof. Dr. Özgür Erçetin

Keywords: ***Wireless Scheduling, Resource Allocation, Cross-layer optimization, Stochastic Control***

Next generation wireless and mobile communication systems are rapidly evolving to satisfy the demands of users. Due to spectrum scarcity and time-varying nature of wireless networks, supporting user demand and achieving high performance necessitate the design of efficient scheduling and resource allocation algorithms. Opportunistic scheduling is a key mechanism for such a design, which exploits the time-varying nature of the wireless environment for improving the performance of wireless systems. In this thesis, our aim is to investigate various categories of practical scheduling problems and to design efficient policies with provably optimal or near-optimal performance.

An advantage of opportunistic scheduling is that it can effectively be incorporated with new communication technologies to further increase the network performance. We investigate two key technologies in this context. First, motivated by the current

under-utilization of wireless spectrum, we characterize optimal scheduling policies for wireless *cognitive radio networks* by assuming that users always have data to transmit. We consider cooperative schemes in which secondary users share the time slot with primary users in return for cooperation, and our aim is to improve the primary systems performance over the non-cooperative case. By employing Lyapunov Optimization technique, we develop optimal scheduling algorithms which maximize the total expected utility and satisfy the minimum data rate requirements of the primary users. Next, we study scheduling problem with *multi-packet transmission*. The motivation behind multi-packet transmission comes from the fact that the base station can send more than one packets simultaneously to more than one users. By considering unsaturated queueing systems we aim to stabilize user queues. To this end, we develop a dynamic control algorithm which is able to schedule more than one users in a time slot by employing hierarchical modulation which enables multi-packet transmission. Through Lyapunov Optimization technique, we show that our algorithm is throughput-optimal. We also study the resulting rate region of developed policy and show that it is larger than that of single user scheduling.

Despite the advantage of opportunistic scheduling, this mechanism requires that the base station is aware of network conditions such as channel state and queue length information of users. In the second part of this thesis, we turn our attention to the design of scheduling algorithms when complete network information is not available at the scheduler. In this regard, we study three sets of problems where the common objective is to stabilize user queues. Specifically, we first study a cellular downlink network by assuming that channels are identically distributed across time slots and acquiring channel state information of a user consumes a certain fraction of resource which is otherwise used for transmission of data. We develop a joint scheduling and channel probing algorithm which collects channel state information from only those users with sufficiently good channel quality. We also quantify the minimum number of users that must exist to achieve larger rate region than Max-Weight algorithm with complete channel state information.

Next, we consider a more practical channel models where channels can be time-

correlated (possibly non-stationary) and only a fixed number of channels can be probed. We develop learning based scheduling algorithm which tracks and predicts instantaneous transmission rates of users and makes a joint scheduling and probing decision based on the predicted rates rather than their exact values. We also characterize the achievable rate region of these policies as compared to Max-Weight policy with exact channel state information. Finally, we study a cellular uplink system and develop a fully distributed scheduling algorithm which can perform over general fading channels and does not require explicit control messages passing among the users. When continuous backoff time is allowed, we show that the proposed distributed algorithm can achieve the same performance as that of centralized Max-Weight algorithm in terms of both throughput and delay. When backoff time can take only discrete values, we show that our algorithm can perform well at the expense of low number of mini-slots for collision resolution.

YENİ NESİL HÜCRESEL TELSİZ AĞLAR İÇİN ÇİZELGELEME ALGORİTMALARI

Mehmet Karaca

Doktora Tezi, 2013

Tez Danışmanı: Doç. Dr. ÖZGÜR ERÇETİN

Anahtar Kelimeler: ***Telsiz hücresel ağlari, fırsatçı çizelgeleme problemleri, kaynak tahsisleme, telsiz haberleşme, stokastic kontrol***

Yeni nesil telsiz haberleşme sistemleri kullanıcıların artan isteklerini karşılamak için hızlı bir gelişme göstermektedir. Fakat, sınırlı frekans tayfı genişliği ve kablosuz kanal karakteristiğinin zamanda ve frekansta değişkenlik göstermesi sebebi ile kullanıcıların isteklerini karşılamak ve yüksek verim elde etmek için kolay uygulanabilir çizelgeleme ve kaynak tahsis algoritmalarının geliştirilmesi gerekmektedir. Bu tezde, farklı çizelgeleme problemleri incelenmiş ve bu problemler için verimli çizelgeleme algoritmaları geliştirilmiştir.

Fırsatçı çizelgeleme algoritmalarının en önemli özelliklerinden biri yeni gelişen teknolojiler ile kolay bir şekilde bütünleşebilmesidir. Bu bağlamda, iki önemli teknoloji incelenmiştir. İlk olarak, frekans tayfının halihazırda verimsiz kullanımını azaltmak için geliştirilmiş olan bilişsel radyo ağları için optimum çizelgeleme algoritmları tasarlanmış

ve incelenmiştir. Bu problemde, ikincil ve birincil kullanıcılar arasında işbirlikçi bir yöntem benimsenerek, birincil kullanıcıların performansının arttırılması amaçlanmıştır. Lyapunov en iyileme yöntemi ile kullanıcıların isterlerini karşılarken toplam sistem faydasını en iyileyen algoritmalar geliştirilmiştir. Frekans tayfının daha verimli kullanılmasına yarıyacak olan diğer bir teknoloji ise çoklu-bilgi iletimi yöntemidir. Bu yöntem ile baz istasyonu iki veya daha fazla kullanıcıya aynı anda hizmet verebilecektir. Bu tezde, çoklu bilgi iletim yöntemi ile yeni çizelgeleme algoritması geliştirilmiş ve bu algoritmanın optimum olduğu Lyapunov analiz yöntemi ile ıspatlanmıştır.

Fırsatçı çizelgeleme algoritmalarının birçok faydası olmasına rağmen, bu tür algoritmaların uygulanabilmesi için baz istasyonunun bütün ağ durumunu bilmesi gerekmektedir. Bu durumlar ise bütün kullanıcıların kanal durumunu ve kuruk uzunluğunun bilinmesini içermektedir. Tezin ikinci bölümü, bu durum bilgisi baz istasyonunda olmadığında geliştirilen çizelgeleme algoritmalarını içermektedir. Bu kısımda üç temel problem belirlenmiş olup bütün problemler için kullanıcıların kuyruk işleminin dengelenmesi amaçlanmıştır. İlk problem için, kullanıcıların baz istasyonundan veri aldıkları varsayılarak ve kullanıcıların kanal durumlarının zaman içinde bağımsız olarak değiştiği kabul edilerek en iyi çizelgeleme algoritması geliştirilmiştir. Ayrıca, bu algoritmanın Max-Weight algoritmasına göre ne kadar performans kazancı sağladığı matematiksel olarak gösterilmiştir.

İkinci problemde ise daha gerçekçi kanal durumları benimsenip ve baz istasyonun ise sadece belli sayıdaki kullanıcıdan kanal durum bilgisi alabileceği kabul edilip yeni çizelgeleme algoritması geliştirilmiştir. Bu algoritma kanal durum bilgisini tahmin ederek ve bu tahmin değeri ile hangi kullanıcılardan gerçek kanal bilgisini alacağına karar vermektedir. Daha sonra ise en iyi kullanıcıya kanalı tahsis etmektedir. Bu algoritmanın destekleyebileceği hız bölgesi tanımlanmış ve belirli durumlar için bu bölge matematiksel olarak belirlenmiştir. Son olarak kullanıcıların baz istasyonuna veri iletmek istedikleri durum incelenmiş, kanal ve kuyruk bilgisi olmadan ve bir merkezi sisteme gerek duymayan dağıtık çizelgeleme algoritması tasarlanması amaçlanmıştır. Sürekli zamanlı geri adım olduğu varsayılarak geliştirilen algoritmanın merkezi sistem olduğundaki verime erişebileceği gösterilmiştir. Daha sonra ayrık zamanlı geri adım kabul edilerek yeni bir dağıtık çizelgeleme algoritması geliştirilmiş olup bu algoritma için ne kadar masraf gerekeceği hesaplanmıştır.

Acknowledgments

I would not have been successful to complete this thesis without the encouragement, help, and guidance that I received over the years from many individuals. First and foremost is my advisor Dr. Özgür Erçetin. Today, when I look back it at all, I feel so fortunate to work with him. I would like to thank Dr. Özgür Erçetin for his generous support, friendship and encouraging guidance throughout my graduate studies. I specially appreciate his openness and teaching me to be an independent researcher.

In addition to my supervisor, I would like to thank Dr. Tansu Alpcan for providing me financial support, great facilities, and a good work environment to conduct my research in Berlin. I would like to thank also Dr. Eylem Ekici for my stay in the Ohio State University and valuable research discussions. I also would like to express my gratitude to my Ph.D. oral examination committee members Dr. Özgür Gürbüz, Dr. Hakan Erdoğan, Dr. Albert Levi and Dr. Hazer İnaltekin for taking their time serving on defense exam committee, and I thank them for kindly reading the thesis and their valuable comments.

I was so fortunate to be surrounded by many great friends during my studies in Sabancı University, who made my time at the university very enjoyable; Alico, Engin, Kayhan, Sarper, Yunus and Nurdagül I thank you for your invaluable friendship and support during the hard years of study. I wish you all the success, fortune and happiness in the world.

Above all, I would like to thank my family. Getting a Ph.D. would not have been possible without their unconditional love, patience and support. My dad and mom have always encouraged me to stay strong, and supported every decision I make. This dissertation is dedicated to them.

I would like to thank Sabanci University for supporting this research. This thesis is also supported in part by European Commission under Marie Curie IRSES grant PIRSES-GA-2010-269132 AGILENet.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

The goal of future wireless networks is to meet the excessive demand for multimedia traffic such as rapid file transfers, peer-to-peer sharing, online gaming and real-time audio/video streaming, all of which require ubiquitous high data rate connection. Next-generation wireless standards such as Long Term Evolution (LTE) [1] and Worldwide Interoperability for Microwave Access (WiMAX) [2] have been developed to be able to provide high-speed transmission over long-distance wireless. For instance, LTE is expected to achieve data rates as high as 1 Gbit/s for users with low mobility, and 100Mbit/s for users with high mobility. However, supporting such bandwidth hungry applications and maintaining an acceptable quality of service (QoS) to network users necessitate high performance requirements on today's wireless systems.

There are many performance metrics to be considered when evaluating a wireless system. The most important performance metric of the present and future wireless networks is the throughput, i.e., the rate of successful data delivery over a communication channel. Another important design and performance metric is the network delay, which specifies how long it takes for a message to travel across the network from source to destination. Network delays have direct impact real-time applications such as voice over IP and real-time audio/video streaming. Also, fairness among users and delay jitter can be considered as performance metrics for quantifying network performance.

Compared with wireline networks, wireless resource is very scarce because in wireless communication users must share a limited radio frequency spectrum. Dynamic resource management in which resources such as bandwidth and frequency are allocated by taking advantage of changing network conditions has been widely applied in modern wireless systems. The key function of effective resource management is scheduling which allocates system resources to a single or subsets of network users at each time, depending on user requirements and a given objective. Although many scheduling algorithms are available for wireline networks [3], they cannot be directly applied to wireless networks. This is due to the fact that i) network conditions vary randomly over time; ii) transmissions of users interfere with each other.

In wireless systems, channel states vary randomly over time due to unavoidable effects such as mobility and fading. These states determine the maximum rate at which data can be reliably transferred across the channel. For instance, users experiencing good channel conditions (e.g., those that are close to the base station) achieve higher data rate, and the contribution of those users to the system is higher in terms of throughput. Hence, good scheduling algorithms must take into account the variability of channel conditions and benefit from it to further improve the performance of wireless system. This type of scheduling is known as *opportunistic* scheduling and is widely applied in wireless system to improve the spectrum efficiency. In many practical cases, some form of fairness or QoS guarantees must be provided to users, and hence, considering only users' channel conditions may not be sufficient for an efficient scheduling. For instance, the number of data packets buffered in users' queues should be considered when some level of delay guarantees is required. Hence, the network state information which includes channel conditions and the amount unfinished work in user's buffers plays an important role when making a scheduling decision.

The availability of network state information at the scheduler depends on the type of communication. Specifically, in a typical cellular network, communication may occur from users to the base station (uplink communication) or from the base station to the users (downlink transmission) as shown in Figure 1.1. In a downlink transmission, the base station is aware of queue lengths of users but not channel state information (CSI),

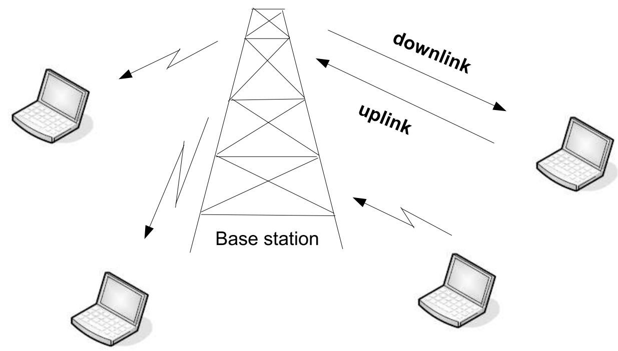

Figure 1.1: Cellular Network.

and this information must be sent back to the base station from users. In modern systems such as LTE and WiMAX, this information is sent back to the base station from users every 1-2 milliseconds. For uplink communication, the scheduler is neither aware of CSI nor queue backlog information. Hence, when designing scheduling policies these issues in uplink and downlink systems must be taken into account.

.

1.2 Focus of Thesis

In this thesis, we focus on designing *opportunistic* scheduling and resource allocation algorithms by considering both complete and incomplete network state information at the scheduler. We first explore how new communication techniques can be incorporated to further increase the network performance in terms of throughout and delay with complete network information at the scheduler. Then, we study scheduling problems with incomplete network information.

1.2.1 Utilization of emerging technologies

When complete network state information (i.e., both channel state and queue backlog information of all users) are available at the scheduler the key communication techniques can be incorporated with opportunistic scheduling to further improve the wireless spectrum efficiency in order to meet the future demand for high-data-rate wireless communication.

Cooperative Communication: Cooperative communication is one of the most important example of emerging techniques, which has recently been migrated as one of the state-of-the-art features of 3GPP LTE-Advanced (LTE-A). The basic idea behind cooperative transmission is that in a wireless environment, the signal transmitted by a source user, is also received by other users, which may aid in transmission as relay nodes. The relays may process and retransmit the signals that they receive. The destination then combines the signals coming from the source and the relays, thereby creating spatial diversity by taking advantage of the multiple receptions of the same data over various terminals and transmission paths. However, in contrast to single-user transmissions, cooperative communications involve multiple nodes transmitting simultaneously to a receiver. This feature of cooperative communication necessities new scheduling algorithms which tend to be further complicated with additional constraints imposed on source and relays nodes.

Multi-packet Transmission: The classical opportunistic scheduling limits its scope to the policies in which only a single user's data is transmitted at any time. Although transmitting to the user with the best channel achieves the maximum aggregate throughput, the channel access rate of a user can be low, which causes long packet delay experienced at the queue buffer. This motivates the design of opportunistic scheduling schemes which can schedule multiple users simultaneously (i.e., multi-user scheduling), consequently, can provide more frequent channel access and low delay. Fortunately, recent advancements in digital signal processing and radio hardware help in multi packet transmission and reception which makes multi-user scheduling possible in future wireless systems. In fact, it is possible to schedule two or more users at the same time by employing new communication techniques such as orthogonal coding and hierarchical modulation for code division multiple access (CDMA) and broadcast systems, respectively. Since two or more users are involved in the transmission within these schemes, more intelligent scheduling algorithms are required, and these algorithms must both take into the account user constraints, and perform interference management and power allocation for the optimal scheduling. As a result, even though complete network state information is available at the scheduler integrating opportunistic scheduling algorithms with new communication techniques bring their own set of challenges.

1.2.2 Incomplete Network State Information

Opportunistic scheduling can provide high system performance by exploiting the variation of user channels. However, in downlink transmission this is not cost-free since the scheduler requires the complete CSI in order to take advantage of the variations of the users' channels, which necessitates a certain amount of signalling between users and the base station. Ideally, the CSI should be sent back to the base station from every user via the uplink control channel at each scheduling instant. The amount of this information becomes too high as the number of users grow. Moreover, it may be impossible to collect CSI from all users due to the limited bandwidth of uplink control channel.

To give an idea of how much resource needs to be allocated for channel state feedback, CDMA/HDR (High Data Rate) system [4] can be considered, where the Signal-to-Noise Ratio (SNR) of each link is measured as channel state information. The value of the SNR is mapped to a value representing the maximum data rate that can support a given level of error performance. There are 11 different SNR levels in HDR system, hence, the channel state information of a single user is 4 bits long. This value is then sent back to the base station via the reverse link data rate request channel (DRC) every 1.67 ms. For instance if there are 25 users in a cell, 100 bits of channel state information is sent back to the base station every 1.67 ms. This requires 60Kbps of channel rate dedicated only for reporting channel states. Comparing this with the minimum data rate of HDR system of 38.4Kbps, and the average data rate of 308Kbps, one can immediately appreciate the need for method to reduce the amount of this feedback. The overhead due to the feedback of channel state information becomes even more significant in a multichannel communication system such as LTE. Also, for uplink transmissions the base station must be aware not only CSI but also queue lengths of users, which in turn results in even higher overhead. Therefore, it is important to design scheduling algorithms which not only achieve high throughput/delay performance but also minimize the cost of acquiring network state information.

1.3 Contributions and the Outline of the Thesis

This thesis studies the problem of maximization of wireless network performance under both complete and incomplete network state information by designing efficient scheduling and resource allocation policies. In this regard, the thesis develops scheduling algorithms which integrates and utilizes the key communication technologies to achieve higher performance, and also presents scheduling algorithms which can operate under incomplete network state information. The contributions are organized in a progressive way, following the different system assumptions that are considered:

- In Chapter 3, assuming infinite backlog at user buffers (i.e., a saturated system) and complete CSI at the base station, we develop optimal scheduling policies which exploit the time-varying channel conditions and realize the benefits of cooperative transmission in a cognitive radio network. We first propose a novel model in which secondary users have the opportunity to transmit their own data if they can improve the performance of a primary user via cooperation. Then, our scheduling policies find the optimal time sharing between primary and secondary users, based on the minimum data rate requirements of primary users. The basic structure of our optimal policies has the form of maximum weight where the weight depends on the online parameters in stochastic approximation technique and the virtual queues in Lyapunov optimization tools.

- In Chapter 4, taking into account randomness in packet arrivals (i.e., unsaturated system) and assuming that complete CSI is available at the base station, we address the problem of stability of user queues by utilizing multi-user scheduling at the base station. We develop a throughput-optimal algorithm which selects two users to be transmitted to at each time slot. This optimal algorithm jointly decides the best two users for scheduling and performs power allocation between those users. In addition, we analytically prove that the proposed algorithm achieves larger achievable rate region compared to the conventional Max-Weight algorithm in which only a single user is allowed to transmit.

- In Chapters 5, 6 and 7, we focus on scheduling problems with incomplete networks state information at the scheduler. Specifically, In Chapter 5 we design a joint scheduling and channel feedback algorithm for a downlink system when channel process is Independent identically distributed (iid) over time slots. We consider a channel probing model where acquisition of a single channel state consumes a certain fraction of data slot. With this probing model, it is possible to obtain the complete CSI at the expense of decreased throughput. Under this setup, we develop joint scheduling and channel probing algorithm whose main property is that it always schedules the user with the highest queue backlog and channel rate product. We characterize the achievable rate region of the algorithm by comparing it with the rate region of Max-Weight algorithm when CSI from all users are collected. For homogenous and heterogeneous channel conditions, we determine the minimum number of users that must be present in the network so that the rate region is expanded.

- In Chapter 6, we extend the channel model introduced in Chapter 5 by considering more practical channels which may be time-correlated (or continuous) and possibly non-stationary. We also consider a more realistic probing model where the base station acquires CSI from only a limited number of users due to the bandwidth constraint on the feedback channel. We develop a joint scheduling channel probing algorithm which tracks and predicts the instantaneous channel conditions by employing Gaussian Process Regression technique based on the actual CSI observed in previous slots. Then, this algorithm determines the set of channels to be probed by considering not only the queue sizes and predicted transmission rates but also by the level of the uncertainty in each channel prediction. We show that the algorithm can stabilize a scaled version (fraction) of the rate region. For two-user case, we quantify the fraction explicitly.

- In Chapter 7, we consider uplink communication channel and aim to stabilize the network when there is neither CSI nor queue size information of users. In this regard, we develop fully distributed scheduling algorithm which requires no message passing between users and performs with only local queue size and channel state information. By assuming continuous backoff time, we develop fully distributed algorithm which is able to schedule the best user at each time slot, and is provably throughput optimal. In other words, the algorithm can achieve the same performance as that of a centralized scheme in terms of both rate region and delay. Then, we consider a more practical IEEE 802.11 network where only discrete backoff time is available. We show that our algorithm is still throughput-optimal but a small number of mini-slots must be introduced for collision resolution.

1.4 Publication Lists

1.4.1 Journal Papers

- M. Karaca, T. Alpcan and O. Ercetin, "Entropy-based Active Learning for Scheduling in Wireless Networks", submitted.

- M. Karaca, Y. Sarikaya O. Ercetin, T. Alpcan, H. Boche, "Joint Opportunistic Scheduling and Selective Channel Feedback", submitted to IEEE Trans. Wireless Communication.

- M. Karaca and O. Ercetin, "Throughput Optimal Multi-user Scheduling via Hierarchical Modulation", accepted to IEEE Wireless Communications Letters.

- M. Karaca, K. Khalil, E. Ekici, and O. Ercetin, "Optimal Scheduling and Power Allocation in Cooperate-to-Join Cognitive Radio Networks", accepted to IEEE/ACM Transactions on Networking.

1.4.2 Conference Papers

- M. Karaca, T. Alpcan, O. Ercetin "Smart Scheduling and Feedback Allocation over Non-stationary Wireless Channels", Proceedings of ICC'12 WS - SCPA.

- M. Karaca, Y. Sarikaya, O. Ercetin, T. Alpcan, H. Boche "Efficient Wireless Scheduling with Limited Channel Feedback and Performance Guarantees, Proceedings of PIMRC'12 WS-WDN.

- K. Khalil, M. Karaca, O. Ercetin, and E. Ekici, "Optimal Scheduling in Cooperate-to-Join Cognitive Radio Networks", Proceedings of IEEE INFOCOM 2011, Shanghai, PRC, April 2011.

- M. Karaca and O. Ercetin, "Optimal Scheduling and Resource Allocation using Hierarchical Modulation in Wireless Networks", Proceedings of The Eighth IEEE and IFIP International Conference on Wireless and Optical Communications Networks, Paris, France, May 2011.

Chapter 2

Background and Related Literature

In this chapter, we briefly list out some necessary techniques and knowledge that will applied later to solve our problems. We begin with the definition of queue and network stability. We then explain the basic idea behind Lyapunov drift theory which will be used through out this thesis as a framework for the analysis. We also explain well known Max-Weight algorithm developed by Lyapunov drift theory. We end the chapter with a detailed review on opportunistic scheduling, cognitive radio, Max-Weight scheduling and other related literature.

2.1 Queues and Stability

Congestion may occur at many real life situations such as waiting lines at post offices, supermarkets, elevators, and in road traffic and computer-communication systems. Roughly, a queueing system describes contention on the resources, where resources are called servers. Queue process is directly related to congestion, and exhibits randomness and variation due to the random nature of arrival and service processes.

Queueing systems provide an important tool in modeling the performance analysis of telecommunication systems. First-in-first-out (FIFO) queues are among the most basic and widely used queueing form. The following equation to represent the dynamics of a discrete time queue

$$Q(t+1) = \max\{Q(t) - D(t), 0\} + A(t), \tag{2.1}$$

where $Q(t)$ represent the amount of unfinished work at the buffer and is called backlog at time t, $A(t)$ and $D(t)$ are real valued random variables which belong to a certain stochastic process. $A(t)$ and $D(t)$ represent the amount of new task arriving at queue and the amount of work processed by the server at time t, respectively. It is assumed that both $A(t)$ and $D(t)$ are non-negative and they are independent of each other.

In communication networks, user's data in bits or packets are stored until being transmitted. The service process usually does not change over time in wired networks whereas it varies according to the instantaneous channel condition in wireless networks. Specifically, in wireless systems, the channel between a user and the base station varies randomly. We represent the channel process of user n by $h_n(t)$. The maximum transmission rate at which user n transmits (or receives) without decoding error is given by [5],

$$R_n(h_n(t)) = \log_2\left(1 + \frac{Ph_n(t)}{\sigma^2}\right), \tag{2.2}$$

where P is the transmission power and σ^2 is the power of the additive white Gaussian noise. In practice, there are finite number of modulation and coding schemes and only a fixed set of data rates can be supported. For instance, in CDMA/HDR system [4] there 11 SNR levels, which means there are 11 different transmission rates which vary between 34.8 kb/s to 2400 kb/s. Also, let $A(t)$ represent the amount of new arrivals that enter the queue during slot t. Let $Q(t)$ represent the current backlog in the queue which may be interpreted as the number of packets or/and bits at time slot t. Then,

$$Q(t+1) = \max\{Q(t) - R(t), 0\} + A(t), \tag{2.3}$$

Regarding a queueing system, the most important performance metrics are stability, throughput and delay experienced by a packet at queue buffer. In this thesis, one of our primary concern is the stability of queue that refers to the behavior of the queue-length process. In a stable network the total number of packets or bits in the network will not become infinite, whereas in an unstable network, queue lengths grow unboundedly. The stability property of a queueing network is a good indicator to the average delay experienced and the throughput achieved by the users. There are a variety of stability definitions of a queue. The most common constraint of queue stability is as follows;

$$\mathbb{E}[A(t)] \leq \mathbb{E}[D(t)]$$

The intuition behind this constraint is that as long as arrival rate does not exceed departure rate, the server is able to perform all the task in finite time, and the queue is stabilized. However, this definition is not sufficient to describe every situation. For example, in the case where there are multiple queues that are served by a server as we consider in our chapters the departure rate or average throughput is actually the optimization objective. Therefore $\mathbb{E}[D(t)]$ cannot be determined before solving the problem at the first place. As a result, we need more general definition for the queue stability. First, we introduce the rate stability:

Definition 2.1. *A queue is rate stable if*

$$\limsup_{t \to \infty} \frac{Q(t)}{t} = 0 \quad \textit{with probability 1} \tag{2.4}$$

However, rate stability requires that arrival rate or departure rate have well-defined limits. For the case where arrival rate or departure rate does not have well-defined limits, we present a more general stability definition.

Definition 2.2. *A queue is strongly stable if*

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(Q(t)) < \infty \tag{2.5}$$

According to the definition of strong stability, the queue backlog will always be finite on average.

Definition 2.3. *A network is strongly stable if all individual queues of the network are strongly stable.*

2.2 Lyapunov Optimization

For general case where there are N users ($N \geq 1$) in the network, a controller (or scheduler) allocates the channel to a single user (or a subset of users) at a given time slot. Let $I_n(t)$ be the scheduler decision, where $I_n(t) = 1$ if user n is scheduled (e.g., full power is assigned to user n) for transmission in slot t, and $I_n(t) = 0$ otherwise. Let $\mathbf{I}(t) = (I_1(t), I_2(t), \cdots, I_N(t))$ be the corresponding vector, and $\mathcal{I}$ be the set of all possible scheduling vectors. Since transmission rate of a user is completely characterized given channel states and the schedule vector, we have

$$R_n(t) = R_n(h_n(t), \mathbf{I}(t))$$

and in vector form,

$$\mathbf{R}(t) = \mathbf{R}(\mathbf{h}(t), \mathbf{I}(t))$$

Let λ be time average expected arrival rate, i.e., $\lambda = \mathbb{E}[A(t)]$. Also let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_N)$ be the arrival rate vector.

Definition 2.4. *The achievable rate region of a network denoted by Λ is the closure of the set of all arrival rate vectors $\boldsymbol{\lambda}$ that can be stably supported by the network, considering all possible strategies for scheduling and resource allocation.*

In [6] and [7] it has been shown that the rate region is given by

$$\Lambda = \sum_{\mathbf{h} \in \mathcal{H}} \pi(\mathbf{h}) \text{Convex-hull}\{\mathbf{R}(\mathbf{h}(t), \mathbf{I}(t)) | \mathbf{I}(t) \in \mathcal{I}\} \tag{2.6}$$

where $\mathcal{H}$ is the set of all possible channel states. Next, we introduce the theory of Lyapunov drift.

Definition 2.5. *An algorithm is said to be throughput-optimal if it ensures network stability for any rate vector within rate region Λ.*

In [6], the authors have developed a throughput-optimal algorithm called Max-Weight algorithm by using Lyapunov drift theorem which we explain next. Basically, *Max-Weight* algorithm schedules the user with the highest queue backlog and transmission rate product at every time slot:

$$n^* = \operatorname*{argmax}_n Q_n(t)R_n(t) \tag{2.7}$$

The Max-Weight algorithm uses queue lengths as user weights so that if a user does not receive enough service, its queue builds up, which forces the algorithm to allocate more resources to that user. This interaction between queue lengths and scheduling guarantees the throughput optimality of resource allocation.

Lyapunov drift is an important mathematical tool that enables us to develop control algorithms for the strong stability of a network. The idea is based on a Lyapunov function which is a non-negative function of all queues in the network [8]. Network control decisions are then given such that Lyapunov function from one slot to the next is minimized. Let $\mathbf{Q}(t) = (Q_1(t), Q_2(t), \cdots, Q_N(t))$ in a network with N users. Suppose that the goal is to stabilize the backlog process $\mathbf{Q}(t)$. Define the following quadratic Lyapunov function and the one-slot conditional Lyapunov drift;

$$L(\mathbf{Q}(t)) \triangleq \sum_{n=1}^{N} Q_n^2(t), \tag{2.8}$$

$$\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\right]. \tag{2.9}$$

where the expectation is taken over all possible states of $\mathbf{Q}(t)$ in one time slot. The Lyapunov drift has the following important theorem relating to queue stability.

Lemma 2.1 ([8], Lemma 4.1)**.** *If there exits constants $B > 0$ and $\epsilon > 0$, such that for* *all time slots t we have:*

$$\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\right] \leq B - \epsilon \sum_{n=1}^{N} Q_N(t) \tag{2.10}$$

then the network is strongly stable and

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \mathbb{E}(Q_n(t)) \leq \frac{B}{\epsilon} \tag{2.11}$$

Lyapunov drift represents the expected change in the Lyapunov function from one slot to the next. If the condition (2.10) is satisfied then these exits a positive $\delta > 0$ such that $\Delta(\mathbf{Q}(t)) < -\delta$ whenever $\sum_{n=1}^{N} Q_n(t) \geq (B+\delta)/\epsilon$. In other words, when queue sizes are sufficiently large then the expected Lyapunov drift becomes negative, which implies that queue sizes do not increase and network stability is achieved. Lyapunov drift theory can also be used to to deal with performance optimization and queue stability problems simultaneously in a unified framework.

Suppose that the goal is to stabilize the backlog process $\mathbf{Q}(t)$ while maximizing the time average of a scalar-valued utility function $g(\cdot)$ of transmission rate process $\mathbf{R}(t)$. Suppose that the optimal value of $g(\cdot)$ is g^*. Define the following quadratic Lyapunov function and conditional Lyapunov drift

$$L(\mathbf{Q}(t)) \triangleq \sum_{n=1}^{N} Q_n^2(t), \tag{2.12}$$

$$\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\right]. \tag{2.13}$$

We restate a result of [8].

Theorem 2.1. *(Lyapunov Optimization citeGeorgiadis:Resource06, Theorem 5.4) For the scalar valued function* $g(\cdot)$, *if there exists positive constants* K, ϵ, B, *such that for all time slots* t *and all unfinished work vectors* $\mathbf{Q}(t)$ *the Lyapunov drift satisfies the* *condition*

$$\Delta(\mathbf{Q}(t)) - K\mathbb{E}[g(\mathbf{R}(t))|\mathbf{Q}(t)] \leq B - \epsilon \sum_{n=0}^{N} Q_n(t) - Kg^*, \tag{2.14}$$

then the time average utility and queue backlog satisfy:

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} g(\mathbb{E}]\mathbf{R}(\tau)]) \geq g^* - \frac{B}{K}, \tag{2.15}$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{l=1}^{L} \mathbb{E}[Q_n(\tau)] \leq \frac{B + K(\bar{g} - g^*)}{\epsilon}, \tag{2.16}$$

where $\bar{g} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[g(\mathbf{R}(\tau))]$.

This is the one of the most important theorem in Stochastic Lyapunov Optimization theory, which establishes the tradeoff between the utility function $g(\cdot)$ and queue backlog $\mathbf{Q}(t)$. Note that the aim of conventional optimization methods is to maximize the system utility. Unlike conventional methods, Lyapunov Optimization first transforms system constraints into queue stability constraints which are usually referred as virtual queues. Then, it minimizes the drift $\Delta(\mathbf{Q}(t))$ plus the penalty $K\mathbb{E}[g(\mathbf{R}(t))|\mathbf{Q}(t)]$ in the lefthand side of (2.14). By doing that, the system constraints is satisfied as in (2.16) and the penalty is minimized (or utility is maximized) as in (2.15). Here, the parameter $K > 0$ is introduced, and K controls the tradeoff between the system constraints and penalty. Specifically, the system utility approaches its maximum (or penalty decreases) as K increases. On the other hand, the queuing backlog increases with K. Hence, at this point it brings the tradeoff between penalty (utility) and system delay: If more emphasis is placed for minimizing the penalty (or maximizing utility), we should choose a larger K; if more emphasis is placed for minimizing the average delay, we should choose a smaller K.

2.3 Related Literature

Scheduling is one of the most active research areas in wireless networking and a large body of work in the literature has been devoted to the development of opportunistic wireless scheduling under different performance criteria and constraints. Depending on the considered system models, these works can be categorized into two primary groups: unsaturated and saturated systems.

There has been a plethora of work designing scheduling polices for saturated systems where users are infinitely backlogged and always have data to transmit. In the literature the assumption of infinite backlogged queues has been widely applied to various communication systems. This is because with this assumption it is easy to obtain closed form solutions with important insights into the problem of utility maximization. In particular, throughput of a system with infinite backlogged queues provides an upper bound on the maximum achievable performance of any arbitrary system. The works [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] considering saturated system with complete CSI are often concerned with channel assignment so that the user with the best channel quality accesses the channel at any given time. Notable among these are Proportional- Fair (PF) scheduling [11], [12], [13] that optimizes aggregate logarithmic utility. Particularly, under resource sharing constraints, the long term fairness is achieved by assigning offsets to user utility functions [9].

In fact, when complete CSI is available at the scheduler, the developed algorithms for both saturated and unsaturated systems can be adapted to the new communication techniques to further improve the performance of wireless networks. Cognitive radio, cooperative communication (Chapter 3) and multi-packet transmission (Chapter 4) are three important examples of such emerging techniques, which we focus on in this thesis.

In Chapter 3 of this thesis, we study the scheduling problem for a saturated system with complete CSI in cognitive radio networks, which was first promoted by Mitola [9]. The motivation for cognitive radios comes from the observation that regulatory bodies [19], [20] in the world (i.e., the Federal Communications Commission) found that most radio frequency spectrum was inefficiently utilized. Cellular network bands are overloaded in most parts of the world, but other frequency bands (such as military, amateur radio and paging frequencies) are insufficiently utilized. Such a spectrum usage pattern is mainly due to the fact that government agencies assign fixed pieces of the spectrum to license holders (primary users) often on a long-term basis and for large geographical regions. Fixed spectrum allocation prevents rarely used frequencies (those assigned to specific services) from being used, even when any unlicensed users would not cause noticeable interference to the assigned service. Therefore, regulatory bodies in the world have been considering to allow unlicensed users (secondary users) in licensed bands if they would not cause any interference to licensed users. These initiatives have focused cognitive-radio research on dynamic spectrum access.

Approaches to cognitive radio can be divided into two categories: commons model and property-rights model [21], [22]. In the commons model, the primary network is oblivious to the secondary network activity and the aim of secondary users is to detect the spectrum holes without interacting with the primary system and exploit the detected transmission opportunities. These spectrum holes represent the absence of primary activity either in time, frequency, or space. A key challenge here is to maximize secondary user's opportunities while limiting the interference caused to the primary users due to imperfect knowledge of the primary user channel occupancy state. Hence, such a scheme naturally requires spectrum sensing for opportunity detection and spectrum access and sharing [23].

In the property-rights model (spectrum leasing), primary users (PUs) are aware of the existence of secondary users (SUs) on a given bandwidth, and willing to lease the spectrum for a fraction of time. In literature, spectrum leasing model is usually coupled with cooperative communication which is an emerging and powerful solution that can improve the performance of wireless systems [24], [25]. The basic motivation behind cooperative transmission is based on the broadcast nature of wireless transmission where the signal transmitted by a source node, it can be received by other nodes in the network, which are usually referred as relay nodes. These relay nodes retransmit the received message, and the destination node can combine the signal coming from both relays and source node and creates spatial diversity by taking advantage of the multiple receptions of the same data. It was shown in [26], [27] that cooperative communication promises significant capacity and multiplexing gain increase in wireless networks.

Cognitive radio system applying spectrum leasing approach was investigated in [28], [29], [30] by considering only one primary transmitter. In [28], the authors presented a spectrum leasing scheme that allows PU to lease its own bandwidth for a fraction of time in exchange for enhanced performance guarantee via cooperation with SUs. As a result, more spectrum access opportunity is left for SU to transmit their own data. The authors in [29], [30] developed a game theoretic framework for a spectrum leasing in which PU actively participates in a non-cooperative game with SUs. In these works, PU plays an active role and allows SUs' access while meeting its own minimum Quality-of-Service (QoS) requirement. On the other hand, SUs aim to achieve energy efficient transmissions as long as they do not cause excessive interference to PU. Extending to the work [28], [29], [30] to a more general case, where there are multiple PUs and SUs, requires more complex scheduling algorithms.

Another emerging technology that can improve the spectrum efficiency is multi-user scheduling. In the classical opportunistic scheduling, only a single user which usually refers to the user (or subset of users) with the best channel quality is allowed to access to the channel at the same time slot and frequency band. However, in a multi-user scheduling scheme, multiple users can be assigned the same time-frequency resource. One method to perform multi-user scheduling is the orthogonal code allocation-based scheduling which was proposed to schedule two users simultaneously over CDMA based network [31]. For 4G cellular systems such as LTE, another multi-user scheduling technique hierarchical modulation (also known superposition coding or embedded modulation) has been applied [32], [33], [34].

Hierarchical modulation (HM) is a a physical layer and modulation-assisted multi-user scheduling scheme. The basic idea behind HM consists of the partitioning the data stream into two parts: the coarse or high-priority (HP) information and the refinement or low-priority (LP) information. After channel encoding, the HP and the LP information are de-multiplexed into a single stream and mapped on non-uniformly spaced constellation (Fig. 2.1.b) points creating different levels of error protection. The HP

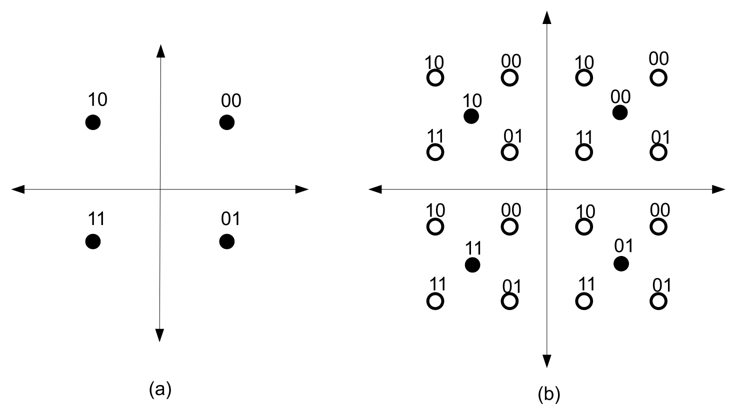

Figure 2.1: 4-QAM (a) and Hierarchical 4/16-QAM (b) constellations.

information is to be received correctly by the corresponding user even in a very bad channel environment, while the LP information is mostly designated to user whose channel has better qualities and higher signal-to-noise ratios (SNR).

The idea behind HM was first applied to digital broadcast systems [35], and it was shown that the proposed scheme improves the throughput of the system compared to the classical opportunistic scheduling. In [36], the authors proposed a multi-user scheduling algorithm and showed that HM offers lower queueing delay at the transmission buffer. This is due to the fact that by scheduling two or more users simultaneously, users can access to the channel more frequently. Actually, as stated in [37] one of the main reasons for the poor delay performance of Max-Weight algorithm is that only one queue is served at a time. This motivates us to design multi-user scheduling algorithm for Max-Weight scheduling by employing hierarchical modulation at the physical layer (Chapter 4).

In unsaturated systems, there is arrival of traffic with finite workload to each user, and queueing and packet arrival dynamics are considered. Under such systems, the primary goal is to stabilize queues (i.e., average queue size is finite). In their seminal work, Tassiulas and Ephremides have shown that opportunistic *Max-Weight* algorithm which schedules the user with the highest queue backlog and transmission rate product at every time slot can stabilize the network and achieves the maximum throughput [6].

They also establish the *achievable rate region* (also known as stability region, network capacity region) which is the convex hull of the set of all arrival rate vectors that can be supported by an appropriate scheduling policy. Next, we explain Max-Weight algorithm in detail.

The work in [6] adopts some idealized assumptions such as i.i.d. packet arrivals and channel conditions over time. These assumptions, however, do not necessarily hold in practice. Furthermore, the same analytical techniques and Max-Weight policy do not necessarily work when these assumptions are removed. Naturally, the result in [6] has been further extended by many researchers [7], [38], [39], [40]. In particular, in the context of general power allocation and routing, Neely and Modiano have extended the results in [6] and established the network layer capacity region for general ergodic channels, arrival processes, and general interference models [39]. They have also shown that Max-Weight type policy is still throughput-optimal in the general network setup.

Max-Weight type algorithms have also been studied in a variety of contexts with different objectives. For instance, Neely developed energy optimal scheduling algorithm which also satisfies the network stability for both single and multi-hop wireless networks [41]. He also studied the energy-delay [42] and utility-delay tradeoff [43], and showed that the performance of the network in terms of both energy and utility can be arbitrarily close to the optimal operating point at the expense of increasing the end-to-end delay. The scheduling policy of [6] has also inspired the congestion control [44], [45], rate control [46] problems. The work in [6] also provided the discovery of another throughput-optimal scheduling algorithm namely the exponential rule [47].

The main difficulty in implementing Max-Weight type scheduling policies for downlink network is having access to channel conditions of users. Hence, all of these papers largely rely on the availability of this information from all users, which could be prohibitively large with increasing numbers of users. There has been significant interest in developing joint channel feedback and scheduling algorithms for downlink wireless systems. Opportunistic feedback has been proposed in [48], [49], [50], [51] where the system is designed primarily for exploiting multiuser diversity. In [48] users contend for the feedback channel if the channel state exceeds a pre-defined threshold. Similarly, in [50], multiple threshold levels are used to reduce the cost for obtaining CSI. For uplink scheduling, the authors in [49] propose an optimization framework in OFDM systems. A random access based feedback protocol for achieving multiuser diversity with limited feedback was proposed in [51]. More recently, similar idea was proposed in [52] where only the users with channels good enough are allowed to send feedback. We refer to the readers to [53] and the references therein for more information on acquiring limited feedback. Most prior works study network capacity and feedback tradeoff by assuming infinitely backlogged user queues (i.e., saturated system). However, when network stability problem, where the aim is to stabilize all user queues, is considered this trade-off cannot be analyzed in the same way since queue size of each user should be taken into account. Network stability problem with infrequent channel state measurements was investigated in [54] and it was shown that achievable rate region shrinks as the frequency of CSI feedback decreases.

On the other hand, employing Max-Weight scheduling for uplink communication not only requires complete CSI but also queue length information from all users at every scheduling time, which brings much more cost than that of downlink system. Hence, the cost of Max-Weight policy has motivated many researchers to develop distributed algorithms for the practical implementation of Max-Weight policy. Carrier Sense Multiple Access (CSMA) is one of the most popular random access protocols in practice, which provide distributed algorithms. Simply, with CSMA each user senses the medium and transmits a packet only if the medium is sensed idle. Due to its simple and distributed nature, it has been widely used in current wireless networks,such as IEEE 802.11 networks. Thus, there exists a huge number of works on CSMA. One of the earliest work in this direction is proposed [55], where randomized policies based on the Pick-and-Compare is proposed. In a more recent work [56], the authors propose distributed schemes to implement a randomized policy similar to the one in [55] that can stabilize the entire rate region. Both policies in [55] and [56]; however, are proposed for time-invariant channels and require message passing. Recently, [57], [58], [59] showed that simple CSMA-type algorithms can achieve throughput-optimality without requiring any message passing between users. However, these works consider static channel model and their delay performance can be very poor.

Chapter 3

Optimal Scheduling in Cognitive Radio Network

In this Chapter, optimal scheduling policies are characterized for wireless cognitive networks under the spectrum leasing model. Such a study is motivated by the observation that these networks through dynamic spectrum access improve the current under-utilization of the spectrum. We propose cooperative schemes in which secondary users share the time slot with primary users in return for cooperation. Cooperation is feasible only if the primary system's performance is improved over the non-cooperative case. First, we investigate a scheduling problem where secondary users are interested in immediate rewards. Here, we consider both infinite and finite backlog cases. Then, we formulate another problem where the secondary users are guaranteed a portion of the primary utility, on a long term basis, in return for cooperation. Finally, we present a power allocation problem where the goal is to maximize the expected net benefit defined as utility minus cost of energy. Our proposed scheduling policies are shown to outperform non-cooperative scheduling policies, in terms of expected utility and net benefit, for a given set of feasible constraints. Based on Lyapunov Optimization techniques, we show that our schemes are arbitrarily close to the optimal performance at the price of reduced convergence rate.

3.1 Overview

Cognitive Radio Networks (CRNs) have recently been investigated extensively [60], [23]. As discussed in Chapter 2, the main advantage that CRN presents is the efficient utilization of the scarce radio spectrum resources. By opportunistically exploiting the under utilized spectrum, unlicensed (i.e., secondary) users can transmit over the licensed bands, provided that they do not *hurt* the performance of the licensed (i.e., primary) users.

One of the most important spectrum sharing model for cognitive radio is *spectrum leasing* [21], [22], where primary users (PUs) own the spectrum and are willing to lease it to SUs in return for some form of service, for instance, cooperation via relaying. Consider the following motivating scenario: in a cellular network, a licensed wireless user is far away from the base station and is experiencing low transmission rates. At the same time, a cognitive user is half way between the licensed user and the base station and thus has better channel conditions. The cognitive user desires to access the channel to send some of its own data to the base station. After coordination, PU agrees to share a portion of its own time slot with SU in exchange for SU relaying PU's data to the base station. In this Chapter, we exploit this cooperative scheme between primary and secondary systems to improve the overall performance.

Optimal scheduling in wireless networks has been extensively studied in the literature under various assumptions and purposes [61], [62], [9], [11], [10], [63], [64], [63], [64]. However, these works assumed a sing-hop system, and no cooperation among users was investigated. Opportunistic scheduling was recently studied for cognitive radio networks under the commons model [65], [66]. In these works, Lyapunov optimization tools were used to design flow control, scheduling and resource allocation algorithms and explicit performance bounds were derived. Using the technique of virtual queues, the joint problem of stabilizing the queues of SUs in addition to satisfying long term constraint on the collision probability or interference on the primary channels is transformed into a queue stability problem. In addition, cognitive radio system applying spectrum leasing approach was investigated in [28], [29], [30]. In [28], the authors presented a spectrum leasing scheme that allows PU to lease its own bandwidth for a fraction of time in exchange for enhanced performance guarantee via cooperation with SUs. As a result, more spectrum access opportunity is left for SU to transmit their own data. The authors in [29], [30] developed a game theoretic framework for a spectrum leasing in which PU actively participates in a non-cooperative game with SUs. In these works, PU plays an active role and allows SUs' access while meeting its own minimum Quality-of-Service (QoS) requirement. On the other hand, SUs aim to achieve energy efficient transmissions as long as they do not cause excessive interference to PU.

In this Chapter, we propose optimal opportunistic scheduling policies for primary *and* secondary users in a cognitive radio network under the spectrum leasing model. Unlike previous work, we consider scheduling of cooperative primary and secondary networks with multiple users sharing a common destination. For example, [28], [29], [30] considered only one primary transmitter and separate receivers for primary and secondary systems. Thus, the only coordination required is among the transmission between the single PU and a subset of SUs. In addition, the authors in [29], [30] did not explicitly model the price paid by SUs to PUs to share the licensed spectrum. In our work, we first consider the optimization of the total expected utility of both primary and secondary systems while satisfying an average performance constraint for each primary user in the network. Here, we develop a cooperative scheduling policy by which the performance is improved and shown to be *at least* as good as the original primary-only system. For a given time slot, users cooperate using decode-and-forward multihop scheme [67] where SUs relay the messages of PUs to a common destination in a portion of the time slot as a levy of using the already licensed spectrum for a fraction of that time slot. The parameters specifying the cooperation strategy are the fraction of the time slot during which SU relays PU's data and the fraction used to transmit SU's own data.

Next, another formulation is considered in which SUs are guaranteed some portion of the primary utility in an average sense, in return for cooperation. This formulation presents a model of *banking* between primary and secondary systems where rewards are gained over the long term. Finally, we formulate a power control problem where the objective is to maximize the net benefit defined as the difference between the value of the utility and the cost of the energy consumption, under minimum requirement constraints on PUs. We employ Lyapunov optimization tools developed in [68], [8] to analyze our proposed schemes and to derive explicit bounds on the performance achieved. We show that our proposed schemes can be pushed arbitrarily close to the optimal with a tradeoff between optimality and the convergence rate of the algorithms.

3.2 Network Model

We consider a cognitive radio network of M PUs and N SUs, all wishing to communicate with a common destination D as shown in Figure 3.1. This destination can be viewed as a base station in a single-cell of a cellular network or as an access point in a Wi-Fi network. We consider a time-slotted system where the time slot is the resource to be

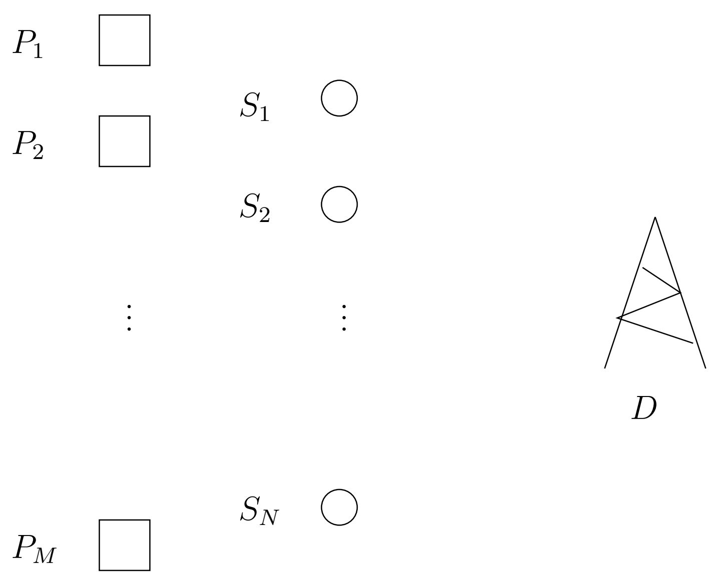

Figure 3.1: Network Model.

shared among different nodes. We adopt a non-interference model where only one node, either primary or secondary, is transmitting at any given time. Random channel gains between each node and other nodes in the network are assumed to be independent and identically distributed (i.i.d) across time according to a general distribution and independent across nodes with values taken from a finite set. Moreover, we assume that channel gains are time-varying, but fixed over the time slot duration. We assume the availability of perfect channel state information of all channels at the scheduler, i.e., knowledge of channel coefficients immediately prior to transmission.

In the following analysis, we use the notation $R^p_m(t)$, $R^s_n(t)$ to denote the transmission rates from PU m to destination and from SU n to destination, respectively, at time slot t. The corresponding random rate vectors are denoted as $\mathbf{R}^p(t)$, $\mathbf{R}^s(t)$. The transmission rate from PU m to SU n is denoted as $R^r_{mn}(t)$, where the corresponding rate matrix is $\mathbf{R}^r(t)$. The transmission rate is a function of the random channel conditions, and thus a measure of the channel quality. We assume that transmission rate processes are ergodic and bounded. As will be clear in the next subsection, since our scheme works by selecting a pair of users (primary and secondary) to transmit at a given time slot, the utility achieved by a user is a function of the cooperating pair. Consequently, the utility function of a PU m when it cooperates with SU n at time slot t is denoted as $U_{mn}(t)$. Similarly, the utility function of a SU n that cooperates with PU m is denoted as $V_{mn}(t)$. These utility functions are measures of the level of satisfaction of users and thus they are generally assumed to be non-decreasing concave functions of the transmission rate.

3.2.1 Cooperative Scheme

To schedule transmissions of different users, a scheduling policy is required. In our cooperative framework, we allow the scheduling policy to either schedule a PU to transmit during a given time slot, or to schedule a pair of primary and secondary users to share the time slot, according to the channel conditions. The scheduling policy Q is a rule that selects the four-tuple (m, n, α, β) to transmit at time-slot t, where α and β specify the cooperation strategy the pair of primary and secondary users m, n use. In a time slot t, the scheduling policy is a function of the rate vectors $\mathbf{R}^p(t)$, $\mathbf{R}^s(t)$, rate matrix $\mathbf{R}^r(t)$ and possibly other variables related to past performance. Note that the scheduling policy we adopt is opportunistic in the sense that it exploits the time-varying nature of the wireless channel.

In our model, we focus on a cooperation based spectrum leasing scenario. Under this model, scheduling is done such that, if feasible, a pair of primary and secondary users cooperatively share a single time slot to improve the performance of the original primary system and allow unlicensed users to access the licensed spectrum, where feasibility is to be defined. Cooperation is achieved as follows: for a fraction $(1-\alpha)$, $0 \leq \alpha \leq 1$, of the time slot, PU m sends its data (intended to destination) to SU n (relay). In the remaining portion of the time slot, the scheduled SU uses the channel to relay PU's data over a β fraction, $0 \leq \beta \leq 1$, and then, transmits its own data during the rest of the time slot, i.e., over $\alpha(1-\beta)$ fraction. A schematic of the time slot structure is shown in Figure 3.2. This cooperative scheme is a form of implementation of the spectrum leasing cognitive radio framework where SUs help primary system improve its performance to access the licensed spectrum. By this scheme, our system is in fact trying to reap the benefits of a form of spatial diversity. We note that the structure of our scheme is similar to the cooperative scheme of [28], however, we do not employ distributed space time coding and allow only one SU to cooperate in a given time slot. We set $n = 0$ by definition for the case when a PU m is scheduled to transmit

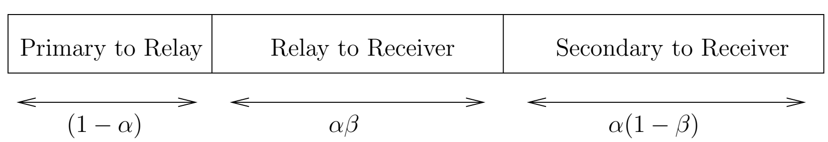

Figure 3.2: General Time Slot Structure.

directly to the destination without cooperating with SUs. This is the case when cooperation is either infeasible or leads to suboptimal utility values. We set $R^r_{m0}(t) = R^p_m(t)$, $m \in \{1, 2, \cdots, M\}$ and $\alpha = 0$ in such cases.

The utility function is taken to be a non-negative non-decreasing concave function of the rate. This choice is of practical interest, since a small increase in the rate in the low rate regime is generally more appreciated than a small increase in the high rate regime. Given a scheduling decision $Q = (m', n', \alpha, \beta)$, we define the utility of the selected primary and secondary users, $U_{m'n'}(Q,t)$ and $V_{m'n'}(Q,t)$, respectively, as

$$U_{m'n'}(Q,t) = \begin{cases} h_1(R^p_{m'}(t)) & ; \text{if } n' = 0 \\ h_1((1-\alpha)R^r_{m'n'}(t)) & ; \text{otherwise} \end{cases} \tag{3.1}$$

$$V_{m'n'}(Q,t) = \begin{cases} 0 & ; \text{if } n' = 0 \\ h_2(\alpha(1-\beta)R^s_{n'}) & ; \text{otherwise} \end{cases} \tag{3.2}$$

in a given time slot t. For all other primary and secondary users (m,n) such that $m \neq m'$ and $n \neq n'$, we set $U_{mn}(Q,t) = V_{mn}(Q,t) = 0$. In the following, we sometimes use the shorthand $U_{mn}(t)$ and $V_{mn}(t)$ in place of $U_{mn}(Q,t)$ and $V_{mn}(Q,t)$ for simplicity. Examples of utility functions that can be used include $h_i(x) = \log(1+x)$ and $h_i(x) = x,\ i \in \{1,2\}$.

Note that we assume scheduler's knowledge of the transmission rates for the primary and secondary users at each time slot. For the scheduler to choose a pair to transmit over a given slot rather than scheduling a PU for direct transmission, feasibility conditions should hold. For a time slot t, the feasibility conditions can be summarized as follows:

$$0 < (1-\alpha)R^r_{mn}(t) \leq \alpha R^s_n(t). \tag{3.3}$$

The strict inequality in (3.3) guarantees validity of the cooperation whereas the second inequality asserts that SU n has a sufficiently good channel to relay primary transmission at a given time slot t. Given α, it can be seen that the optimal value of β is given by

$$\beta^* = \frac{(1-\alpha)R^r_{mn}}{\alpha R^s_n}. \tag{3.4}$$

If $\beta < \frac{(1-\alpha)R^r_{mn}}{\alpha R^s_n}$, SU n does not have sufficient time to relay the data of m. If $\beta > \frac{(1-\alpha)R^r_{mn}}{\alpha R^s_n}$, then unnecessary time is wasted by SU n. Thus, in the following we use the notation $Q = (m,n,\alpha)$ for the decision of a scheduling policy Q. Note that (3.3) implies $\beta \leq 1$.

Since we are interested in the maximization of the total expected utility of both primary and secondary systems, (3.4) is required to ensure superiority over non-cooperative schemes as will be clear in Section 3.3.

Let $\mathcal{F}$ be the set of feasible policies at a given time slot. The set $\mathcal{F}$ is constructed from all the tuples (m, n, α) such that (3.3) holds for some $0 \leq \alpha \leq 1$. We set the tuple $(m, 0, 0) \in \mathcal{F}$ by definition.

Let the total utility of the system (both primary and secondary), when scheduling policy Q is employed at a given time slot t, be $W(Q, t)$. Then,

$$W(Q, t) = \sum_{m=1}^{M} \sum_{n=0}^{N} U_{mn}(Q, t) + V_{mn}(Q, t). \tag{3.5}$$

Note that when the scheduling policy Q selects the tuple (m', n', α), the system receives a reward of $W(Q, t) = U_{m'n'}(Q, t) + V_{m'n'}(Q, t)$. The total expected utility is defined as $\bar{W}(Q, t) \triangleq \mathbb{E}[W(Q, t)]$ where the expectation is taken over the random transmission rates (random channel conditions), and possibly over the randomized policy.

3.2.2 Lyapunov Drift with Optimization

In our work, we use Lyapunov drift and optimization tools to show the optimality of our schemes. The advantage of this tool is the ability to provide a simple way to find optimal scheduling algorithms for complex models and to prove their optimality. Basically, this simplicity comes from defining each constraint as a virtual queue and then transforming the problem into a network stability problem [68].

We first introduce two definitions: Let $Z_i(t)$, $i \in \{1, 2, \cdots, L\}$ be a queue backlog process and $\mathbf{Z}(t) = (Z_1(t) Z_2(t) \cdots Z_L(t))$ in a network with L nodes. Suppose that the goal is to stabilize the backlog process $\mathbf{Z}(t)$ while maximizing the time average of a scalar-valued utility function $g(\cdot)$ of another process $\mathbf{R}(t)$. Suppose that the optimal value of $g(\cdot)$ is g^*. Define the following quadratic Lyapunov function and conditional

Lyapunov drift

$$L(\mathbf{Z}(t)) \triangleq \sum_{l=1}^{L} Z_l^2(t), \tag{3.6}$$

$$\Delta(\mathbf{Z}(t)) \triangleq \mathbb{E}\left[L(\mathbf{Z}(t+1)) - L(\mathbf{Z}(t))|\mathbf{Z}(t)\right]. \tag{3.7}$$

We give a modified version of Theorem 2.1.

Theorem 3.1. *(Lyapunov Optimization) [8] For the scalar valued function $g(\cdot)$, if there exists positive constants K, ϵ, B, such that for all time slots t and all unfinished work vectors $\mathbf{Z}(t)$ the Lyapunov drift satisfies the condition*

$$\Delta(\mathbf{Z}(t)) - K\mathbb{E}[g(\mathbf{R}(t))|\mathbf{Z}(t)] \leq B - \epsilon \sum_{l=0}^{L} Z_l(t) - Kg^*,$$

then the time average utility and queue backlog satisfy:

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[g(\mathbf{R}(\tau))] \geq g^* - \frac{B}{K}, \tag{3.8}$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{l=1}^{L} \mathbb{E}[Z_l(\tau)] \leq \frac{B + K(\bar{g} - g^*)}{\epsilon}, \tag{3.9}$$

where $\bar{g} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[g(\mathbf{R}(\tau))]$.

We note that Theorem 3.1 is a modified version of Theorem 2.1. Specifically, in our analysis, the function $g(\cdot)$ represents the total utility of the system in a time slot given by (3.5) which is function of the utility matrices (3.1), (3.2) and the scheduling policy Q. Hence, our objective is to maximize the time average of the instantaneous utilities. On the other hand, in [8], the utility of each user is a function of the time average of the instantaneous rates. However, by using Jensen's inequality and noting that the utility function is concave, the same lower bounds in (3.8) and (3.9) apply for our objective. We note that by using T-slot Lyaponuv drift techniques [8], similar results can be derived for more general (i.e., correlated) channel processes.

In our problem, since the utility of individual primary and secondary users is bounded, since achievable rates are bounded the total utility $W(Q,t)$ is bounded. It follows that the total expected utility can be pushed arbitrarily close to the optimum by choosing K sufficiently large. However, this leads to increasing bound on the average queue size given in (3.9).

3.3 Primary Constraints and Immediate Rewards

In this section, the goal is to schedule the transmissions of primary and secondary nodes to achieve maximum average sum utility of primary and secondary systems while maintaining minimum performance levels for each primary node. Here, the secondary node n is allowed to access the spectrum only if cooperation improves the instantaneous utility of a primary node m Hence, we define $\mathcal{F}_1$ as the set of tuples (m,n,α) satisfying the following condition:

$$R_m^p(t) \leq (1-\alpha)R_{mn}^r(t) \leq \alpha R_n^s(t). \tag{3.10}$$

for some $0 < \alpha < 1$. This constraint sets an upper bound on the range of α for the possible cooperation between each pair (m,n). Note that $\mathcal{F}_1 \subset \mathcal{F}$. We discuss two types of scheduling policies. First, we consider stationary scheduling polices that depend only on the values of the rates $\mathbf{R}^p(t), \mathbf{R}^s(t), \mathbf{R}^r(t)$. Then we investigate the more general time-varying policies.

3.3.1 Problem Formulation

The optimal opportunistic scheduling problem with minimum performance constraints was previously solved in [9]. By including N SUs to the system, our model can be viewed as a generalization to the model in [9]. In addition, setting $N=0$ in our scheme yields the scheme in [9] as will be shown in the next subsection.

The problemcan be stated formally as follows:

$$\max_{Q \in \mathcal{F}_1} \bar{W}(Q,t)$$
$$\text{s.t. } \mathbb{E}\left[\sum_{n=0}^{N} U_{mn}(Q,t)\right] \geq C_m, \tag{3.11}$$

$m \in \{1,2,\ldots,M\}$, where C_m is the minimum performance constraint for each PU m. To compare to the non-cooperative system, an example of the choice of the constraints C_m is given at the end of Section 3.3.2.

The aforementioned problem formulation along with (3.10) implies that SUs are rewarded access to the channel immediately during a time slot if their cooperation improves the performance of the primary system.

3.3.2 Optimal Stationary Policy

In this subsection, we propose a stationary scheduling policy in a form similar to the optimal policies reported in [9], and show that it solves (3.11) for the given cognitive radio network.

Scheduling Algorithm Q_{1a}:

For every time slot t and the given the values of $U_{mn}(t)$ and $V_{mn}(t)$ for all (m,n), the solution to the scheduling problem (3.11) is given by

$$Q_{1a} = \underset{(m,n,\alpha)\in\mathcal{F}_1}{\operatorname{argmax}} \left\{\lambda_m^* U_{mn}(Q,t) + V_{mn}(Q,t)\right\}, \tag{3.12}$$

where $\lambda_m^*, m \in \{1,2,\ldots,M\}$ are real-valued parameters satisfying:

1. $\min_m \lambda_m^* = 1$.
2. $\mathbb{E}\left[\sum_{n=0}^{N} U_{mn}(Q_{1a},t)\right] \geq C_m$ for all m.
3. If $\mathbb{E}\left[\sum_{n=0}^{N} U_{mn}(Q_{1a},t)\right] > C_m$, then $\lambda_m^* = 1$.

Theorem 3.2. *Scheduling Algorithm* Q_{1a} *solves* (3.11).

Proof. The proof is similar to the proof of the optimal policies in [9]. However, the scheduling policy Q_{1a} in our work decides a tuple of three variables each time slot instead of only choosing a primary user.

In the following, we drop the parameter t. Let Q be a scheduling policy satisfying $\mathbb{E}\left[\sum_{n=0}^{N} U_{mn}(Q)\right] \geq C_m$ for all $m \in \{1, 2, \ldots, M\}$. Then,

$$\begin{aligned}
\bar{W}(Q) \leq \bar{W}(Q) &+ \sum_{m=1}^{M} (\lambda_m^* - 1)\left(\sum_{n=0}^{N} \mathbb{E}\left[U_{mn}(Q)\right] - C_m\right) \\
&= \sum_{m=1}^{M}\sum_{n=0}^{N} \mathbb{E}[U_{mn}(Q) + V_{mn}(Q)] \\
&\quad + \sum_{m=1}^{M}\sum_{n=0}^{N} (\lambda_m^* - 1)\mathbb{E}[U_{mn}(Q)] \\
&\quad - \sum_{m=1}^{M} (\lambda_m^* - 1)C_m \\
&= \sum_{m=1}^{M}\sum_{n=0}^{N} \mathbb{E}[\lambda_m^* U_{mn}(Q) + V_{mn}(Q)] \\
&\quad - \sum_{m=1}^{M} (\lambda_m^* - 1)C_m
\end{aligned}$$

where the first inequality follows since $\lambda_m^* \geq 1$. From the definition of Q_{1a}, we have

$$\begin{aligned}
&\sum_{m=1}^{M}\sum_{n=0}^{N} \mathbb{E}[\lambda_m^* U_{mn}(Q) + V_{mn}(Q)] \\
&\leq \sum_{m=1}^{M}\sum_{n=0}^{N} \mathbb{E}[\lambda_m^* U_{mn}(Q_{1a}) + V_{mn}(Q_{1a})]
\end{aligned}$$

□

The structure of the derived scheduling policy suggests that when a primary user m experiences unfavorable channel conditions, the associated parameter λ_m^* will be larger than one. Then, it attains average utility that is only equal to its corresponding constraint. Otherwise, this primary user is granted a utility strictly larger than its minimum requirement.

The policy in (3.12) is stationary since it only depends on the values of the utility functions. Note that for any time slot t, given the values of $R^p_m(t)$, $R^s_n(t)$ and $R^r_{mn}(t)$ for all m and n, the scheduler is able to construct the set of feasible policies $\mathcal{F}_1$ by associating the ranges $r^-_\alpha \leq \alpha \leq r^+_\alpha$ and $r^-_\beta \leq \beta \leq r^+_\beta$ for each pair (m, n). These ranges are chosen to satisfy the feasibility conditions (3.10), where $0 \leq r^-_\alpha, r^+_\alpha, r^-_\beta, r^+_\beta \leq 1$. Then it decides which pair (or single primary user) are relatively best according to (3.12). The choice of the pair (m, n) is a combinatorial optimization problem which may require discrete exhaustive search. The optimal value of α can be obtained since (3.12) can be shown to be concave in α. The parameters $\lambda^*_m, m \in \{1, 2, \ldots, M\}$ depend on the choice of $h_1(.), h_2(.)$ and the distribution of the utility functions which in turn depends on the distribution of the underlying channel variations. Hence, λ^*_m needs to be estimated online in practice. This can be carried out using stochastic approximation techniques similar to the one explained in [9]. An estimation technique is presented in Section 3.6.

Example: The above algorithm can be compared to non-cooperative algorithms as follows. Consider for example the utilitarian fairness constraints problem solved in [9] with the constraints $a_m = \gamma_m \bar{W}(\hat{Q})$ for each PU $m \in \{1, 2, \ldots, M\}$ where $\sum_{m=1}^{M} \gamma_m \leq 1$ and $\bar{W}(\hat{Q})$ is the average performance achieved under the optimal (primary-only) scheduling policy $\hat{Q}$. According to this definition of a_m, the problem is always feasible. Let Γ be the improvement factor with respect to the system with no cooperation. Note that SUs can have access to the channel if they help improving the performance of primary system. Thus, PUs achieve utility which is at least the same as that in the non-cooperative case (i.e., $\Gamma \geq 1$). On the other hand, Γ is upper bounded such that $\Gamma \leq \Gamma_{max}$ due to the capacity region constraint (see Section 3.3.4). Hence, $1 \leq \Gamma \leq \Gamma_{max}$.

Now consider a network of M primary and N secondary users such that the scheduler executes the optimal policy $\hat{Q}$ to schedule only the PUs but does not act on it and simultaneously executes and implements our cooperative scheduling policy Q_{1a}. Since the scheduling policy $\hat{Q}$ converges as the number of time slots $t \to \infty$ [9], we can set $C_m = \Gamma a_m$ in (3.11). As long as $1 \leq \Gamma \leq \Gamma_{max}$, it follows that our cooperative scheme improves the performance of individual PUs over the non-cooperative scheme, and hence improves the overall performance.

3.3.3 Optimal Time Varying Policy

In this subsection, we solve problem (3.11) using the stochastic network optimization tool of [8]. This tool yields a scheduling policy that is similar in structure to (3.12). However, the policy derived in this subsection does not need the computation of the online parameters λ_m^*.

Define the time average expected utility as follows.

$$\bar{W}(Q) = \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}[W(Q,\tau)], \tag{3.13}$$

where $W(Q,t)$ is defined in (3.5).

Let $U_m^t(Q,t) \triangleq \sum_{n=0}^{N} U_{mn}(Q,t)$, $V_n^t(Q,t) \triangleq \sum_{m=1}^{M} V_{mn}(Q,t)$. For each of the constraints in (3.11), we construct a virtual queue such that the queue dynamics is given by

$$X_m(t+1) = [X_m(t) - U_m^t(Q,t)]^+ + C_m, \tag{3.14}$$

$m \in \{1,2,\ldots,M\}$, where $[x]^+ \triangleq \max\{x,0\}$. Note that stabilizing the queues in (3.14) is equivalent to satisfying the constraints in (3.11) since a queue is stable if the arrival rate is less than the service rate. We assume that $U_m^t(Q,t)$ and $V_n^t(Q,t)$ are bounded such that $U_m^t(Q,t) \le U^{max}$, $V_n^t(Q,t) \le V^{max}$ for all $m \in \{1,2,\ldots,M\}$, $n \in \{1,2,\ldots,N\}$, $t \ge 0$ and for all $Q \in \mathcal{F}_1$. These upper bounds are justified since we assume bounded transmission rates. Let $\mathbf{X}(t) = (X_1(t)X_2(t)\cdots X_M(t))$ be the vector of virtual queues. Define the following quadratic Lyapunov function and conditional Lyapunov drift:

$$L_1(\mathbf{X}(t)) \triangleq \sum_{m=1}^{M} X_m^2(t), \tag{3.15}$$

$$\Delta_1(\mathbf{X}(t)) \triangleq \mathbb{E}\left[L_1(\mathbf{X}(t+1)) - L_1(\mathbf{X}(t))|\mathbf{X}(t)\right]. \tag{3.16}$$

Define the following conditional expectation:

$$\bar{U}^t_m(Q,t) \triangleq \mathbb{E}[U^t_m(Q,t)|\mathbf{X}(t)]. \tag{3.17}$$

The following Lemma is useful in establishing the optimality of our algorithm.

Lemma 3.1. *For every time slot t and any policy Q, the Lyapunov drift in* (3.16) *can be upper bounded as follows:*

$$\begin{aligned}\Delta_1(\mathbf{X}(t)) - K\mathbb{E}[W(Q,t)|\mathbf{X}(t)] &\leq B_1 + 2\sum_{m=1}^{M} X_m(t)C_m \\ -\sum_{m=1}^{M}(K+2X_m(t))\bar{U}^t_m(Q,t) &- \sum_{n=1}^{N} K\bar{V}^t_n(Q,t),\end{aligned} \tag{3.18}$$

where $B_1 = \sum_{m=1}^{M} C_m^2 + M(U^{max})^2$ and K is a system parameter that characterizes a tradeoff between performance optimization and delay in the virtual queues.

Proof. We use the simplified notation $\bar{U}^t_m$ in place of $\bar{U}^t_m(Q,t)$. From the dynamics of the virtual queues (3.14), we can write

$$X_m^2(t+1) \leq X_m^2(t) + C_m^2 + (U^t_m(t))^2 - 2X_m(t)[U^t_m(t) - C_m]$$

for $m \in \{1,2,\ldots,M\}$, where the above inequality follows from the fact that $\left([a]^+\right)^2 \leq (a)^2 \forall a$. Therefore, the Lyapunov drift in (3.16) can be upper bounded as

$$\begin{aligned}\Delta_1(\mathbf{X}(t)) \leq &\sum_{m=1}^{M} C_m^2 + \mathbb{E}[(U^t_m(t))^2|\mathbf{X}(t)] - 2X_m(t)\bar{U}^t_m \\ &+ 2X_m(t)C_m\end{aligned}$$

Using the bounds on the utility functions U^{max}, we have

$$\Delta_1(\mathbf{X}(t)) \leq B_1 + 2\sum_{m=1}^{M} X_m(t)(C_m - \bar{U}^t_m)$$

where $B_1 = \sum_{m=1}^{M} C_m^2 + M(U^{max})^2$. Subtracting the term $K\mathbb{E}[W(Q,t)|\mathbf{X}(t)]$ from both sides, expanding terms, rearranging terms and using $V_0^t(t) = 0 \; \forall \; t$, (3.18) follows. $\square$

Now, we present our opportunistic scheduling algorithm that involves cooperation between primary and secondary nodes to achieve better performance.

Scheduling Algorithm Q_{1b}: At each time slot t, observe the virtual queue backlog $X_m(t)$ for each primary node m and the achievable transmission rates, and choose (m, n, α) solving the following optimization problem.

$$Q_{1b} = \underset{(m,n,\alpha)\in\mathcal{F}_1}{\text{argmax}} \left\{ \left(1 + \frac{2X_m(t)}{K}\right) U_{mn}(t) + V_{mn}(t) \right\} \tag{3.19}$$

Then, update the virtual queues according to the queue dynamics in (3.14).

Note that we assume knowledge of the utility functions and channel states at the scheduler at each time slot. Hence, the queue states are known constants in the above optimization problem. Comparing to Algorithm Q_{1a}, let $\tilde{\lambda}_m(t) \triangleq 1 + \frac{2X_m(t)}{K} \geq 1$. It is clear that both algorithms have exactly the same form. However, contrary to the algorithm in Section 3.3.2, Q_{1b} does not require the knowledge of the statistics of the channel states or need the computation of online parameters.

Compared to the non-cooperative scheme in [9] which requires M multiplications, it can be seen that Q_{1b} requires $2M(N + 1)$ operations ($M(N + 1)$ multiplications and $M(N + 1)$ additions). In addition, an algorithm to compute the best α for each pair (m, n) is needed in our scheme. However, the complexity of our scheme can be reduced by allowing the base station to select only a subset of available SUs with strong channels to be considered in scheduling.

We analyze our algorithm using the Lyapunov drift with optimization [8]. We define a class of policies that will be useful to prove the optimality of the scheduling algorithm Q_{1b}. Consider the class of scheduling algorithms $\mathcal{S}$ that schedules nodes according to a stationary and possibly randomized function of only the achievable rates and independent of the queue states. It was shown in [8, 68] that the optimality is achieved within the class of stationary policies $\mathcal{S}$, for a large class of network flow problems including fairness problems. Since the channel states are chosen from a finite set and the set $\{(m, n, \alpha) \mid m \in \{1, 2, \cdots, M\}, n \in \{1, 2, \cdots, N\}, \alpha \in [0, 1]\}$ is closed and bounded, we have the following lemma (which can be proved using similar arguments as in [68]). Let the feasibility region of (3.11) be Λ and let $\boldsymbol{\epsilon} \triangleq (\epsilon \; \epsilon \cdots \; \epsilon)$.

Lemma 3.2. *If the vector* $\mathbf{C} = (C_1 \; C_2 \cdots C_M)$ *is feasible (i.e.,* $\mathbf{C} \in \Lambda$*), then there exists a stationary randomized policy* Q_{s_1} *that solves* (3.11) *and satisfies the following:*

$$\mathbb{E}[W(Q_{s_1}, t)] = \bar{W}_1^*, \tag{3.20}$$

$$\mathbb{E}\left[U_m^t(Q_{s_1}, t)\right] \geq C_m, m \in \{1, 2, \ldots, M\}, \tag{3.21}$$

where $\bar{W}_1^*$ *is the optimal performance for the problem* (3.11) *over all scheduling policies. Moreover, if* $\mathbf{C}$ *is strictly interior to* Λ*, then there exists* $\epsilon > 0$ *such that* $(\mathbf{C}+\boldsymbol{\epsilon}) \in \Lambda$ *and a stationary scheduling policy* $Q_{s_1(\epsilon)}$ *satisfying:*

$$\mathbb{E}\left[U_m^t(Q_{s_1(\epsilon)}, t)\right] \geq C_m + \epsilon, m \in \{1, 2, \ldots, M\} \tag{3.22}$$

with an optimal total average utility $\bar{W}_1^*(\epsilon)$ *such that* $\bar{W}_1^*(\epsilon) \leq \bar{W}_1^*$ *where* $\bar{W}_1^*(\epsilon) \to \bar{W}_1^*$ *as* $\epsilon \to 0$.

We are now ready to present bounds on the performance of our proposed algorithm Q_{1b}. The following Theorem shows that all the virtual queues are strongly stable [8]. Hence, all time average constraints in (3.11) are satisfied.

Theorem 3.3. *If* $\mathbf{C}$ *is strictly in the interior of* Λ*, then the proposed algorithm in* Q_{1b} *stabilizes the virtual queues and achieves the following bounds:*

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[W(Q_{1b}, \tau)] \geq \bar{W}_1^* - \frac{B_1}{K}, \tag{3.23}$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[\sum_{m=1}^{M} X_m^2(\tau)\right] \leq \frac{B_1 + KW^{max}}{\epsilon_{max}}, \tag{3.24}$$

where $W^{max} = U^{max} + V^{max}$ *and* $B_1 = \sum_{m=1}^{M} C_m^2 + M(U^{max})^2$ *and* ϵ_{max} *is the largest* ϵ *such that* $\mathbf{C}+\ \boldsymbol{\epsilon} \in \Lambda$.

Proof. Consider the upper bound given by Lemma 3.1. From Lemma 3.2, there exists a stationary policy $Q_{s_1(\epsilon)}$ that satisfies the constraints (3.22). By definition of Q_{1b},

$RHS_{Q_{1b}} \leq RHS_{Q_{s_1(\epsilon)}}$ where RHS_Q is the right hand side (RHS) of inequality (3.18) evaluated for the policy Q. Now consider evaluating $RHS_{Q_{s_1(\epsilon)}}$ using (3.22). Expanding the RHS of (3.18) and using the property that the utility is independent of queue states, it is straightforward to see that $RHS_{Q_{s_1(\epsilon)}} = B_1 - 2\epsilon \sum_{m=1}^{M} X_m(t) - K\bar{W}_1^*(\epsilon)$. It follows that

$$\begin{aligned}\Delta_1(\mathbf{X}(t)) - K\mathbb{E}[W(Q_{1b}, t)|\mathbf{X}(t), \mathbf{Y}(t)] &\leq RHS_{Q_{1b}} \\ \leq RHS_{Q_{s_1(\epsilon)}} = B_1 - 2\epsilon \sum_{m=1}^{M} X_m(t) &- K\bar{W}_1^*(\epsilon),\end{aligned}$$

which is in exactly the same form of the condition in Theorem 3.1. Applying the result of Theorem 3.1, we have the following bounds

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[W(Q_{1b}, \tau)] \geq \bar{W}_1^*(\epsilon) - \frac{B_1}{K}, \tag{3.25}$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[\sum_{m=1}^{M} X_m^2(\tau)\right] \leq \frac{B_1 + KW^{max}}{2\epsilon}, \tag{3.26}$$

where (3.26) follows since $0 \leq W(Q, t) \leq W^{max}$ for all Q. The choice of ϵ affects the bounds only and does not affect the policy Q_{1b}. Therefore, (3.25) and (3.26) can be optimized separately. Taking $\epsilon \to 0$ in (3.25) yields (3.23) and taking $\epsilon = \frac{\epsilon_{max}}{2}$ in (3.26) yields (3.24), concluding the proof. □

From (3.24) and (3.23), it is clear that the parameter K specifies a tradeoff between optimality and the average length of the virtual queues. Thus, for large virtual queues, the system experiences larger transient times to achieve the optimal performance and hence needs more time to adapt to possible changes in channel statistics [8].

3.3.4 A Note on Feasibility

In the algorithms developed in Sections 3.3.2 and 3.3.3, we assumed the feasibility of the set of constraints on the primary users' performance. In fact, the feasibility region characterization depends on the statistics of the channel conditions. Since our scheduling schemes can only improve the performance of the primary-only network as a special case, the feasibility region given in [9] is strictly a subset of the feasibility region of our policy. In addition, it can be shown, using similar techniques as in [9], that our feasibility region is convex. Specifically, the region is a subset of an M-dimensional space such that the vertex on the mth axis is $(0\ 0 \cdots \mathbb{E}[\tilde{U}_m] \cdots 0)$, where $\mathbb{E}[\tilde{U}_m]$ is the average utility achieved by applying our cooperative algorithm on a network composed of only the mth primary node in addition to N secondary nodes.

Considering the example presented in Section 3.3.2, Γ_{max} specified the maximum gain the cooperative system can achieve over the non-cooperative counterpart. It is clear that Γ_{max} can be characterized by the boundary of the feasibility region. More specifically, if $(a_1\ a_2 \cdots a_M)$ is the performance vector in the non-cooperative system defined in Subsection 3.3.2, and if the feasibility region of our cooperative system is Λ, then Γ_{max} is given by:

$$\begin{aligned} \Gamma_{max} = \max_{\Gamma \geq 1}\ & \Gamma \\ \text{s.t. }\ & (\Gamma a_1\ \Gamma a_2 \cdots \Gamma a_M) \in \Lambda. \end{aligned} \tag{3.27}$$

The solution to (3.27) can be determined numerically if the channel statistics are known. A more rigorous characterization of the feasibility region is beyond the scope of this work and is part of our future work.

3.3.5 The Downlink Case

The analysis and algorithms developed in the work are mainly focusing on the uplink scenario. However, it can be easily seen that the same analysis and results can be applied in the downlink case. In this subsection, we state the differences for the sake of completeness. In the downlink case and when cooperation is feasible, the base station (or access point) D uses the first $(1-\alpha)$ fraction of the time slot to transmit to a secondary node n. Then, the secondary node relays data to the scheduled primary node m during the next $\alpha\beta$ fraction. The rest of the time slot is dedicated to the downlink of the secondary node's data. We denote the achievable transmission rates from base station to primary and secondary nodes m and n as $\tilde{R}_m^p(t)$ and $\tilde{R}_n^s(t)$, respectively, at time slot t. We also denote the achievable rate from secondary node n to primary node m as $\tilde{R}_{mn}^r$. Similar to the uplink case, the utility functions for a scheduled pair (m, n) are defined as

$$\tilde{U}_{mn}(Q,t) = \begin{cases} h_3(\tilde{R}_m^p(t)) & ; \text{if } n = 0 \\ h_3((1-\alpha)\tilde{R}_n^s(t)) & ; \text{otherwise} \end{cases} \tag{3.28}$$

$$\tilde{V}_{mn}(Q,t) = \begin{cases} 0 & ; \text{if } n = 0 \\ h_4(\alpha(1-\beta)\tilde{R}_n^s) & ; \text{otherwise} \end{cases} \tag{3.29}$$

where $h_3(\cdot)$ and $h_4(\cdot)$ are some concave functions. Feasibility regions can be defined as follows. For immediate rewards problem, it can be seen that optimal scheduling policy selects the tuple (m, n, α) from $\mathcal{F}_{d_1}$ satisfying

$$\tilde{R}_m^p(t) \leq (1-\alpha)\tilde{R}_n^s(t) \leq \alpha\tilde{R}_{mn}^r(t) \tag{3.30}$$

for some $0 < \alpha < 1$. For the long term rewards problem, the feasibility set $\mathcal{F}_d$ is constructed from all tuples (m, n, α) such that

$$0 < (1-\alpha)\tilde{R}_n^s(t) \leq \alpha\tilde{R}_{mn}^r(t) \tag{3.31}$$

for some $0 \leq \alpha \leq 1$, where

$$\beta^* = \frac{(1-\alpha)\tilde{R}_n^s}{\alpha\tilde{R}_{mn}^r} \tag{3.32}$$

for both cases.

3.3.6 The Finite Backlog Case

In Sections 3.3.2 through 3.3.5, we determined optimal stationary and time-varying policies solving (3.11) with the assumption of all nodes having infinite backlogs. In this section, we investigate the solution of the same problem when secondary nodes have *finite* backlogs. The main difference in this case is that secondary nodes will not be willing to cooperate to relay primary nodes' data if they do not have sufficient data of their own to transmit to the base station. Hence, the achievable utilities for the infinite backlogs case constitute an upper bound for the finite backlogs case. Also note that we do not further elaborate on a system when both primary and secondary nodes have finite backlogs, since the analysis of this more complicated model provides little additional insight.

Let $A_n(t)$ and $L_n(t)$ be the amount of data arriving to secondary node n and the current backlog of the same node at time t, respectively. Under scheduling decision $Q = (m', n', \alpha)$, the time evolution of secondary node backlog $L_n(t)$, $n \in \{1, 2, \cdots, N\}$ is given as follows:

$$L_n(t+1) = \begin{cases} \max\{L_n(t) - \alpha(1-\beta)R_n^s(t), 0\} + A_n(t) & ; \text{if } n = n' \\ L_n(t) + A_n(t) & ; \text{otherwise} \end{cases} \tag{3.33}$$

Unlike the infinite backlog case, when scheduled, secondary node n' can transmit only $\min\{L_{n'}(t), \alpha(1-\beta)R_{n'}^s(t)\}$ amount of data at time slot t. Hence, under scheduling decision $Q = (m', n', \alpha)$ the utility of secondary node with finite backlogs is modified as follows:

$$\hat{V}_{m'n'}(Q, t) = \begin{cases} 0 & ; \text{if } n' = 0 \\ h_2(\min\{L_{n'}(t), \alpha(1-\beta)R_{n'}^s(t)\}) & \\ & ; \text{otherwise} \end{cases} \tag{3.34}$$

The following scheduling algorithm opportunistically determines a primary-secondary node pair.

Scheduling Algorithm Q_{1c}**:** At each time slot t, observe the virtual queue backlog $X_m(t)$ for every primary node m, the actual queue backlog $L_n(t)$ for every secondary node n, the achievable transmission rates, and choose (m, n, α) solving the following optimization problem:

$$Q_{1c} = \underset{(m,n,\alpha)\in\mathcal{F}}{\operatorname{argmax}} \left\{ \left(1 + \frac{2X_m(t)}{K}\right) U_{mn}(t) + \hat{V}_{mn}(t) \right\} \tag{3.35}$$

Then, update the virtual and actual queues according to the queue dynamics in (3.14), and (3.33), respectively.

Note that the only difference between Q_{1c} and Q_{1b} is that $V_{mn}(t)$ in Q_{1b} is replaced by $\hat{V}_{mn}(t)$ in Q_{1c}. Hence, Q_{1c} is the optimal scheduling algorithm. However, unlike Q_{1b}, Q_{1c} is a non-convex optimization problem for a given (m, n) pair due to the definition of $\hat{V}_{mn}(t)$ given in (3.34). The problem can be transformed into a convex program using auxiliary variables as shown in proof of Proposition 3.1.

Proposition 3.1. *For a given primary-secondary node pair* (m, n)*, let* α *be the solution of* (3.19)*, and* β *be defined as in* (3.4)*. If* $\alpha(1-\beta) > \frac{L_n(t)}{R_n(t)}$*, then the optimal* α^* *for* (3.35) *is given as*

$$\alpha^* = \frac{L_n(t) + R^r_{mn}(t)}{R^r_{mn}(t) + R^s_n(t)},$$

and if $\alpha(1-\beta) \leq \frac{L_n(t)}{R_n(t)}$*, then the optimal solution for* (3.35) *is* $\alpha^* = \alpha$*.*

Proof. By using a similar idea as in [43], we introduce an auxiliary variable π_{mn} to obtain a convex problem. For a given (m, n) pair, the solution of the following problem gives the optimal α:

$$\max_{\alpha,\pi_{mn}} \left\{ \tilde{\lambda}_m(t) h_1\left((1-\alpha)R^r_{mn}(t)\right) + h_2\left(\pi_{mn}\right) \right\} \tag{3.36}$$

$$\text{s.t.} \quad R^p_m(t) \leq (1-\alpha)R^r_{mn}(t), \tag{3.37}$$

$$(1-\alpha)R^r_{mn}(t) \leq \alpha R^s_n(t), \tag{3.38}$$

$$L_n(t) \geq \pi_{mn}, \tag{3.39}$$

$$\alpha(R^r_{mn}(t) + R^s_n(t)) - R^r_{mn}(t) \geq \pi_{mn}, \tag{3.40}$$

Note that the objective function is modified and is expressed with respect to the auxiliary variable π_{mn} and α. It can be shown that (3.36)-(3.40) is jointly concave in α

and π_{mn} (i.e., Hessian matrix of the objective function is negative semidefinite since all eigenvalues of it are negative). Hence, at a given time slot first-order optimality conditions given by KKT equations are sufficient for global optimality of (3.36)-(3.40).

In the following, we drop the parameter t. Let μ_3 and μ_4 be the dual variables associated to the constraints in (3.39) and (3.40), respectively. Let μ_3^*, μ_4^*, α^* and π_{mn}^* denote the values taken by μ_3, μ_4, α and π_{mn} at the optimal solution respectively. From KKT conditions, the following equalities hold.

$$\frac{dh_2}{d\pi_{mn}}|_{\pi_{mn}=\pi_{mn}^*} - \mu_3^* - \mu_4^* = 0. \tag{3.41}$$

$$\mu_3^*(L_n - \pi_{mn}^*) = 0. \tag{3.42}$$

$$\mu_4^*\left(\alpha^* - \frac{\pi_{mn}^* + R_{mn}^r}{R_{mn}^r + R_n^s}\right) = 0. \tag{3.43}$$

It is clear that the objective function in (3.36) decreases with α. If $\pi_{mn}^* = L_n$, at the optimal solution α takes its minimum value which is given by (3.40). Thus, α^* is given as follows,

$$\alpha^* = \frac{L_n + R_{mn}^r}{R_{mn}^r + R_n^s}.$$

If $\pi_{mn}^* \neq L_n$, (3.41) and (3.42) yield the following equalities,

$$\mu_3^* = 0.$$
$$\mu_4^* = \frac{dh_2}{d\pi_{mn}}|_{\pi_{mn}=\pi_{mn}^*} > 0$$

From (3.43), α^* is given by

$$\alpha^* = \frac{\pi_{mn}^* + R_{mn}^r}{R_{mn}^r + R_n^s}.$$

Thus,

$$\pi_{mn}^* = \alpha^*(R_{mn}^r + R_n^s) - R_{mn}^r. \tag{3.44}$$

Clearly, (3.44) states that α^* is obtained following the same steps in infinite backlog case. This completes the proof. □

3.4 Secondary Constraints and Long Term Rewards

3.4.1 Formulation and Optimal Algorithm

In this section, we study a generalized version of the problem studied in Section 3.3. Here, a long term constraint is imposed on the minimum performance of each secondary node. More specifically, a portion of the primary utility achieved by cooperation is guaranteed for each cooperating secondary node, in an average sense. In fact, the formulation of the problem below allows for the idea of *banking* between primary and secondary nodes. That is, in contrast to the immediate rewards of Section 3.3, here, secondary nodes are guaranteed a specific share over a large number of time slots. This is achieved by allowing α to take values such that $\alpha \leq 1$, i.e., we lift the constraint imposed in the first inequality of (3.10). The problem is formulated as follows:

$$\begin{aligned}
&\max_{Q \in \mathcal{F}} \ \bar{W}(Q) \\
\text{s.t.:}1)\ &\mathbb{E}\left[U_m^t(Q,t)\right] \geq C_m,\ m \in \{1,2,\ldots,M\} \\
2)\ &\mathbb{E}\left[V_n^t(Q,t)\right] \geq \mathbb{E}\left[\sum_{m=1}^{M} \phi(U_{mn}(Q,t))\right], \\
&n \in \{1,2,\ldots,N\},
\end{aligned} \tag{3.45}$$

where $\phi(\cdot)$ is a non-negative, non-decreasing scalar-valued function. We assume that the constraints in (3.45) are within the feasibility region. Define $\nu_n(Q,t) \triangleq \sum_{m=1}^{M} \phi(U_{mn}(Q,t))$. For each of the constraints above, we construct virtual queues such that the queue dynamics are given by

$$X_m(t+1) = [X_m(t) - U_m^t(Q,t)]^+ + C_m, \tag{3.46}$$

$$Y_n(t+1) = [Y_n(t) - V_n^t(Q,t)]^+ + \nu_n(Q,t), \tag{3.47}$$

$m \in \{1, 2, \ldots, M\}$, $n \in \{1, 2, \ldots, N\}$. We assume that $\nu_n(Q,t) \leq \nu^{max}$ for all $n \in \{1, 2, \ldots, N\}$, $t \geq 0$ and for all $Q \in \mathcal{F}$. Let $\mathbf{Y}(t) = (Y_1(t)Y_2(t) \cdots Y_N(t))$ be the vector of virtual queues of secondary nodes. Define the following quadratic Lyapunov function and conditional Lyapunov drift

$$L_2(\mathbf{X}(t), \mathbf{Y}(t)) \triangleq \sum_{m=1}^{M} X_m^2(t) + \sum_{n=1}^{N} Y_n^2(t), \tag{3.48}$$

$$\begin{aligned}&\Delta_2(\mathbf{X}(t), \mathbf{Y}(t)) \triangleq \\ &\mathbb{E}\left[L_2(\mathbf{X}(t+1), \mathbf{Y}(t+1)) - L_2(\mathbf{X}(t), \mathbf{Y}(t)) | \mathbf{X}(t), \mathbf{Y}(t)\right].\end{aligned} \tag{3.49}$$

We also define the following conditional expectation.

$$\bar{\nu}_n(Q,t) \triangleq \mathbb{E}[\nu_n(Q,t) | \mathbf{X}(t), \mathbf{Y}(t)]. \tag{3.50}$$

The Lyapunov drift in (3.49) is bounded by the following Lemma where the proof is very similar to the proof of Lemma 3.1 and is omitted for brevity.

Lemma 3.3. *For every time slot t, the Lyapunov drift defined in* (3.49) *can be upper bounded as follows.*

$$\begin{aligned}&\Delta_2(\mathbf{X}(t), \mathbf{Y}(t)) - K\mathbb{E}[W(Q,t) | \mathbf{X}(t), \mathbf{Y}(t)] \\ &\quad \leq B_2 + 2\sum_{m=1}^{M} X_m(t) C_m - \sum_{m=1}^{M} (K + 2X_m(t)) \bar{U}_m^t(Q,t) \\ &\quad - \sum_{n=1}^{N} (K + 2Y_n(t)) \bar{V}_n^t(Q,t) + \sum_{n=1}^{N} 2Y_n(t) \bar{\nu}_n(Q,t),\end{aligned}$$

where

$$B_2 = \frac{1}{2}\left(\sum_{m=1}^{M} C_m^2 + M(U^{max})^2 + N((\nu^{max})^2 + (V^{max})^2)\right)$$

and K is a system parameter that characterizes a tradeoff between performance optimization and unfinished work in the virtual queues.

Now we present our opportunistic scheduling algorithm.

Scheduling Algorithm Q_2: At each time slot t, the scheduler observes the virtual queue states $X_m(t)$, $Y_n(t)$ and the achievable rates $R^p_m(t)$, $R^r_{mn}(t)$ and $R^s_n(t)$ for all $m \in \{1, 2, \ldots, M\}$ and $n \in \{1, 2, \ldots, N\}$, and then solves the following optimization problem:

$$Q_2 = \underset{(m,n,\alpha)\in\mathcal{F}}{\operatorname{argmax}} \left\{ \left(1 + \frac{2X_m(t)}{K}\right) U_{mn}(t) + \left(1 + \frac{2Y_n(t)}{K}\right) V_{mn}(t) - \left(\frac{2Y_n(t)}{K}\right) \phi(U_{mn}(t)) \right\}$$

The virtual queues are then updated according to the queue dynamics in (3.46), (3.47).

The structure of the scheduling policy suggests that when a secondary virtual queue $Y_n(t)$ is congested, then the system has a *debt* to pay to SU n. This is accomplished by favoring instantaneous allocations that reduce this debt by increasing payments (i.e., higher weight for $V_{mn}(t)$) and reduced additional debt (i.e., lower weight for $\phi(U_{mn}(t))$). Therefore, it is possible that the system allocates an entire time slot to a SU without requiring the relay of a PU's data. Similarly, it is also possible that a SU relays primary data without obtaining immediate share of that time slot to transmit its own data.

3.4.2 Algorithm Analysis

The analysis follows the same strategy as in Section 3.3.3. Let $\bar{W}_2^*$ be the optimal time average system utility achieved over all scheduling policies for the problem (3.45), and consider the class of stationary randomized scheduling algorithms that are independent of the queue states.

Lemma 3.4. *If vectors* $\mathbf{C}$ *and* $\mathbb{E}[\boldsymbol{\nu}] = \mathbb{E}[(\nu_1\ \nu_2 \cdots\ \nu_N)]$ *are feasible, then there exists a stationary randomized policy* Q_{s_2} *that solves* (3.45) *and satisfies the following:*

$$\mathbb{E}[W(Q_{s_2}, t)] = \bar{W}_2^*, \tag{3.51}$$

$$\mathbb{E}\left[U^t_m(Q_{s_2}, t)\right] \geq C_m,\ m \in \{1, 2, \ldots, M\}, \tag{3.52}$$

$$\mathbb{E}\left[V^t_n(Q_{s_2}, t)\right] \geq \mathbb{E}\left[\nu_n(Q_{s_2}, t)\right], \tag{3.53}$$

where $\bar{W}_2^*$ *is the optimal performance for the problem* (3.45) *over all scheduling policies. Moreover, if* $\mathbf{C}$ *and* $\mathbb{E}[\boldsymbol{\nu}]$ *are strictly interior to the feasibility region, then there exists* $\epsilon' > 0$ *and a stationary scheduling policy* $Q_{s_2(\epsilon')}$ *satisfying:*

$$\mathbb{E}\left[U_m^t(Q_{s_2(\epsilon')}, t)\right] \geq C_m + \epsilon', \; m \in \{1, 2, \dots, M\}, \tag{3.54}$$

$$\mathbb{E}\left[V_n^t(Q_{s_2(\epsilon')}, t)\right] \geq \mathbb{E}\left[\nu_n(Q_{s_2}, t)\right] + \epsilon', \tag{3.55}$$

with an optimal total average utility $\bar{W}_2^*(\epsilon')$ *such that* $\bar{W}_2^*(\epsilon') \leq \bar{W}_2^*$ *where* $\bar{W}_2^*(\epsilon') \to \bar{W}_2^*$ *as* $\epsilon' \to 0$.

Theorem 3.4. *If the constraints in* (3.45) *are feasible, then the proposed algorithm* Q_2 *stabilizes the virtual queues and achieves the following bounds.*

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[W(Q_2, \tau)] \geq \bar{W}_2^* - \frac{B_2}{K},$$
$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[\sum_{m=1}^{M} X_m^2(\tau) + \sum_{n=1}^{N} Y_n^2(\tau)\right] \leq \frac{B_2 + KW^{max}}{\epsilon'_{max}},$$

where $\bar{W}_2^*$ *is the optimal value for the time average expected utility,* $B_2 = \sum_{m=1}^{M} C_m^2 + M(U^{max})^2 + N((\nu^{max})^2 + (V^{max})^2)$, $W^{max} = U^{max} + V^{max}$ *and* ϵ'_{max} *is the largest* ϵ' *such that constraints* (3.54) *and* (3.55) *are feasible.*

Proof. The proof uses Lemma 3.3 and Lemma 3.4 and is similar to the proof of Theorem 3.3. □

3.5 Maximization of Net Benefit

Transmission power control is an essential part of wireless communications, and it is especially important for wireless devices with limited energy resources. In this section, our objective is to investigate scheduling and power control policies when PUs and SUs are allowed to cooperate. For this purpose, we define "net benefit" of a user as the difference between the total average utility and the total weighted average energy consumption. Our objective is to determine optimal dynamic joint power control and scheduling policy that maximizes aggregate net benefits of primary and secondary systems considering immediate rewards.

Let $E^p_{mn}(Q,t)$ and $E^s_{mn}(Q,t)$ be the consumed energy by PU m and SU n under policy Q at time slot t, respectively. Also we define the net benefit of primary and secondary users, $F_{mn}(Q,t)$ and $G_{mn}(Q,t)$, as follows [64]:

$$F_{mn}(Q,t) = U_{mn}(Q,t) - \rho \cdot E^p_{mn}(Q,t), \tag{3.56}$$

$$G_{mn}(Q,t) = V_{mn}(Q,t) - \rho \cdot E^s_{mn}(Q,t), \tag{3.57}$$

where ρ is the cost per unit transmission energy. Let the total net benefit of both systems when policy Q is employed at a given time slot t, be $H(Q,t)$. Then,

$$H(Q,t) = \sum_{m=1}^{M}\sum_{n=0}^{N} F_{mn}(Q,t) + G_{mn}(Q,t). \tag{3.58}$$

Define

$$\begin{aligned} F^t_m(Q,t) &\triangleq \sum_{n=0}^{N} F_{mn}(Q,t) \\ , G^t_n(Q,t) &\triangleq \sum_{m=1}^{M} G_{mn}(Q,t) \\ , E^p_m(Q,t) &\triangleq \sum_{n=0}^{N} E^p_{mn}(Q,t). \end{aligned}$$

These values are upper-bounded such that $F^t_m(Q,t) \leq F^{max}$, $G^t_n(Q,t) \leq G^{max}$ and $E^p_m(Q,t) \leq E^{max}$.

The total expected net benefit for time slot t is defined as $\bar{H}(Q,t) \triangleq \mathbb{E}[H(Q,t)]$. Now, we consider the following optimization problem:

$$\begin{aligned} &\max_{Q\in\mathcal{G}} \ \bar{H}(Q,t) \\ \text{s.t.1) } &\mathbb{E}\left[U^t_m(Q,t)\right] \geq C_m, m \in \{1,2,\cdots,M\} \\ \text{2) } &\mathbb{E}\left[G^t_n(Q,t)\right] \geq 0,\ n \in \{1,2,\ldots,N\}, \end{aligned} \tag{3.59}$$

where C_m is the minimum performance constraint for each PU m, and $\mathcal{G}$ is the set of feasible policies to be defined later. If SU n relays the data of PU m then the net benefit of SU may be negative. In this case, SU n may not be willing to join cooperation. Hence, we impose the non-negativity constraint on the expected net benefit of secondary system in (3.59).

Here, the scheduling policy Q is a rule that selects $(m, n, \alpha, \beta, P^p_{mn}, P^s_{mn})$ at time slot t where P^p_{mn} and P^s_{mn} are the transmission powers from PU m to SU n and from SU n to destination to relay data of PU m, respectively. Under this model, if cooperation is infeasible, PU m transmits directly to the destination at power of P^p_{m0}. In addition, SU n transmits its own data to the destination using the power level P^s_{0n}. We use the same notational convention with the data rates R.

Even though our results are general for all channel state distributions, in numerical evaluations, we assume all channels to be *Gaussian*. We represent the uplink channel for PU m, cross channel between PU m and SU n ($m \in \{1, 2, \cdots, M\},\ n \in \{1, 2, \cdots, N\}$) with a power gain (magnitude square of the channel gains) $l_{m0}(t)$ and $l_{mn}(t)$, respectively, at time slot t. We normalize the power gains such that the (additive Gaussian) noise has unit variance. In the following, we employ information theoretic expressions for the achievable data rates. Thus,

$$R^p_{mn}(t) = \log(1 + P^p_{mn}(t) l_{mn}(t))$$
$$R^s_{mn}(t) = \log(1 + P^s_{mn}(t) l_n(t)).$$

In addition, peak power constraints are imposed in the network such that,

$$0 \le P^p_{mn}, \le P^m_{max}, \tag{3.60}$$
$$0 \le P^s_{mn}, P^s_n \le P^n_{max}, \tag{3.61}$$

$\forall m \in \{0, 1, 2, \cdots, M\}$ and $n \in \{0, 1, 2, \cdots, N\}$. Given these definitions, the optimization problem in (3.59) along with constraints (3.3), (3.60) and (3.61) is a non-convex optimization problem. Hence, it is hard to find a closed form solution for this problem.

Here, we consider a simplified version of the problem, where the values of α and β are taken as constants satisfying the following,

$$(1-\alpha) = \alpha\beta = \alpha(1-\beta) = \frac{1}{3}. \tag{3.62}$$

We fix the duration of each of the three phases in our cooperative scheme to one third of the time slot as implied by (3.62). Clearly, this is a suboptimal solution, since we do not consider α and β as decision variables. Nevertheless, this model has practical applicability. In many real networks, including cellular networks [69], users are assigned fixed time slots to complete their transmissions. In such systems, transmission power control is the main tool to adjust transmission rates.

Since transmission durations are equal, the following holds for all $m \in \{1, 2, \cdots, M\}$ and $n \in \{1, 2, \cdots, N\}$,

$$R^p_{mn}(t) = R^s_{mn}(t). \tag{3.63}$$

Thus, given instantaneous channel conditions and for a given value of P^p_{mn}, we can determine P^s_{mn}. Hence, we use the notation $Q = (m, n, P^p_{mn}, P^s_{0n})$ to denote the joint scheduling and power control decision of policy Q.

Let $\mathcal{G}$ be the set of tuples $(m, n, P^p_{mn}, P^s_{0n})$ satisfying (3.60), (3.61), (3.62), and the following condition:

$$R^p_{m0}(t) \leq \frac{1}{3} R^p_{mn}(t) \tag{3.64}$$

where $\forall m \in \{1, 2, \cdots, M\}, n \in \{1, 2, \cdots, N\}$. Clearly, the above inequality states that SU n is allowed to join cooperation if it improves the instantaneous utility of PU m. Moreover, set $\mathcal{G}$ is constructed from all $(m, n, P^p_{mn}, P^s_{0n})$. $E^p_{mn}(Q, t)$ and $E^s_{mn}(Q, t)$ under scheduling policy Q are given as follows,

$$E^p_{mn}(Q,t) = \begin{cases} P^p_{m0} & ; \text{if } n = 0 \\ \frac{1}{3} P^p_{mn} & ; \text{otherwise} \end{cases} \tag{3.65}$$

$$E^s_{mn}(Q,t) = \begin{cases} 0 & ; \text{if } n = 0 \\ \frac{1}{3} P^s_{mn} + \frac{1}{3} P^s_{0n} & ; \text{otherwise} \end{cases} \tag{3.66}$$

We solve problem (3.59) using the stochastic network optimization tool of [8]. For the first set of constraints in (3.59) we construct the system of virtual queues defined in (3.67). Also, for each of the second set of constraints in (3.59), we construct virtual queues $S_n(t)$ with queue dynamics given as follows,

$$X_m(t+1) = [X_m(t) - U_m(Q,t)]^+ + C_m, \tag{3.67}$$

$$S_n(t+1) = [S_n(t) - G^t_n(Q,t)]^+, \tag{3.68}$$

$n \in \{1, 2, \ldots, N\}$. Now we present our joint power allocation and scheduling algorithm.

Scheduling Algorithm Q_3 **:** At each time slot t, observe the virtual queue states $X_m(t)$ and $S_n(t)$ and select $(m, n, P^p_{mn}, P^s_{0n})$ solving the following optimization problem:

$$Q_3 = \underset{(m,n,P^p_{mn},P^s_{0n}) \in \mathcal{G}}{\text{argmax}} \left\{ \left(1 + \frac{X_m(t)}{K}\right) U_{mn}(t) - \rho \cdot E^p_{mn}(t) + \left(1 + \frac{S_n(t)}{K}\right) G_{mn}(t) \right\}$$

Theorem 3.5. *If the constraints in* (3.59) *are feasible, then the proposed algorithm* Q_3 *stabilizes the virtual queues and achieves the following bounds:*

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[H(Q,\tau)] \geq \bar{H}^* - \frac{B_3}{K},$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[\sum_{m=1}^{M} X^2_m(\tau) + \sum_{n=1}^{N} S^2_n(\tau)\right] \leq \frac{B_3 + K H^{max}}{\varepsilon_{max}},$$

where $\bar{H}^*$ *is the solution of* (3.59) *among class of policies* $\mathcal{G}$*,* H_{max} *is the maximum net* *benefit that can be achieved at any time slot,* $B_3 = \frac{1}{2}\left(\sum_{m=1}^{M} C_m^2 + M(U^{max})^2 + N(G^{max})^2\right)$ *and* ε_{max} *can be defined similarly as* ϵ_{max} *in Section 3.3.3.*

Proof. The proof is similar to the proof of Theorem 3.3, and involves finding an upper bound on the conditional Lyapunov drift for a quadratic Lyapunov function of virtual queues $X_m(t)$ and $S_n(t)$, and uses Lemma 3.4. $\square$

It can be seen that when there is no cost for power consumption, i.e., $\rho = 0$, the scheduling algorithm Q_3 has the same form as the scheduling algorithm Q_{1b}. In this sense, Q_{1b} can be considered as a specific case of Q_3.

3.6 Numerical Results

In this section, we simulate a wireless network with $M = 4$ primary nodes and a varying number of secondary nodes all communicating with a common destination. First, we present a comparison between our cooperative scheduling scheme and the optimal non-cooperative scheme. Channel states vary randomly between 'Good' and 'Bad' for primary and secondary users. Transmission rates corresponding to the channel states: {'Good', 'Bad'} are set to {100, 15} units/slot and channel states evolve independently across users and across time. For all pairs (m, n), the transmission rates are given by $R_m^p(t) = \{100,\ 10\}$ with probability $\{0.5,\ 0.5\}$, $R_{mn}^r(t) = \{100,\ 10\}$ with probability $\{0.6,\ 0.4\}$, and $R_n^s(t) = \{100,\ 10\}$ with probability $\{0.6,\ 0.4\}$. Given these channel statistics, we run the simulation for $200,000$ time slots which is sufficient for the convergence of algorithm Q_{1a} for the above channel statistics. For the utility functions, we employ the functions $h_1(x) = h_2(x) = \log(1 + x)$.

For the constraints on the primary users performance in the non-cooperative system, we adopt a fair sharing policy, that is, the achievable primary system utility is to be divided evenly among the primary users. We set the constraints C_m in (3.11) as in the example given is Section 3.3.2 for the sake of comparison with non-cooperative systems. Applying scheduling policy Q_{1a}, we let $\Gamma = 1.01$ and use a stochastic approximation approach to estimate the parameters $\lambda_m^*, m \in \{1, 2, \ldots, M\}$ as follows. First, from the constraints on the primary user performance, we see that for $m \in \{1, 2, \cdots, M\}$, λ_m^* is the root to the following equation.

$$f_m(\lambda_m) = (\lambda_m - 1)\left(\mathbb{E}\left[\sum_{n=0}^{N} U_{mn}(Q_{1a}, t)\right] - C_m\right)$$

But since we only have knowledge about the instantaneous channel gains, we need to estimate the distribution of the utility functions. Hence, using the observation we have, we can write an estimate g_m^k of f_m^k as:

$$g_m^k(\lambda_m) = (\lambda_m^k - 1)\left(\sum_{n=0}^{N} U_{mn}(Q_{1a}, t) - C_m\right)$$

where k is the iteration index. Since this estimator is unbiased ($\mathbb{E}[g_m^k - f_m^k(\lambda_m^k)] = 0, \forall m$), then, we can use a stochastic approximation algorithm of the form

$$\lambda_m^{k+1} = \lambda_m^k - \delta^k g_m^k$$

where δ^k can be taken to be $1/k$ [9]. For a given time slot t, it can be shown that (3.12) is a concave function in α. The optimization is then done over all pairs so that α satisfies the condition (3.10), then the tuple $(m, n, \alpha) \in \mathcal{F}_1$ that maximizes (3.12) is selected by the scheduler at this time slot. For a pair (m, n), since the objective function is concave and the constraint is linear in α. Then, Karush-Kuhn-Tucker conditions are both necessary and sufficient to solve the problem (3.12), along with (3.10) [70]. In Figure 3.3, the average system utility is plotted with respect to the number of cognitive users. The cooperative scheme achieves higher average system utility compared to the non-cooperative scheme. For $N = 1$, the constraints are infeasible to achieve, however the Q_{1a} still performs better than the non-cooperative policy. For $N \geq 1$, the constraints are feasible. Moreover, exploiting the opportunity relaying offers, we could achieve non zero secondary system average utility. Figure 3.3 also shows the per user (primary) performance. It can be seen that the smallest per user performance is still better than the non-cooperative case with at least the value Γ. Scheduling policy Q_{1b} yields the

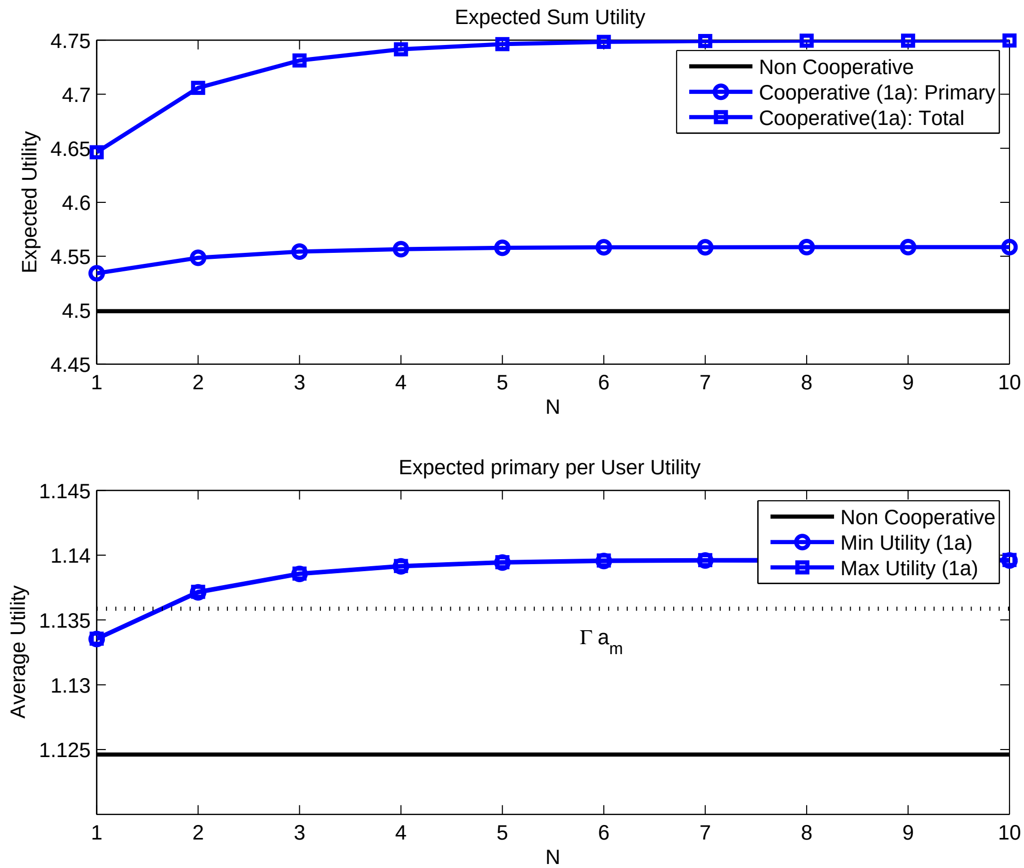

Figure 3.3: Average total system utility.

same results as Q_{1a} under the same channel parameter selection, using $K = 5,000$.

In the second experiment, we evaluate the performance of our scheduling policy when SUs have finite backlogs. Figure 3.4 depicts the average utility per PU versus average aggregate arrival rate when SUs have finite backlogs. In this experiment, we apply scheduling policy Q_{1c} for the same channel statistics used in the first simulation and set $M = 4$, $N = 3$. It is assumed that at $t = 0$, all queues of SUs are empty, i.e., $L_n(0) = 0 \quad \forall n$, and new data arrives at secondary users with the same arrival rate according to independent Bernoulli processes. If all SUs have no data to transmit at a time slot t, i.e., $L_n(t) + A_n(t) = 0$ $\forall n$, they do not join cooperation. In this case, PUs transmit to the base station directly. Hence, the primary utility achieved is the same as in the non-cooperative case. As the arrival rate increases, SUs start to be backlogged, and join the network. Therefore, the utility of primary users increases as well. When

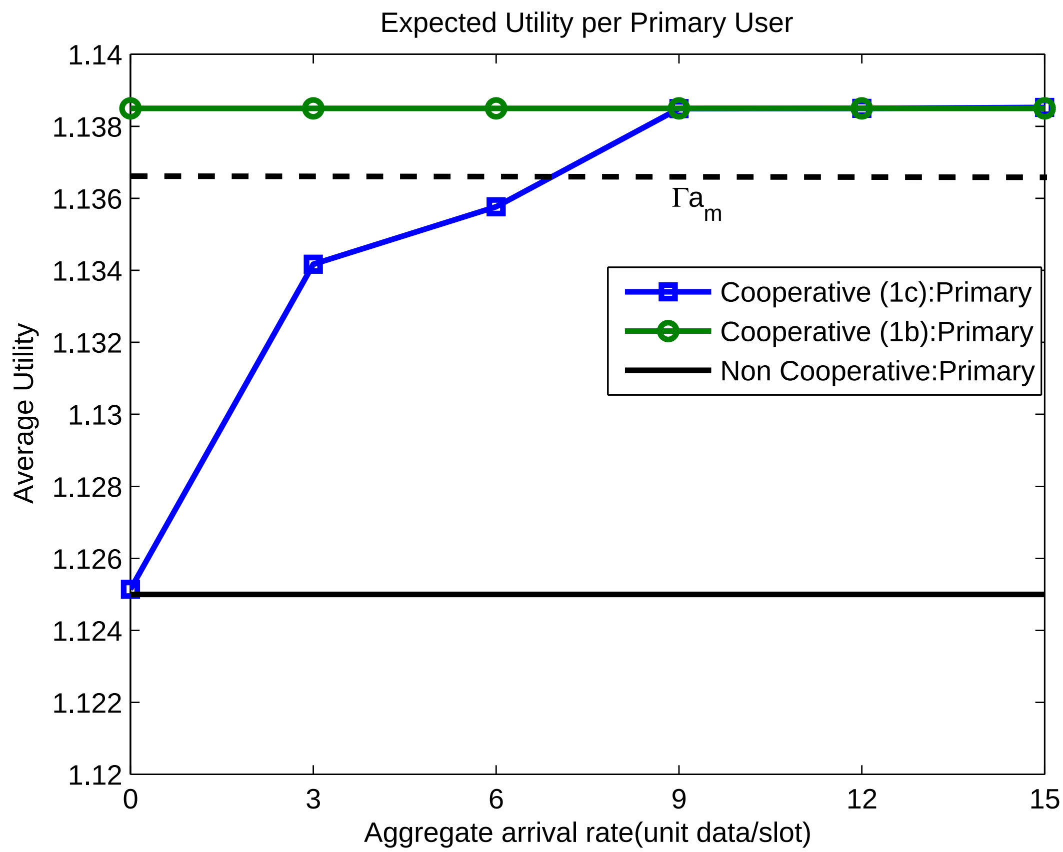

Figure 3.4: Average per primary user utility.

secondary users have sufficiently large backlog, primary users can achieve the utility obtained in infinite backlog case at most.

Next, in the third experiment, we simulate the long term rewards scenario and apply scheduling policy Q_2 when $M = 4$ and the number of SUs is fixed at $N = 5$ for the same channel statistics used in the first simulation. In Figure 3.5, the running average of the expected utility of SU 3 up to time t is plotted and compared to the average primary utility achieved through cooperation with SU 3 for $\phi(U_{mn}(Q,t)) = bU_{mn}(Q,t)$, and $b = 1.2$. In other words, for every packet of primary traffic SU n relays, SU n is rewarded by being scheduled to send 1.2 of its own packets on the long term. In this experiment, we set $C_m = 3.5 \; \forall m$. Stability is achieved for all primary and secondary virtual queues.

We next evaluate the performance of the scheduling algorithm Q_3 in terms of net benefit, average utility and energy consumption. We assume a scenario where $M = 4$,

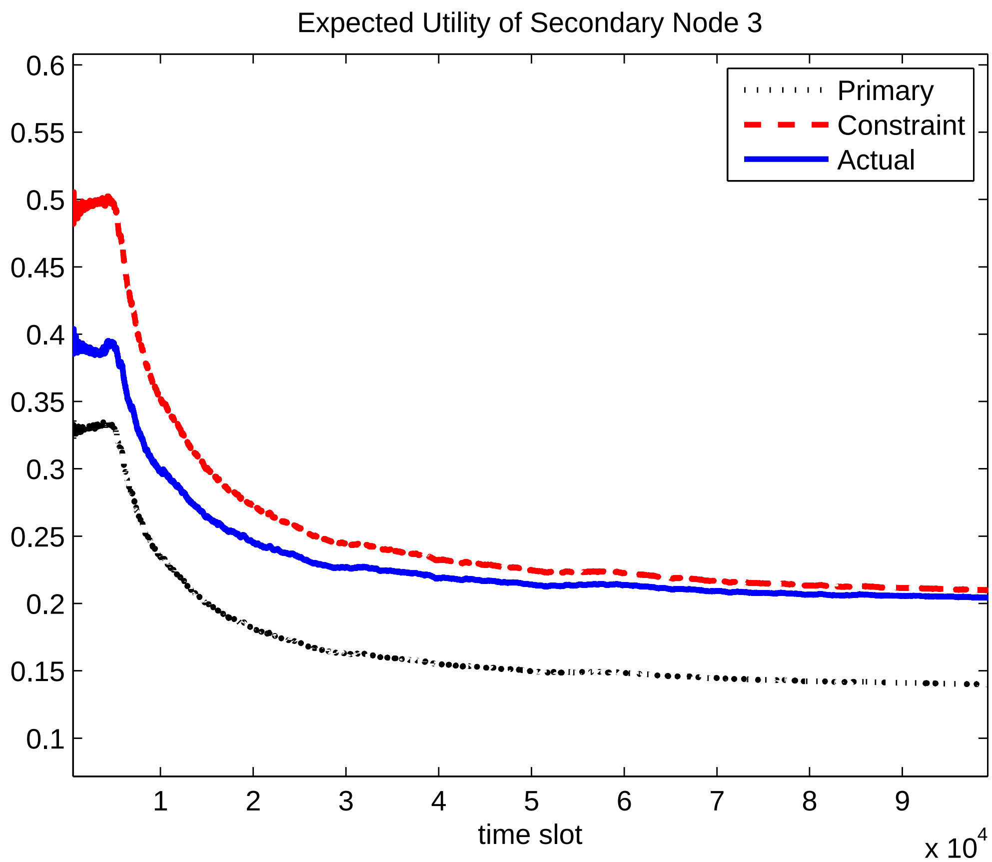

Figure 3.5: Performance of secondary node 3.

$\rho = 1$ and for varying number of N. The utility achieved by primary users in the non-cooperative system where the objective is to maximize the net benefit and $\rho = 1$ are taken as the minimum requirements in this experiment. We set $C_m = 1.095\ \forall m$ ($\Gamma = 1$). We assume that all users have infinite backlog, and $P^m_{max} = P^n_{max} = 10$. In addition, the channel gains are adjusted so that the transmission rates in the first simulation with the maximum transmission power can be obtained. In Figure 3.6, the average system net benefit with respect to the number of SUs is plotted. Clearly, our cooperative scheme performs better than non cooperative scheme in terms of average system net benefit. Note that all constraints in (3.59) are satisfied in this experiment.

In Figure 3.7, the average utility and energy consumption of primary and secondary users are plotted for increasing values of ρ. We set $M = 4$, $N = 5$ and $C_m = 1\ \forall m$. If users do not pay a penalty cost for transmission power (i.e., $\rho = 0$) they transmit at the maximum transmission power, thus the total average energy con-

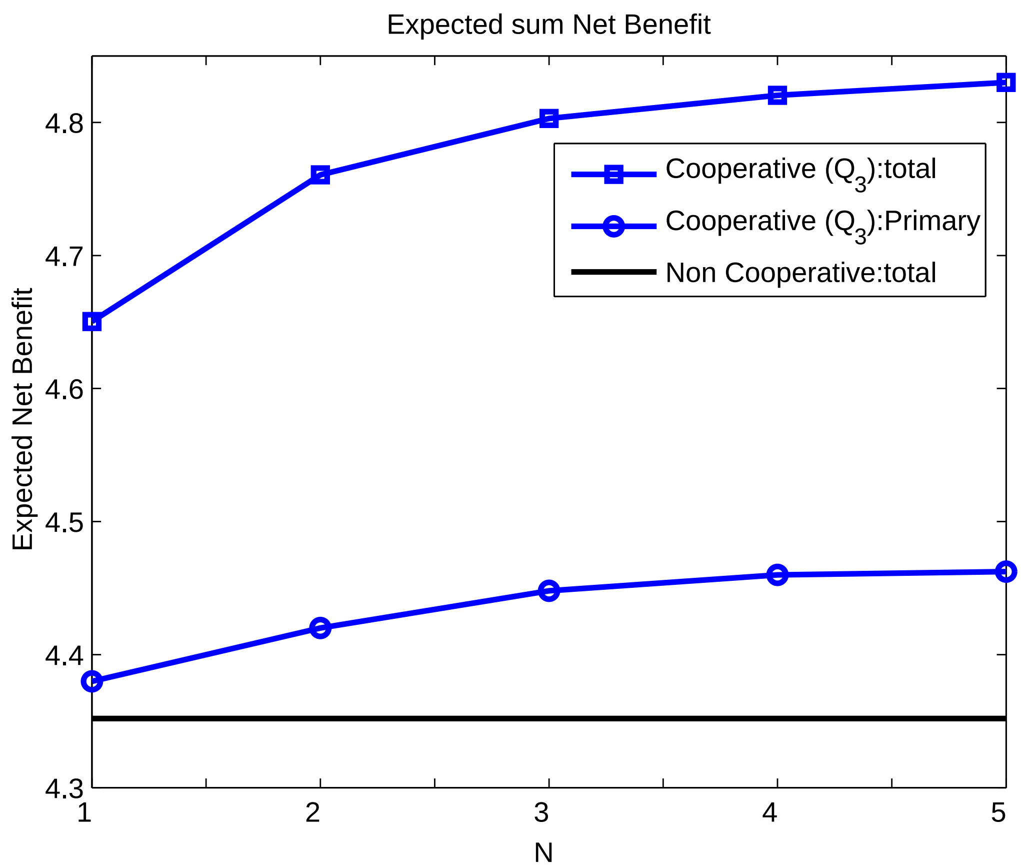

Figure 3.6: Average net benefit.

sumption is maximized. It is seen that the average utility achieved per a primary user, which is equal to 1, is same as the minimum requirement of the corresponding PU. However, energy consumption of primary system reduces by allowing SUs to join cooperation. That is, PUs conserve energy by cooperating with SUs and applying power control algorithm Q_3. Thus, primary net benefit increases as shown in Figure 3.6. As ρ increases, SUs become unwilling to join cooperation, and prefer not to transmit in order to satisfy their net benefit requirements. Consequently, their average utility and energy consumption decrease. When SUs leave cooperation, PUs transmit directly to destination, and only the minimum primary requirements are achieved since the energy expenditure is costly. The average energy consumption is fixed at approximately 0.025 in this case.

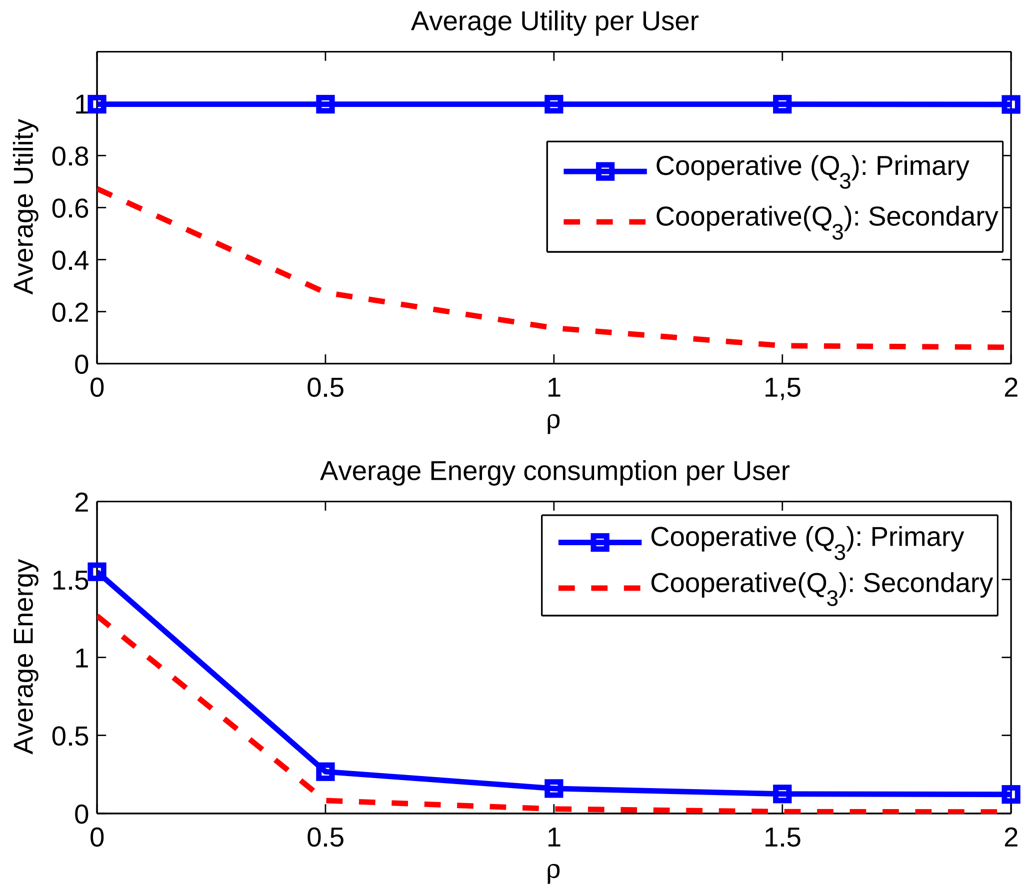

Figure 3.7: Average utility and energy consumption.

3.7 Chapter Summary

We have developed optimal scheduling policies by exploiting the time-varying channel conditions and realizing the benefits of cooperative transmission in a cognitive radio network. The proposed model is based on the idea that secondary users can have opportunity to transmit their own data if they can improve the performance of a primary user via cooperation. First, we have studied the immediate reward strategy considering the cases in which secondary users have infinite and finite backlogs. Then, long term rewards is studied where we introduce the idea of banking between primary and secondary users. In this banking model, secondary users are guaranteed a portion of the primary utility on a long term basis, instead of immediate utility. Finally, we have investigated the energy-utility trade off by considering the power control and scheduling problems jointly where the objective is to maximize the net benefit. Numerical results show that our cooperative schemes improve the basic primary network performance in addition to giving SUs the ability to communicate in return for their cooperation. In addition, energy consumption of primary system can be reduced by allowing secondary users to join the network.

Chapter 4

Multi-user Scheduling via Hierarchical Modulation

In Chapter 3, we have investigated the performance of cognitive radio network which is one of the most important emerging technologies. In this Chapter, we investigate multi-user transmission that is considered as another key technology in future wireless systems. We study the network stability problem when more than one users are allowed to be scheduled simultaneously. The idea is to encode messages destined to multiple users experiencing different channel conditions by employing hierarchical modulation. For two-user scheduling problem, we develop a throughput-optimal algorithm which can stabilize the network whenever this is possible (i.e., traffic load is within the achievable rate region.). In addition, we analytically prove that the proposed algorithm achieves a larger achievable rate region compared to the conventional Max-Weight algorithm which employs uniform modulation and transmits to a single user. We demonstrate the efficacy of the algorithm on a realistic simulation environment using the parameters of High Data Rate protocol in a Code Division Multiple Access system [4]. Simulation results show that with the proposed algorithm, the network can carry higher traffic load with lower delays.

4.1 Overview

As we mentioned in Chapter 2 Max-Weight algorithm is *throughput-optimal*, i.e., it stabilizes the network for all arrival rate vectors that are *strictly* within the achievable rate region. The performance of Max-Weight algorithm has been investigated in depth for the single user scheduling case. However, determining a throughput-optimal algorithm when more than one users are scheduled simultaneously has not received much attention. Moreover, as stated in [37], the main reason for the poor delay performance of Max-Weight algorithm is that only one queue is served at a time. In this Chapter, we investigate multi-packet transmission technologies within Max-Weight algorithm, which enables us to schedule more than one users at a scheduling time. In particular, we propose a modulation-assisted multi-packet transmission technique that employs hierarchical modulation (HM) that is also termed superposition coding or embedded constellations. HM enables a base station (or a transmitter) to transmit information to two or more users simultaneously rather than a single user in a given time slot. Therefore, it is possible to serve more than one queue at a time.

The authors in [35] are the first to show the advantage of HM in broadcast systems. In [36], the authors proposed a multi-user scheduling algorithm and showed that HM offers lower queueing delay at the transmission buffer. However, in neither of these works, the authors considered the stability of the network. Network utility maximization problem with HM was investigated in [71]. Meanwhile, [41] proposed scheduling and flow control algorithms but did not take into account the effect of modulation on the performance of the algorithm.

In this Chapter, our contribution can summarized as follows: *i*) we propose a throughput optimal algorithm, namely Max-Weight with Hierarchical Modulation (MWHM) when two users are scheduled simultaneously; *ii*) we give the conditions under HM that should be employed by considering both analytical and implementation issues. *iii*) we prove that the proposed algorithm achieves larger rate region compared to the conventional Max-Weight algorithm; *iv*) we develop a lower complexity version of MWHM algorithm; *v*) we demonstrate via realistic simulations that our algorithm not only keeps the network stable with higher arrival rate but also reduces the average delay.

4.2 System Model

We consider a cellular system with a single base station (BS) transmitting to N users. Let $\mathcal{N}$ denote the set of users in the cell. Time is slotted, $t \in \{0, 1, 2, \ldots\}$. Let T_s denote the length of the time slot in seconds. Let $h_n(t)$ represent channel gain of user n at time t, $n \in \{1, 2, \ldots, N\}$. The gain of the channel is constant over the duration of a time slot but varies between slots.

HM is one of the techniques for multiplexing and modulating multiple data streams into one single symbol stream, where those multiple symbols are superimposed together before transmission. In this work, for the sake of ease of exposition we assume that only two layers of hierarchical modulation is used to serve two users simultaneously. Let BS transmit to two users, i.e., user n and user m, at time slot t by employing two layers of HM. Assume that user j has a better channel than user i , i.e., $h_n \leq h_m$. Then, user n is assigned to QPSK constellation which we refer to as *base layer*. User m is assigned to 16-QAM constellation which we refer to as *incremental layer*. More information about hierarchical modulation can be found in [36] and references there in. Since two modulated signals are mixed before being transmitted, they interfere with each other at the receiver side. However, in [72], the authors propose a decoding technique to cancel the interference seen at the incremental layer. Specifically, when mixed signal reaches to the receivers, the data at the base layer is first decoded and removed. Hence, the data at the incremental layer does not suffer from the transmission at the base layer. In [72], the achievable rates for user n and user m are given respectively as follows:

$$R_n^b(t) = T_s \times BW \times \log\left(1 + \frac{h_n(t)P_{n,b}(t)}{h_n(t)P_{m,i}(t) + \sigma}\right), \tag{4.1}$$

$$R_m^i(t) = T_s \times BW \times \log\left(1 + \frac{h_m(t)P_{m,i}(t)}{\sigma}\right), \tag{4.2}$$

where $P_{n,b}(t)$ and $P_{m,i}(t)$ are the transmission powers for user n and m at the base and incremental layers, respectively. σ is the noise power and BW is the bandwidth of the channel. We assume that the BS transmits at fixed power and the total power consumption is equal to P, i.e., $P_{n,b}(t) + P_{m,i}(t) = P \quad \forall t$. $R_n^k(t)$ is upper-bounded such that $R_n^k(t) \leq \mu_{max}$ $\forall t$, $k \in \{b, i\}$. Note that when UM is applied at the physical layer, full power is assigned to a single user and the amount of data that can be transmitted in that case is given by,

$$R_n^{um}(t) = T_s \times BW \times \log\left(1 + \frac{h_n(t)P}{\sigma}\right). \tag{4.3}$$

where $n \in \{1, 2, \ldots, N\}$. Let $A_n(t)$ be the amount of data (bits or packets) arriving into the queue of user n at time slot t and $A_n(t) \leq A_{max}$ $\forall t$, and assume that $A_n(t)$ is a time and user independent stationary process. We denote the arrival rate vector as $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_N)$, where $\lambda_n = \mathbb{E}[A_n(t)]$. Let $\mathbf{Q}(t) = (Q_1(t), Q_2(t), \cdots, Q_N(t))$ denote the vector of queue sizes, where $Q_n(t)$ is the queue length of user n at time slot t. The dynamics of the queue of user n is given as,

$$Q_n(t+1) = [Q_n(t) + A_n(t) - R_n^k(t)]^+. \tag{4.4}$$

where $(x)^+ = \max(x, 0)$ and $k \in \{b, i\}$. Let Λ denote the *achievable rate region* (or rate region) defined as the closure of the set of all arrival rate vectors for which there exists an appropriate scheduling policy stabilizing the network.

4.3 Throughput Optimal scheduling

In this section, we give a throughput optimal scheduling algorithm when HM is employed. However, we start with the conventional Max-Weight algorithm employing UM to schedule a single user at every time slot.

Max-Weight with UM (MWUM): At time t, given $h_n(t)$ and $Q_n(t)$ for all $n \in \mathcal{N}$, schedule user n^* which has the maximum queue length and service rate product, i.e., [6]:

$$W_n^{um}(t) = Q_n(t)R_n^{um}(t) \tag{4.5}$$

$$n^* = \underset{n\in\mathcal{N}}{\text{argmax}}\ W_n^{um}(t). \tag{4.6}$$

We define $W_u^*(t) \triangleq W_{n^*}^{um}(t)$. Let Λ_u denote the rate region achieved by MWUM.

Max-Weight with HM (MWHM): At time t, given $h_n(t)$ and $Q_n(t)$ for all $n \in \mathcal{N}$ schedule *two* users (n^*, m^*) such that $h_{n^*}(t) \leq h_{m^*}(t)$ to maximize the sum of queue length and service rate products, i.e.,:

$$W_{n,m}^{hm}(t) = Q_n(t)R_n^b(t) + Q_m(t)R_m^i(t) \tag{4.7}$$

$$(n^*, m^*) = \underset{\substack{(n,m)\in\mathcal{N}, n\neq m \\ h_n(t)\leq h_m(t)}}{\text{argmax}}\ w_{n,m}^{hm}(t). \tag{4.8}$$

We define $W_h^*(t) \triangleq W_{n^*,m^*}^{hm}(t)$. Let Λ_h denote the rate region achieved by MWHM. Since BS has limited power budget, power allocation must be performed to determine $R_n^b(t)$ and $R_m^i(t)$.

Power Allocation with HM: Recall that Max-Weight type scheduling algorithms aim to maximize the weight $W_{n,m}^{hm}(t)$ (or $W_n^{um}(t)$) at each time slot. It is easy to determine the maximum weight achieved by UM, $W_{n^*}^{um}(t)$. However, the maximum weight under HM depends on power allocations $P_{n,b}^*(t)$ and $P_{m,i}^*(t)$. Without loss of generality, for a given pair of users, e.g., user n and user m such that $h_n(t) \leq h_m(t)$, the optimal power allocation maximizing the weight $w_{n,m}^{hm}(t)$ is obtained by solving the following optimization problem:

$$\max_{P_{n,b}(t), P_{m,i}(t)} \ Q_n(t)R_n^b(t) + Q_m(t)R_m^i(t) \tag{4.9}$$

$$\text{s.t.} \quad P_{n,b}(t) + P_{m,i}(t) = P \tag{4.10}$$

Note that $P_{n,b}^*(t)$ and $P_{m,i}^*(t)$ both have a non-zero value when (4.9)-(4.10) is a convex problem. Now, we give a Lemma which states the necessary conditions for this to hold.

For notational convenience, we drop the time index. Let us define,

$$B \triangleq \frac{h_n^2}{(h_n P_{m,i} + \sigma)^2} \text{ and } C \triangleq \frac{h_m^2}{(h_m P_{m,i} + \sigma)^2},$$

where $0 \leq P_{m,i}(t) \leq P$.

Lemma 4.1. *The problem* (4.9)-(4.10) *is a convex optimization problem when the following inequality is satisfied*

$$Q_n B \leq Q_m C \tag{4.11}$$

Proof. We show that the objective function in (4.9) is concave under the given condition. The objective function can be rewritten by noting that $P_{n,b} = P - P_{m,i}$. Since the parameters T_s and BW do not effect the concavity, we have the following objective function,

$$f = Q_n \log\left(1 + \frac{h_n(P - P_{m,i})}{h_n P_{m,i} + \sigma}\right) + Q_m \log\left(1 + \frac{h_m P_{m,i}}{\sigma}\right).$$

Taking the second derivative of f with respect to $P_{m,i}$ yields,

$$\frac{d^2 f}{dP_{m,i}^2} = Q_n B - Q_m C. \tag{4.12}$$

For concavity, $\frac{d^2 f}{dP_{m,i}^2}$ must be less than or equal zero, i.e., $\frac{d^2 f}{dP_{m,i}^2} \leq 0$. Thus, $q_n B \leq q_m C$. □

As long as the condition in Lemma 4.1 is satisfied, the optimal power allocation can be found by taking the first derivative of f and setting it to zero. The first derivative of f with respect to $P_{m,i}$ is given by,

$$\frac{df}{dP_{m,i}} = -Q_n \sqrt{B} + Q_m \sqrt{C} = 0. \tag{4.13}$$

Thus, we have,

$$P^*_{m,i} = \frac{\sigma(Q_m h_m - Q_n h_n)}{h_n h_m (Q_n - Q_m)}, \tag{4.14}$$

$$P^*_{n,b} = P - P^*_{m,i}. \tag{4.15}$$

Lemma 4.2. *If $\boldsymbol{\lambda} \in \Lambda_h$ (i.e., $\boldsymbol{\lambda}$ is feasible), then MWHM algorithm stabilizes the network and it is throughput optimal.*

Proof. We can write the following inequality by using the fact $([a]^+)^2 \leq (a)^2, \quad \forall a$:

$$\begin{aligned} Q_n^2(t+1) \leq & Q_n^2(t) + (R_{max})^2 + (A_{max})^2 \\ & - 2Q_n(t)[R_n(t) - A_n(t)] \end{aligned} \tag{4.16}$$

Define the following Lyapunov function and conditional Lyapunov drift:

$$L(\mathbf{Q}(t)) \triangleq \sum_{n=1}^{N} Q_n^2(t), \tag{4.17}$$

$$\Delta(t) \triangleq \mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) | \mathbf{Q}(t)\right]. \tag{4.18}$$

By using (4.16) and (4.17), one can show that the Lyapunov drift of the system satisfies the following inequality at every time slot,

$$\Delta(t) \leq B - \sum_n \mathbb{E}\left\{Q_n(t) R_n(t) | \mathbf{Q}(t)\right\} - \sum_n Q_n(t) \lambda_n \tag{4.19}$$

where $B = \frac{N}{2}((R_{max})^2 + (A_{max})^2)$. Note that the second term in the right hand side of (4.19) can be rewritten as follows when two users are scheduled:

$$\sum_n \mathbb{E}\left\{Q_n(t) R_n(t) | \mathbf{Q}(t)\right\} = \sum_{\substack{(i,j) \\ h_i \leq h_j}} \mathbb{E}\left\{Q_i(t) R_i(t) + Q_j(t) R_j(t) | \mathbf{Q}(t)\right\}$$

Now, it is easy to see that MWSC minimizes the right hand side of (4.19) at every time slot. Therefore, according to Lyapunov-Foster criteria, $\mathbf{Q}(t)$ process is positive recurrent

Markov chain and MWHM can stabilize the network whenever this is possible [8].

□

4.4 Max-Weight Algorithm with Dynamic Modulation (MWDM)

Note that MWHM can only be used when there is an inner point solution to the problem (4.9)-(4.10), i.e., $0 < P^*_{m,i} < P$. If the solution is on the boundary, i.e., $P^*_{m,i} = 0$ or $P^*_{m,i} = P$, then full power is assigned to a single user and HM is no longer employed, i.e., transmission to a single user is optimal.

Now, we propose Max-Weight algorithm with dynamic modulation (MWDM) that dynamically decides which coding (HM or UM) must be employed at every time slot. Let $W_d(t)$ and Λ_d be the maximum weight at time t and the rate region achieved by MWDM, respectively. MWDM is implemented as follows:

- Step 1: The scheduler applies MWUM and finds the maximum weight $W_u(t)$ by using (4.5) and (4.6).
- Step 2: The scheduler applies MWHM as follows: For every pair of users (n, m), find $W^{hm}_{n,m}(t)$:
  - if $h_n \leq h_m$, then user n is embedded at the base layer whereas user m is the incremental layer or vice versa.
  - Check whether the condition in Lemma 4.2 is satisfied.
  - If not, $W^{hm}_{n,m}(t) = \max\{W^{um}_n, W^{um}_m\}$ and UM is employed.
  - Otherwise, determine the optimal power allocation $P^*_{m,i}$ and $P^*_{n,b}$ according to (4.14) and (4.15), respectively.
  - Then determine $W^{hm}_{n,m}(t)$ according to (4.7).
- Step 3: After finding $W^{hm}_{n,m}$ for all pairs, determine $W^*_h(t)$.

- Step 4: If $W_h^*(t) > W_u^*(t)$, then $W_d^*(t) = W_h^*(t)$ and HM is employed. Otherwise, $W_d^*(t) = W_u^*(t)$ and UM is employed.

Note that the layer assignment within MWDM algorithm is done by only considering the relative measures of h_n and h_m. When transmit power is another parameter, one can design a different algorithm which considers SINR values of users.

Let us define the expected weights achieved by MWDM and MWUM as $\mathbb{E}[W_d(t)]$ and $\mathbb{E}[W_u(t)]$, respectively.

Theorem 4.1. *The achievable rate region of MWUM algorithm is a subset of the achievable rate region of MWDM, i.e.,* $\Lambda_u \subseteq \Lambda_d$.

Proof. We use the theorem given in [40] to prove the Lemma. According to the theorem, if $\mathbb{E}[W_d(t) \geq \mathbb{E}[W_u(t)]$, then MWDM can at least achieve the rate region of MWUM. The average weight achieved by MWDM is always greater than or equal to the weight achieved by MWUM since MWDM can either apply HM or UM according the maximum weight. Thus, the following inequality holds at every time slot,

$$W_d^*(t) \geq W_u^*(t) \tag{4.20}$$

Taking the expectation of both sides of (4.20) yields that $\mathbb{E}[W_d^*(t)] \geq \mathbb{E}[W_u^*(t)]$. □

Hence, MWDM can be used to increase the total network throughput.

4.4.1 A low complexity algorithm

Note that the implementation of MWDM algorithm requires the calculation of the optimal power allocation and the weight of every pair of users. This requires a computational complexity of $O(N^2)$. Now, we propose a low complexity algorithm (L-MWDM) that has a computational complexity of $O(N)$.

L-MWDM algorithm is implemented as follows:

- Step 1: Determine user n^* according to (4.5) and (4.6) which is the optimal user selected under UM. Determine $W_u(t)$ by using (4.5).

- Step 2: For every user $m \neq n^*$, $m \in \mathcal{N}$ do:
  - if $h_{n^*} \leq h_m$, then user n^* is embedded at the base layer whereas user m is embedded at the incremental layer or vice versa.
  - Check the condition in Lemma 2 is satisfied.
  - If it is not, $w^{hm}_{n^*,m}(t) = \max\{w^{um}_{n^*}, w^{um}_{m}\}$ and UM is employed.
  - Otherwise, determine the optimal power allocation $P^*_{n^*,b}$ and $P^*_{m,i}$ according to (4.14) and (4.15), respectively. Then, determine $w^{hm}_{n^*,m}(t)$ according to (4.7).
- Step 3: After finding $w^{hm}_{n^*,m}(t)$ for every user $m \neq n^*$ determine $W_h(t)$.
- Step 4: If $W^*_h(t) > W^*_u(t)$, then $W_d(t) = W_h(t)$ and HM is employed. Otherwise, $W_d(t) = W_u(t)$ and UM is employed.

Note that the difference between MWDM and L-MWDM is that MWDM checks the weights achieved by every pair of users. Hence, its complexity increases quadratically with the number of users. On the other hand, L-MWDM calculates the weights assuming that user n^* is always scheduled. Thus, its complexity is linear with N. However, the maximum weight obtained with MWDM is always greater or equal to that of L-MWDM.

4.5 Simulation Results

In our simulations, we model a single cell downlink transmission utilizing high data rate (HDR) [4]. The base station serves 20 users and keeps a separate queue for each user. Time is slotted with length $T_s = 1.67$ ms as defined in HDR specifications. We set $BW = 1.25$ MHz, $P = 10$ Watts and $\sigma = 10^{-6}$ Watts. Packets arrive at each slot according to Bernoulli distribution. The size of a packet is set to 128 bytes which corresponds to the size of an HDR packet. The wireless channel between the BS and each user is modeled as a *correlated* Rayleigh fading according to Jakes' model with different Doppler frequencies varying randomly between 5 Hz and 15 Hz. Figure

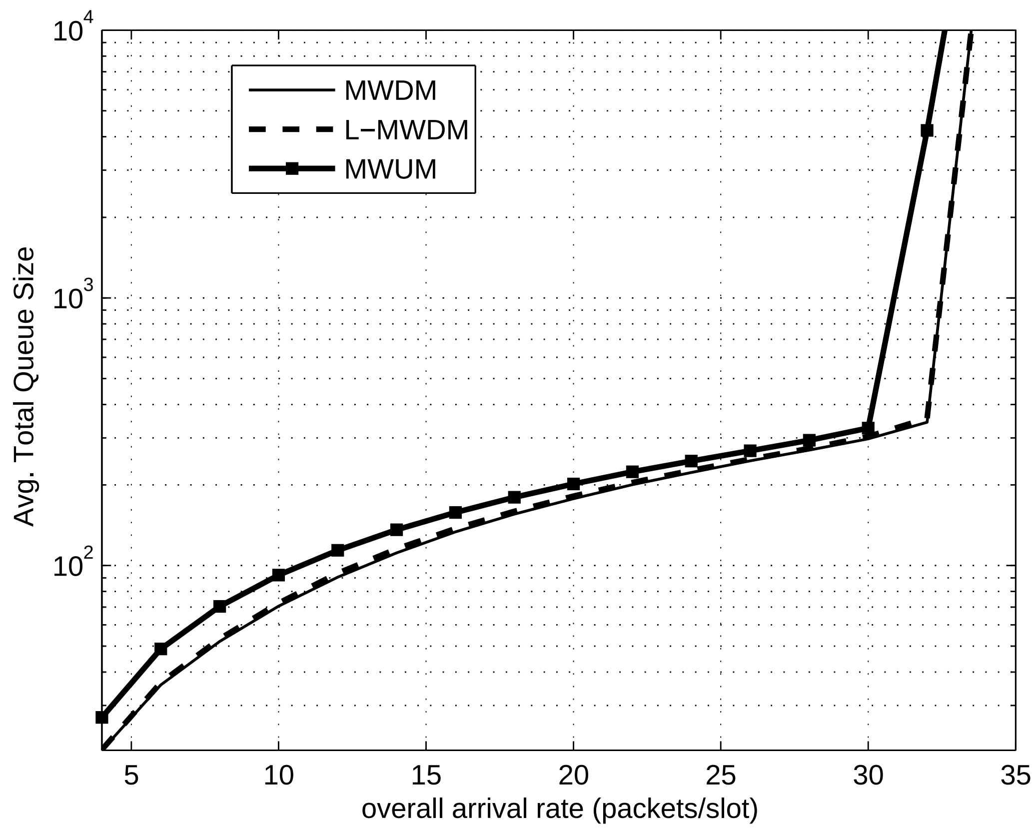

Figure 4.1: Average total queue sizes vs. overall mean arrival rate.

4.1 depicts the maximum arrival rate that can be supported by MWDM, L-MWDM and MWUM. Clearly, as the overall arrival rate exceeds 30 packets/slot queue sizes suddenly increase with MWUM and the network becomes unstable. However, MWDM and L-MWDM improves over MWUM by supporting the overall arrival rate of up to 32 packets/slot. Therefore, MWDM can achieve a larger rate region than MWUM as verified analytically in Theorem 4.1. Figure 4.1 also shows the sum of the queue lengths vs. mean of overall arrival rate (packets/slot), i.e., the increase is about 1.2 Mbps. As the average arrival rate increases the average queue backlogs increase as well in all algorithms. Following Littles' Law, larger queue backlogs yield longer network delays. However, due to the possibility of serving more than one queue at a time, MWDM and L-MWDM outperform MWUM in terms of the average delay. This result indicates that MWDM and L-MWDM are better techniques for delay sensitive applications. In addition, the delay performance of MWDM and L-MWDM are very close. Recall that

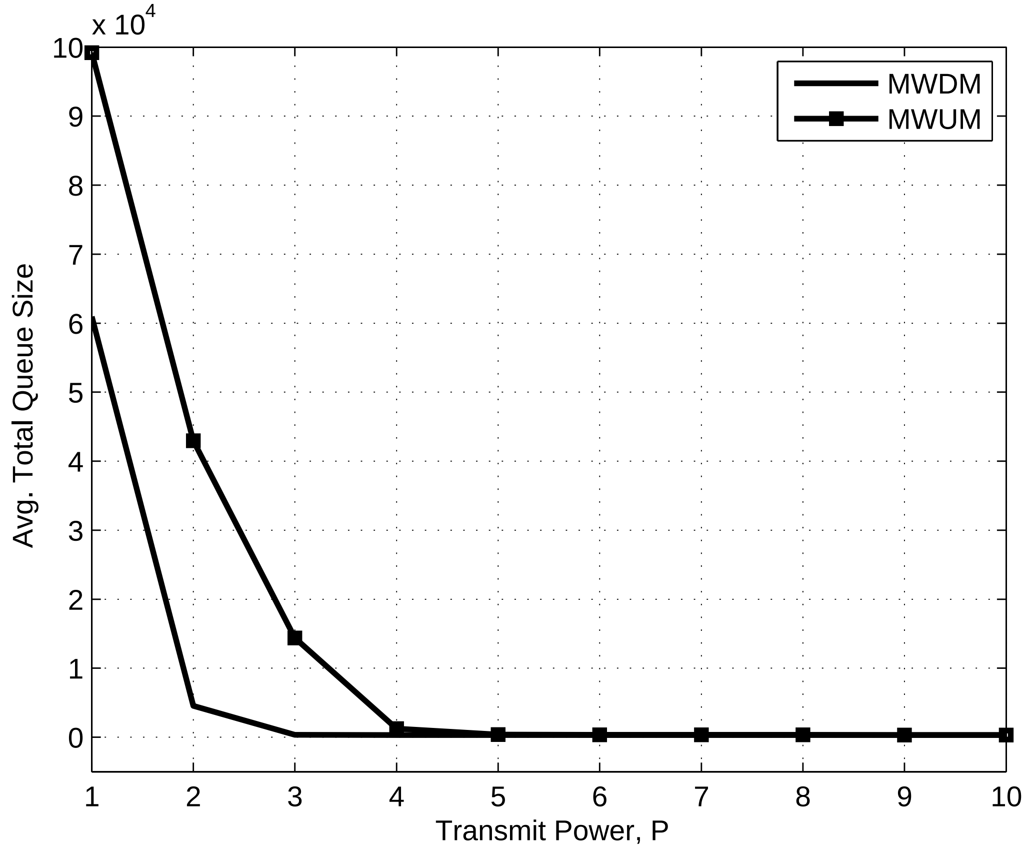

Figure 4.2: Average total queue sizes vs. transmit power, P

L-MWDM always schedules user n^* which is the optimal user in MWUM algorithm. Similar to L-MWDM, MWDM schedules user n^* most of the time.

Figure 4.2 shows the sum of the queue lengths vs. transmit power, P in Watts when the overall mean arrival rate is 28 packets/slot. Clearly, as P increases, the average queue size with both MWDM and MWUM decreases as well. However, the steady state queue length is achieved with MWUM at $P = 4$ whereas MWDM reaches to the steady state at $P = 3$. As a conclusion, HM based scheduling requires less power than the uniform constellation based scheduling to stabilize the network.

4.6 Chapter Summary

In this work, we investigate the advantages of transmissions of more than one data streams simultaneously in a network stability problem. We propose to use hierarchical modulation with Max-Weight algorithm when two user are scheduled simultaneously. First, we give the optimal power allocation among users. Then, we show that the proposed algorithm can support higher user traffic compared to the conventional Max-Weight Algorithm. In addition, we demonstrate that with the proposed algorithm the average delay reduces dramatically. HM is a good technique for scheduling problem especially when the number of users in the system is large and BS transmits high transmission powers.

Chapter 5

Joint Scheduling and Selective Channel Feedback

In previous Chapters, we have investigated the optimality of opportunistic scheduling algorithms (including Max-Weight type scheduling algorithms) under the assumption that the complete network state information (both CSI and queue length information of all users) is available at the scheduler. The majority of existing works employing Max-Weight algorithm make this assumption and the associated overhead of channel probing usually is not taken into account. In this Chapter, we study Max-Weight algorithm with incomplete CSI. We design Scheduling and Selective Feedback algorithm (SSF) taking into account the overhead due to the acquisition of CSI. SSF algorithm collects CSI from only those users with sufficiently good channel quality so that it always schedules the user with the highest queue backlog and channel rate product at every slot. We characterize the achievable rate region of SSF algorithm by showing that SSF supports $1 + \epsilon$ fraction of the rate region when CSI from all users are collected. We also show that the value of ϵ depends on the expected number of users which do not send back their CSI to the base station. For homogenous and heterogeneous channel conditions, we determine the minimum number of users that must be present in the network so that the rate region is expanded, i.e., $\epsilon > 0$. We also demonstrate numerically in a realistic simulation setting that this rate region can be achieved by collecting CSI from less than 50% of all users in a CDMA based cellular network utilizing high data rate

(HDR) protocol.

5.1 Overview

To exploit the advantages of multi-user diversity in the downlink the base station requires the instantaneous channel state information (CSI) of users. As other opportunistic scheduling algorithms, throughput-optimal Max-Weight algorithm requires complete CSI. A common assumption in the literature is that the *exact* and *complete* channel state information of all users is available at every time slot. However, in general base station is unaware of the users' channel state information, which must be acquired by consuming a fraction of resource which is otherwise used for data transmission. Hence, acquiring full CSI is usually very costly.

As mentioned in Chapter 2, there has been significant interest in developing joint feedback and scheduling algorithms for wireless systems [48], [49], [50], [51] [52]. We refer to the readers to [53] and the references therein for more information on acquiring limited feedback. Most prior works study network capacity and feedback tradeoff by assuming infinitely backlogged user queues. However, when network stability problem is considered this trade-off cannot be analyzed in the same way since the queue size of each user should be taken into account.

In the sequel, we review the related work that studies the network control problem with incomplete CSI. In [73], the authors proposed a joint scheduling and channel feedback algorithm stabilizing the network by allowing the base station to receive CSI from a subset of users. However, throughput-optimality of that algorithm can only be shown under certain conditions, i.e., when channel distributions are known. In [74], the authors proposed a variant of the algorithm in [73] to analyze queue-overflow performance with limited CSI. In [75], a feedback allocation algorithm was proposed for multi-channel system where only a limited number of CSI can be acquired at a time and channel distributions are known at the BS. Note that the algorithms in [73], [75], [74] require channel statistics to achieve throughput-optimality, which is impossible in practice. In [76], the authors proposed a scheduling algorithm under imperfect CSI in single-hop networks. In this work, the authors aimed to obtain a tradeoff between energy consumption and network capacity, when energy cost of acquiring CSI is taken into account.

In [77], the authors proposed joint channel estimation and opportunistic scheduling algorithm by assuming i.i.d. channel processes. In [78], the authors proposed to estimate the channel statistics by using some portion of the time slots for observation slot with some probability over i.i.d. channels. Unlike [77] and [78], we do not estimate channel statistics at all. Hence, our algorithm is robust under more general channel conditions such as time-correlated or non-stationary channels. In [79], it was assumed that wireless channels evolve as Markov-modulated ON/OFF processes. With this assumption, a exploitation-exploration trade-off was investigated. Similarly, in [80], a two state discrete time Markov chain with a bad state which yields no reward and a good state which yields reward was considered. The performance of [79] and [80] depends on the underlaying stochastic process of the channel evolves according to a fixed stationary process such ergodic Markov chain. In practice, such an assumption does not hold most of the time. For instance, the measurement study in [81] shows that the wireless channel exhibits time-correlated and non-stationary behavior.

Unlike the works above, we assume a more flexible channel acquiring model where receiving CSI from a single user consumes a certain portion of resource which is otherwise used for transmission of data (e.g., β fraction of a time slot). A similar model was also considered in [82]. However, in that work the network stability problem was not investigated. In this sense, the most related work is [83], where the optimal feedback and scheduling scheme for a single-channel downlink is determined. Specifically, in [83], the server has a cost for obtaining CSIs and it gains a reward (defined as queue weighted throughput) for each scheduled user. The problem of finding optimal set of users from which CSI is obtained. Then, the user scheduled in each slot is transformed into an optimal stopping time problem which is solved as a Markov Decision Process (MDP) given a priori channel distributions of all users. Unlike our algorithm, the optimal algorithm in [83] has high computational complexity due to its MDP formulation especially when the number of channel states is large. However, the performance of

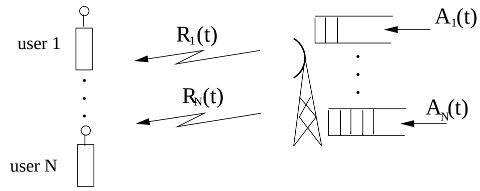

Figure 5.1: Cellular Network Model

our algorithm does not suffer from larger number of channel states and, interestingly, it improves as the the number of channel state increases.

In this work, our contributions are summarized as follows:

- We develop a *scheduling and selective feedback (SSF)* algorithm that schedules users according to the queue-lengths at the base station without full CSI. We show that SSF algorithm achieves a fraction $1 + \epsilon$ of the achievable rate region of Max-Weight algorithm acquiring channel state information from all users.

- For homogeneous and heterogeneous channels, we identify the minimum number of users that must be present in the system to reap benefits of SSF algorithm.

- We evaluate the performance of the proposed SSF algorithm with numerical experiments, and demonstrate that SSF can support higher arrival rates than that of another prominent algorithm which was proposed in [83].

5.2 System Model

We consider a multiuser downlink network with N users and a single base station (BS), where users wish to receive data from the BS via the downlink channel as shown in Figure 5.1. We consider a time-slotted system where the time slot is the resource to be shared among different nodes. We adopt a non-interference model where only one node is transmitting at any given time. Random channel gains between base station and other nodes in the network are assumed to be independent and identically distributed (iid) across time according to a general distribution and independent across nodes with values taken from a finite set. Moreover, we assume that channel gains are time-varying, but fixed over the time slot duration. Practical systems, such as CDMA/HDR system [4], implement adaptive modulation schemes which adjust signal constellation, coding rate/scheme, etc., relative to the channel fading. When the instantaneous Signal-to-Noise-Ratio (SNR) falls within a given range, the associated signal constellation is transmitted. Since only a discrete finite set of L constellations is available, only a fixed set of data rates $\mathcal{R} = \{r_1, r_2, \ldots, r_L\}$ can be supported. Assume that without loss of generality, $r_k > r_l$ if $l > k$. Let $R_n(t) \in \mathcal{R}$ be the supported data rate on the downlink channel to node n at time slot t. The steady state distribution of supported channel rates for node n is given as $\Pr[R_n(t) = r_l] = p_n^l$, for all $l = 1, \ldots, L$, where all rates are observed with non-zero probability, i.e., $p_n^l > p^{min}$, and $p^{min} > 0$, for all l and n.

The base station does not have the knowledge of channel states of the receivers at the beginning of the slot, but it has to acquire this information. At the beginning of each time slot, t, base station broadcasts a pilot signal with a fixed and known power. Each node n determines its supportable data rate $R_n(t)$ by measuring the received SNR. Acquiring the supportable data rates from the receivers consumes resources. In this work, we adopt the same model as in [83] and assume that acquiring channel state information from a receiver consumes a fraction β of a slot. Let $\mathcal{S}(t)$ be the set of users from which channel state information is acquired, where $S(t)$ is the cardinality of set $\mathcal{S}(t)$. Note that $0 < S(t) \leq N$. Hence, if base station decides to transmit to node n, then it can transmit only $(1 - \beta S(t))R_n(t)$ bits. Clearly, we assume that the number of users is upper bounded by $N < \frac{1}{\beta}$. Since the transmitter can transmit to a receiver only after knowing its channel state, there is a trade off between the number of nodes from which channel state information is acquired and the fraction of time left for actual data transmission.

We use indicator variable $\mathcal{I}_n(t)$ to represent the scheduler decision, where $\mathcal{I}_n(t) = 1$ if node $n \in \mathcal{S}(t)$ is scheduled for transmission in slot t, and $\mathcal{I}_n(t) = 0$ otherwise. By definition, at most one user can be served at a time slot, i.e., $\sum_{n=1}^{N} \mathcal{I}_n(t) = 1$, $\forall t$. Assuming unit slot length, the amount of data that can be transmitted to user n when channel states of $S(t)$ users are acquired at slot t is,

$$D_n(t) = (1 - \beta S(t))R_n(t)\mathcal{I}_n(t). \tag{5.1}$$

Base station maintains a separate queue for each node n. Packets arrive according a stationary arrival process that is independent across users and time slots. Let $A_n(t)$ be the amount of data arriving into the queue of user n at time slot t. Let $\mathbf{Q}(t) = (Q_1(t), Q_2(t), \ldots, Q_N(t))$ denote the queue length vector. The dynamics of queue length for node n is given as

$$Q_n(t+1) = [Q_n(t) + A_n(t) - D_n(t)]^+, \tag{5.2}$$

where $[x]^+ = \max(x, 0)$. We say that the system is stable if the mean queue length for all the receiver nodes is finite. In their seminal paper, Tassiulas and Ephremides [6] have shown that Max-Weight algorithm could ensure stability of the user buffers whenever this is at all possible. Max-Weight scheduling policy schedules the user n^* for which the transmission rate weighted by the queue length is the maximum, i.e.,

$$n^* = \operatorname*{argmax}_n W_n(t) = \operatorname*{argmax}_n Q_n(t) R_n(t), \tag{5.3}$$

However, Max-Weight policy neglects the cost of channel state acquisition and thus, assumes perfect knowledge of channel states at the beginning of time slot. In this Chapter, we develop an algorithm that can find the user with maximum weighted rate, i.e., $n^* = \operatorname{argmax}_n W_n(t)$ without acquiring the channel states from all users. Consequently, in the following section, we characterize the stability region of the system defined by the set of arrival rates $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)$, where $\lambda_n = \mathbb{E}[A_n(t)]$ such that our proposed selective channel state acquisition and scheduling policy can stabilize.

Discussion: The convex hull of the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exists an appropriate scheduling policy that stabilizes the network is called *achievable rate region*. When the cost of acquiring channel state information is neglected, i.e., $\beta = 0$, the achievable rate region is the largest, since the resources are used completely for only actual data transmission. Let Λ_h denote this *hypothetical rate region*, the boundary of which can never be achieved in real systems [84]. On the other extreme, let Λ_f denote the achievable rate region achieved when channel state information is acquired from *all* users with non-zero acquisition cost, i.e., $\beta > 0$. Clearly, $\Lambda_f \subset \Lambda_h$.

Finally, let Λ denote the achievable rate region of any other scheduling algorithm that selectively acquires channel state information from only a portion of users. In the next section, we propose a scheduling and selective feedback algorithm, and show that its achievable rate region satisfies $\Lambda_f \subset \Lambda \subseteq \Lambda_h$. Hence, our proposed algorithm is more efficient than Max-Weight algorithm acquiring channel state information from all users.

5.3 Scheduling and Selective Feedback (SSF) Algorithm

In this section, we propose a joint scheduling and selective feedback algorithm that determines at each time slot the user with the maximum weighted rate without acquiring channel states from every user. The key idea is to identify and eliminate those users which has no possibility of having the maximum weighted rate, and acquire channel states from the rest of the users. For this purpose, algorithm follows a two step procedure. In the first step, base station identifies the user with the largest downlink queue length, say user n, and acquires its channel state, i.e., $R_n(t)$. Base station broadcasts $R_n(t)$ to all users, and asks them to forward their channel state only if their rates are larger than $R_n(t)$. The procedure is explained in detail in Algorithm 1. The following Theorem shows that SSF and Max-Weight algorithms make the same scheduling decision at every time slot.

Theorem 5.1. *SSF algorithm schedules the user with the maximum weighted rate at every time slot.*

Proof. Let $W_{i^*}(t)$ be the weighted rate of the user with the maximum queue size at time slot t, i.e., $i^* = \text{argmax}_{1 \leq n \leq N} Q_n(t)$. Also let $W^*(t)$ be the maximum weighted rate among all users, i.e., $W^*(t) = \max_{1 \leq n \leq N} W_n(t)$. SSF algorithm schedules the user with the maximum weighted rate from set of users $\mathcal{S}(t) = \{j : R_j(t) \geq R_{i^*}(t)\}$. For any user $k \notin \mathcal{S}(t)$, $R_k(t) < R_{i^*}(t)$, and since $Q_{i^*}(t) > Q_k(t)$, then $W_k(t) < W_{i^*}(t)$. Hence, $\max_{n \in \mathcal{S}(t)} W_n(t) = W^*(t)$. □

Algorithm 1: SSF Algorithm

At every time slot, do:
(1) Selective Feedback:

Step 1: Determine the user which has the maximum queue length,

$$i^* \triangleq \operatorname*{argmax}_{1\le i\le N}\{Q_i(t)\}$$

Step 2: Acquire the channel state of user i^*, i.e., $R_{i^*}(t)$.

Step 3: Broadcast the value of $R_{i^*}(t)$.

Step 4: The users with rates higher than $R_{i^*}(t)$ report their channel states to base station.

$$\mathcal{S}(t) \triangleq \{1 \le j \le N : R_j(t) > R_{i^*}(t)\}.$$

(2) scheduling decision:
Base station schedules the user n^* with the maximum weighted rate:

$$n^* = \operatorname*{argmax}_{n\in\mathcal{S}(t)\cup\{i^*\}} W_n(t), \tag{5.4}$$

i.e., $\mathcal{I}_{n^*}(t) = 1$, and updates queue lengths according to (5.2).

5.3.1 Achievable Rate Region of SSF Algorithm

Let $M(t)$ denote the number of users which *do not send* their channel state information at time t. The total fraction of time slot consumed for acquisition of channel state information is calculated as follows: A fraction β of time slot is used to acquire channel state information of user i^*, and another fraction β of time slot is used to broadcast $R_{i^*}(t)$. Then, those users with rates higher than $R_{i^*}(t)$ forward their channel state information, where each such feedback also consumes a fraction β of time slot. Hence, in total a fraction $(N - M(t) + 1)\beta$ of a time slot is consumed for acquisition of channel state information. The following theorem characterizes the achievable rate region of SSF algorithm, Λ, as compared to that of Max-Weight algorithm, i.e., Λ_f.

Theorem 5.2. *SSF algorithm can support a fraction* $(1 + \epsilon)$ *of the rate region* Λ_f*, i.e.,* $\Lambda \subseteq (1 + \epsilon)\Lambda_f$*, where*

$$\epsilon = \frac{\beta\left(\mathbb{E}[M(t)] - 1]\right)}{1 - \beta N}. \tag{5.5}$$

Proof. We first define the following two functions:

$$f_s(\boldsymbol{Q}(t)) = \mathbb{E}\left[\sum_{n \in \mathcal{S}(t)} (1 - \beta S(t)) W_n(t) I_n(t) | \boldsymbol{Q}(t)\right],$$

$$f_m(\boldsymbol{Q}(t)) = \mathbb{E}\left[\sum_{n \in \mathcal{N}} (1 - \beta N) W_n(t) I_n(t) | \boldsymbol{Q}(t)\right],$$

where the expectation is taken with respect to the randomness of channel variations and scheduling decisions. Given $\boldsymbol{Q}(t)$, both Max-Weight algorithm with full CSI and SSF schedules the same user with the maximum weighted rate at every time slot. Hence, the value of $W_n(t)I_n(t)$ is the same for both functions, and the only difference between $f_s(\boldsymbol{Q}(t))$ and $f_m(\boldsymbol{Q}(t))$ appears in the number of reported CSI. The performance of SSF algorithm is determined by using the following theorem proven in [40].

Theorem 5.3. *[40] If for some $\epsilon > 0$ SSF algorithm guarantees*

$$f_s(\boldsymbol{Q}(t)) \geq (1+\epsilon) f_m(\boldsymbol{Q}(t))$$

for all $\boldsymbol{Q}(t)$, then SSF can achieve $1+\epsilon$ fraction of the rate region Λ_f, i.e., $\Lambda \subseteq (1+\epsilon)\Lambda_f$.

Note that $f_s(\boldsymbol{Q}(t))$ can be rewritten as follows:

$$\begin{aligned} f_s(\boldsymbol{Q}(t)) &= \mathbb{E}\left[\sum_n (1-\beta(N+1-M(t)))W_n(t)I_n(t)|\boldsymbol{Q}(t)\right] \\ &= f_m(\boldsymbol{Q}(t)) + \mathbb{E}\left[\sum_n (\beta M(t)-\beta)W_n(t)I_n(t)|\boldsymbol{Q}(t)\right] \end{aligned} \tag{5.6}$$

Now, we consider the value of $f_s(\boldsymbol{Q}(t))/f_m(\boldsymbol{Q}(t))$ such that

$$\begin{aligned} &f_s(\boldsymbol{Q}(t))/f_m(\boldsymbol{Q}(t)) = \frac{f_m(\boldsymbol{Q}(t)) + \mathbb{E}\left[\sum_{n\in\mathcal{S}(t)} (\beta M(t)-\beta)W_n(t)I_n(t)|\boldsymbol{Q}(t)\right]}{f_m(\boldsymbol{Q}(t))} \\ &= 1 + \frac{\mathbb{E}\left[\sum_n (\beta M(t)-\beta)W_n(t)I_n(t)|\boldsymbol{Q}(t)\right]}{f_m(\boldsymbol{Q}(t))}, \end{aligned} \tag{5.7}$$

where

$$\begin{aligned} &\mathbb{E}\left[\sum_{n\in\mathcal{S}(t)} (\beta M(t)-\beta)W_n(t)I_n(t)|\boldsymbol{Q}(t)\right] = \\ &\sum_{m=0}^{N-1} (\beta m-\beta)\mathbb{E}\left[\sum_{n\in\mathcal{S}(t)} W_n(t)I_n(t)|\boldsymbol{Q}(t), M(t)=m\right] \times \Pr[M(t)=m] \end{aligned} \tag{5.8}$$

Note that

$$\mathbb{E}\left[\sum_{n\in\mathcal{S}(t)} W_n(t)I_n(t)|\boldsymbol{Q}(t), M(t)=m\right] = \frac{f_m(\boldsymbol{Q}(t))}{1-\beta N}$$

Hence, (5.8) can be rewritten as follows:

$$
\begin{aligned}
(5.8) &= \frac{f_m(\boldsymbol{Q}(t))}{1-\beta N} \sum_{m=0}^{N-1} [(\beta m - \beta)] \Pr[M(t) = m] \\
&= \frac{\beta f_m(\boldsymbol{Q}(t)) \left(\mathbb{E}[M(t)] - 1]\right)}{1-\beta N}
\end{aligned}
$$

Thus we have,

$$
f_s(\boldsymbol{Q}(t))/f_m(\boldsymbol{Q}(t)) \geq 1 + \frac{\beta \left(\mathbb{E}[M(t)] - 1\right)}{1-\beta N} \tag{5.9}
$$

Hence, the proposed algorithm can support $(1+\epsilon)$ fraction of the rate region Λ_f where $\epsilon = \frac{\beta\left(\mathbb{E}[M(t)]-1\right)}{1-\beta N}$. This completes the proof. □

Corollary 5.1. *The upper bound on ϵ is given by,*

$$
\epsilon_{up} = \frac{\beta N}{1-\beta N} \tag{5.10}
$$

According to Theorem 5.2, the performance of SSF algorithm depends on the expected number of users from which channel state information is not acquired, i.e., $\mathbb{E}[M(t)]$. Clearly, if $\mathbb{E}[M(t)] > 1$, then $\epsilon > 0$. Next, we calculate $\mathbb{E}[M(t)]$ under homogeneous and heterogeneous channel models.

5.3.2 Performance of SSF with Homogenous Channels

We first calculate the performance of SSF algorithm by considering a homogenous channel model, where the channel state probability distributions are the same for all users, i.e.,

$$
p_n^k = p_k, 1 \leq n \leq N.
$$

Lemma 5.1. *When channels are homogenous,* $\mathbb{E}[M(t)]$ *is given as follows:*

$$\mathbb{E}[M(t)] = \left[p_1 + p_2 \sum_{k=2}^{L} p_k + p_3 \sum_{k=3}^{L} p_k + \cdots + p_L^2\right](N-1). \tag{5.11}$$

Proof. Let $\mathbb{E}[M(t)|R_{i^*}(t) = r_k, n = i^*]$ be the conditional expectation of number of users from which channel state information is *not* acquired when the user with the maximum queue length and its channel state are given. Note that for homogenous channels, the following equality holds,

$$\mathbb{E}[M(t)|R_{i^*}(t) = r_k, n = i^*] = \mathbb{E}[M(t)|R_{i^*}(t) = r_k]. \tag{5.12}$$

The value of $\mathbb{E}[M(t)]$ is simply the expectation of (5.12) over the channel state distribution:

$$\mathbb{E}[M(t)] = \sum_{k=1}^{L} \mathbb{E}[M(t)|R_{i^*}(t) = r_k] \Pr[R_{i^*}(t) = r_k].$$

Note that if $R_{i^*}(t) = r_1$, then SSF does not acquire channel state information from other users in the network, since user i^* is the user with the maximum weight. Hence with probability, $\Pr[R_{i^*}(t) = r_k] = p_1$, the remaining $N-1$ users do not report their channel states, i.e., $\mathbb{E}[M(t)|R_{i^*}(t) = r_1] = (N-1)$. Meanwhile, if $R_{i^*}(t) = r_2$, user $j \neq i^*$ does not report its channel state if $R_j(t) \leq r_2$, which occurs with probability $\Pr[R_j(t) \leq r_2] = \sum_{k=2}^{L} p_k$. Thus, $\mathbb{E}[M(t)|R_{i^*}(t) = r_2] = (\sum_{k=2}^{L} p_k)(N-1)$. Thus, in general, for homogeneous channels, the conditional distribution of random variable $M(t)$ given that the channel state of the user with the maximum queue length is r_l, is a binomial distribution with probability $\sum_{k=l}^{L} p_k$. Then, $\mathbb{E}[M(t)]$ is as follows,

$$\mathbb{E}[M(t)] = \left[p_1 + p_2 \sum_{k=2}^{L} p_k + p_3 \sum_{k=3}^{L} p_k + \ldots + p_L^2\right](N-1)$$

$\square$

Next, we consider the special case when the channels are homogenous and channel state probabilities are *uniformly* distributed, i.e., $p_k = \frac{1}{L}$ for all k.

Lemma 5.2. *When channels are homogenous and uniformly distributed,* $\mathbb{E}[M(t)]$ *is given as follows:*

$$\mathbb{E}[M(t)] = \left[\frac{1}{2} + \frac{1}{2L}\right](N-1). \tag{5.13}$$

Proof. If all channels are uniformly distributed, then $p_k = \frac{1}{L}$. Hence, by (5.11) $\mathbb{E}[M(t)]$ is obtained as

$$\begin{aligned}\mathbb{E}[M(t)] &= \left[\frac{1}{L} + \frac{1}{L}\sum_{k=2}^{L}\frac{1}{L} + \frac{1}{L}\sum_{k=3}^{L}\frac{1}{L} + \ldots + \frac{1}{L^2}\right](N-1) \\ &= \left[\frac{1}{L} + \frac{1}{L^2}(L-1) + \frac{1}{L^2}(L-2) + \ldots + \frac{1}{L^2}\right](N-1)\end{aligned} \tag{5.14}$$

Re-arranging (5.14), we obtain

$$\mathbb{E}[M(t)] = \left[\frac{1}{L} + \frac{L(L-1)}{2L^2}\right](N-1) = \left[\frac{1}{2} + \frac{1}{2L}\right](N-1) \tag{5.15}$$

□

Corollary 5.2. *For homogenous channels with uniform channel state distribution,* ϵ *is given as:*

$$\epsilon = \frac{\beta\left(\left[\frac{1}{2} + \frac{1}{2L}\right](N-1) - 1\right)}{1 - \beta N}. \tag{5.16}$$

To give an idea of the description of the value of ϵ, let us consider a typical HDR system where there are 11 SNR levels (i.e., $L = 11$), and we assume $N = 15$ and $\beta = 0.02$. Then by using (5.16), $\epsilon = 0.19$. That means we can support an arrival rate which is 19% higher than that of Max-Weight algorithm with full CSI. Note that the minimum value of ϵ is zero which occurs at $\beta = 0$. Intuitively, if there is no probing cost then there will be no advantage of employing SSF algorithm since Max-Weight algorithm with complete CSI results in also zero cost. As β increases, ϵ increases as well. Thus, higher probing cost (i.e., higher β) more advantages using SSF algorithm.

We now give our main result for homogenous channels.

Theorem 5.4. *For homogeneous channels, SSF algorithm guarantees to achieve a larger rate region than Λ_f, i.e., $\epsilon > 0$, if the number of users is greater that 3, i.e., $N > 3$.*

Proof. We prove the theorem by showing that $\mathbb{E}[M(t)]$ is a jointly convex function of $(p_1, p_2, \ldots, p_L)$, and the minimum of this convex function is achieved when channels are uniformly distributed.

Lemma 5.3. $\mathbb{E}[M(t)]$ *is a jointly convex function of* $(p_1, p_2, \ldots, p_L)$.

Proof. By noting $p_1 = 1 - \sum_{k=2}^{L} p_k$, the Hessian of $\mathbb{E}[M(t)]$ in (5.11) can be given as follows,

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & 2 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & 1 & 2 & 1 & \cdots & 1 \\ 0 & 1 & 1 & 1 & 1 & 2 & \cdots & 1 \\ 0 & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 1 & 1 & 1 & 1 & \cdots & 2 \end{pmatrix} (N-1)$$

Now, we show that H is positive definite matrix. Let $\mathbf{x} = [x_1\ x_2\ x_3 \ldots\ x_L]$ be any vector and $\mathbf{x} \in \mathbb{R}^{L-1}$. If $\mathbf{x}H\mathbf{x}^T > 0$ then, H is positive definite matrix and $\mathbb{E}[M(t)]$ is convex function of $p_1, p_2, \ldots, p_L$ [70]. By simple manipulations, we obtain

$$\mathbf{x}^T H \mathbf{x} = \left[\sum_{l=1} x_l^2 + \left(\sum_{l=1} x_l \right)^2 \right] (N-1) > 0. \tag{5.17}$$

□

Lemma 5.4. $\mathbb{E}[M(t)]$ *has the minimum value when channels are uniformly distributed.*

Proof. We already showed that $\mathbb{E}[M(t)]$ is jointly convex function of $(p_1, p_2, \ldots, p_L)$. Hence, the first order conditions can be obtained as follows by noting that $p_1 = 1 - \sum_{k=2}^{L} p_k$

$$\begin{aligned}
\frac{\partial \mathbb{E}[M(t)]}{\partial p_2} &= -1 + 2p_2 + (p_3 + p_4 + \ldots + p_L) = 0 \\
\frac{\partial \mathbb{E}[M(t)]}{\partial p_3} &= -1 + 2p_3 + (p_2 + p_4 + \ldots + p_L) = 0 \\
\frac{\partial \mathbb{E}[M(t)]}{\partial p_4} &= -1 + 2p_4 + (p_2 + p_3 + p_5 + \ldots + p_L) = 0 \\
&\vdots \\
\frac{\partial \mathbb{E}[M(t)]}{\partial p_L} &= -1 + 2p_L + (p_2 + p_3 + p_5 + \ldots + p_{L-1}) = 0
\end{aligned}$$

In matrix notation, these equations can be represented as

$$\begin{pmatrix} 2 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 2 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & 2 & 1 & \cdots & 1 \\ 1 & 1 & 1 & 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & 1 & \cdots & 2 \end{pmatrix} \begin{pmatrix} p_2 \\ p_3 \\ p_4 \\ p_5 \\ \vdots \\ p_L \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{5.18}$$

Solving this linear system, we have,

$$p_k - p_l = 0, \forall k, l \ k \neq l \tag{5.19}$$

Thus,

$$p_k = \frac{1}{L}, \forall k. \tag{5.20}$$

Thus, when the channel distributions are uniform, $\mathbb{E}[M(t)]$ has the minimum value. □

We now prove the second part of the theorem. We show that when channels are uniform, $\mathbb{E}[M(t)]$ is a decreasing function of L.

Lemma 5.5. $\mathbb{E}[M(t)]$ *is a decreasing function of* L.

Proof. From (5.15),

$$\mathbb{E}[M(t)] = \left[\frac{1}{2} + \frac{1}{2L}\right](N-1) \tag{5.21}$$

Taking the derivative of $\mathbb{E}[M(t)]$ with respect to L yields that,

$$\frac{d\mathbb{E}[M(t)]}{dL} = \left[\frac{-1}{2L^2}\right](N-1) < 0. \tag{5.22}$$

Thus, $\mathbb{E}[M(t)]$ is a decreasing function of L. $\square$

Now, it is easy to see that in (5.15), taking $L \to \infty$ yields

$$\lim_{L\to\infty} \mathbb{E}[M(t)] = \lim_{L\to\infty} \left[\frac{1}{2} + \frac{1}{2L}\right](N-1) = \frac{N-1}{2}. \tag{5.23}$$

In the limiting case, when $N > 3$, $\mathbb{E}[M(t)] > 1$. In addition, according to Lemma 5.5 if L is finite, $\mathbb{E}[M(t)]$ is still greater than 1 whenever $N > 3$ since $\mathbb{E}[M(t)]$ decreases as L increases. We can conclude that when all channels are uniformly distributed, and when $N > 3$, then $\mathbb{E}[M(t)] > 1$. As a result, we guarantee to expand the rate region, i.e., $\epsilon > 0$. In addition, according to Lemma 5.3 and Lemma 5.4, $\mathbb{E}[M(t)]$ has its minimum value when channels are uniform. Therefore, for homogenous channels, when $N > 3$, then $\epsilon > 0$ and rate region is expanded. This completes the proof. $\square$

Note that the exact value of ϵ for homogeneous channels can be calculated by Theorem 5.2 and Lemma 5.1. Also note that Theorem 5.4 holds if $\frac{1}{\beta} > 3$.

5.3.3 Performance of SSF with Heterogenous channel

Next, we consider the case when user channels are not identical, i.e., $p_n^k \neq p_m^k$, for all $n \neq m$. First, in the following Lemma, we give a lower bound for $\mathbb{E}[M(t)]$.

Lemma 5.6. *When channels are heterogenous, $\mathbb{E}[M(t)]$ is lower bounded as follows:*

$$\mathbb{E}[M(t)] \geq (N-1)\left(p^{min} + (p^{min})^2\left[\frac{L(L-1)}{2}\right]\right) \tag{5.24}$$

Proof. Now, we consider the case that the channels are not identical, i.e., $p_n^k \neq p_m^k$, where $n \neq m$ and $\forall k$. The main difference between homogenous and heterogenous cases is that (5.12) does not hold for heterogenous channels. Let $\chi_n(t)$ be the event that user n is the user with maximum queue size at time t. Also, let $\varphi_k(t)$ be the event that the user with maximum queue size has channel state k at time t, i.e., $R_{i^*}(t) = r_k$. The probabilities of these events are denoted by $\Pr[\chi_n(t)]$ and $\Pr[\varphi_k(t)]$, respectively. Then $\mathbb{E}[M(t)]$ can be found as follows:

$$\begin{aligned}
\mathbb{E}[M(t)] &= \sum_{n=1}^{N}\sum_{k=1}^{L}\mathbb{E}[M(t)|\chi_n(t),\varphi_k(t)]\Pr[\chi_n(t),\varphi_k(t)] \\
&= \Pr[\chi_1(t)]\left(p_1^1(N-1) + p_1^2\sum_{n=2}^{N}\sum_{k=2}^{L}p_n^k + \ldots + p_1^L\sum_{n=2}^{N}p_n^L\right) \\
&+ \Pr[\chi_2(t)]\left(p_2^1(N-1) + p_2^2\sum_{\substack{n=1\\n\neq 2}}^{N}\sum_{k=2}^{L}p_n^k + \ldots + p_2^L\sum_{\substack{n=1\\n\neq 2}}^{N}p_{nL}\right) \\
&\vdots \\
&+ \Pr[\chi_N(t)]\left(p_N^1(N-1) + p_N^2\sum_{\substack{n=1\\n\neq N}}^{N}\sum_{k=2}^{L}p_n^k + \ldots + p_N^L\sum_{\substack{n=1\\n\neq N}}^{N}p_n^L\right).
\end{aligned}$$

Note that $p_n^k \geq p^{min}$ for all n, k. Hence, a lower bound on $\mathbb{E}[M(t)]$ can be given as follow,

$$\begin{aligned}
\mathbb{E}[M(t)] \geq [&p^{min}(N-1) + p^{min}(p^{min}(L-1)(N-1) \\
&+ p^{min}(p^{min}(L-2)(N-1) + \ldots \\
&+ p^{min}(p^{min}(N-1)]
\end{aligned}$$

By rearranging, we have,

$$\mathbb{E}[M(t)] \geq (N-1)\left(p^{min} + (p^{min})^2\left[\frac{L(L-1)}{2}\right]\right)$$

This completes the proof. $\square$

The following theorem gives our main result for heterogeneous channels.

Theorem 5.5. *For heterogeneous channels SSF algorithm guarantees to achieve a larger rate region than Λ_f, i.e., $\epsilon > 0$, if the number of users satisfies*

$$N > \frac{1}{\left(p^{min} + (p^{min})^2\left[\frac{L(L-1)}{2}\right]\right)} + 1, \tag{5.25}$$

Proof. Recall that in order to achieve larger rate region, $\mathbb{E}[M(t)]$ should be larger than 1. Lemma 5.6 gives a lower bound for $\mathbb{E}[M(t)]$. Hence, if the right hand side of inequality (5.24) is greater than 1, then $\mathbb{E}[M(t)] > 1$ and SSF guarantees to achieve larger rate region. $\square$

In an HDR system, there are 11 SNR levels which correspond to 11 different transmission rates, i.e., $L = 11$ [4]. In addition, as given in [4], almost all data rates occur with a probability higher than $p^{min} = 0.05$ in a typical embedded sector. Hence, for a typical HDR system, the number of users in the system should be at least 7, in order to reap the benefits of SSF algorithm. If the number of users in the system is lower than 7, then it is optimal to acquire channel state information from all users in the system.

Corollary 5.3. *For heterogenous channels, ϵ is given by,*

$$\epsilon \geq \frac{\beta\left((N-1)\left(p^{min} + (p^{min})^2\left[\frac{L(L-1)}{2}\right]\right) - 1\right)}{1 - \beta N} \tag{5.26}$$

5.3.4 Implementation Issues

Here, we mention some implementation issues arising when SSF algorithm is applied. In practice, CSI is sent back to the base station at a basic rate specified by the system so that this information can be decoded at the base station without any error. In HDR system, the minimum supportable rate is 38.4kb/s, and a single CSI is coded by using 4 bits. Hence, the required time for the transmission of a single CSI is equal to 0.1 millisecond (ms). In HDR system the duration of time slot is 1.67 ms, so 6.25% fraction of the time slot is consumed for each CSI acquisition. Hence, for a typical HDR system $\beta = 0.0625$.

Note that when all users send feedback, the base station can inform each user about its specific time slot to send feedback in the beginning and it does not change after that. For instance, at every time slot the base station first acquires CSI from user 1, then from user 2 and so on. However, with SSF a user will not know the other users who will report their channel information, and hence it cannot predict its order in sending feedback. In dynamic CSI feedback, there should be a mechanism to control the users' feedback order which resolves possible collisions during CSI transmissions on the uplink channel. Developing a mechanism to acquire CSI from a limited number of users remains as a research challenge. Note that any mechanism that is proposed to solve this problem brings some additional timing cost. Here, we propose a simple heuristic to prevent collision during the transmission of CSI: the base station always starts collecting CSI with a specific user as in the static case. If a user has a data rate greater than that of user i^*, it first announces it to the base station by sending 1 bit signal. Then, the base station waits until the user transmits its CSI, and then switches to the next user. Otherwise, it sends 0 bit signal, and the base station switches to the next user. This mechanism effectively controls the feedback order and prevents collisions but it brings some additional timing cost due to transmission of additional one bit length signals. Let us denote κ as the fraction of time slot consumed for all probing processes within SSF algorithm including acquisition of CSI, broadcasting ID, switching, etc. We next determine the effect of timing cost on the rate region of SSF algorithm.

Corollary 5.4. *When in total a fraction $(N - M(t) + \kappa)\beta$ of a time slot is consumed where $\kappa \geq 1$ then SSF algorithm guarantees to achieve a larger rate region than Λ_f,* *i.e.,* $\epsilon > 0$*, if the number of users satisfies*

$$N > \frac{\kappa}{\frac{1}{2} + \frac{1}{2L}} + 1, \tag{5.27}$$

Proof. Note that $\mathbb{E}[M(t)]$ depends only on the channel statistics and hence additional probing cost does not effect $\mathbb{E}[M(t)]$. Thus, Lemma 5.2 is still valid and, $\mathbb{E}[M(t)] = [\frac{1}{2} + \frac{1}{2L}](N-1)$. By following the same lines of the proof of Theorem 5.2, one can determine that

$$\epsilon = \frac{\beta \left(\mathbb{E}[M(t)] - \kappa\right)}{1 - \beta N}. \tag{5.28}$$

Clearly, when $\mathbb{E}[M(t)] > \kappa$ then, $\epsilon > 0$. Thus,

$$\mathbb{E}[M(t)] = [\frac{1}{2} + \frac{1}{2L}](N-1) > \kappa,$$

and $N > \frac{\kappa}{\frac{1}{2}+\frac{1}{2L}} + 1$. □

Hence, there must exist at least $\left\lceil \frac{\kappa}{\frac{1}{2}+\frac{1}{2L}} + 1 \right\rceil$ users to achieve a larger rate region compared to Max-Weight algorithm with complete CSI, where $\lceil x \rceil$ is the smallest integer that is greater than x. In other words, when additional probing cost is considered in terms of β the minimum number of users that is required to achieve larger rate region increases.

5.4 Numerical Results

In our simulations, we model a single cell CDMA downlink transmission utilizing high data rate (HDR) [4]. The base station serves 15 users and keeps a separate queue for each user. Time is slotted with length $T = 2$ ms. Packets arrive at each slot according to Poisson distribution for each users with mean λ_n. The size of a packet is set to 128 bytes which corresponds to the size of an HDR packet. Each channel has 11 possible states with rates as given in [4].

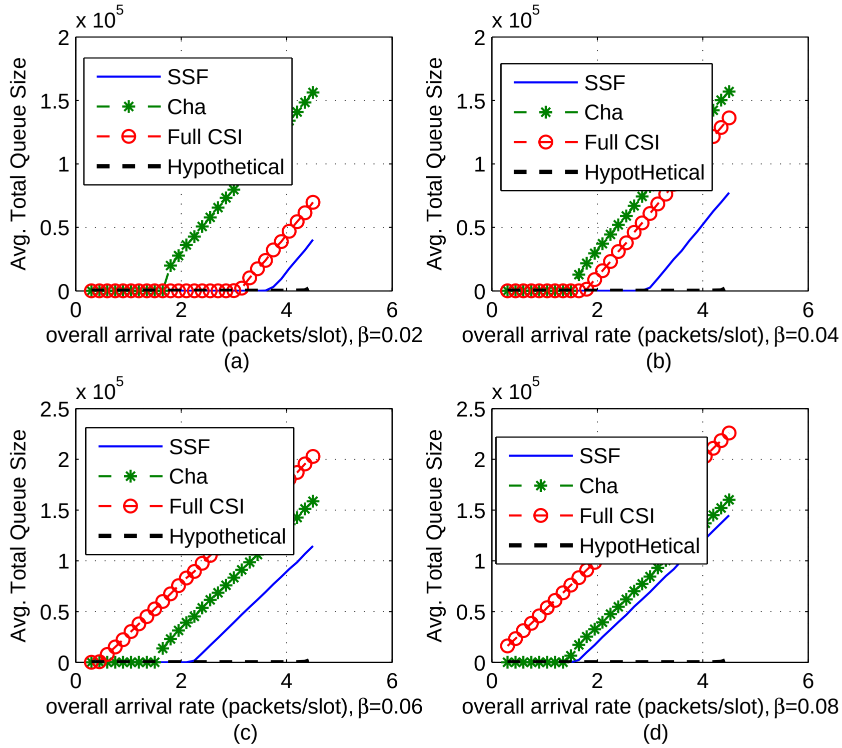

Figure 5.2: Performance of SSF algorithm with Homogenous and Uniform channels.

We compare SSF algorithm with the algorithm in [83], which we call Cha. Specifically, for homogenous channels, an optimal acquisition strategy is to receive CSI from users in the decreasing order of their queue lengths as shown in Corollary 1 in [83]. However, for heterogenous channels, an optimal strategy is not characterized. Hence, we compare Cha and SSF only when channels are homogenous.

5.4.1 Homogenous Channels

First, we evaluate the performance of SSF algorithm when channels are homogenous and uniformly distributed.

Uniform Channels

In uniform case, $p_n = 1/11$ for all users since there are 11 channel states and the channel state distributions are identical. Note that $\beta < \frac{1}{N} = 0.067$, and we choose different values of β that are close and far away from this value. Figure 5.2 depicts

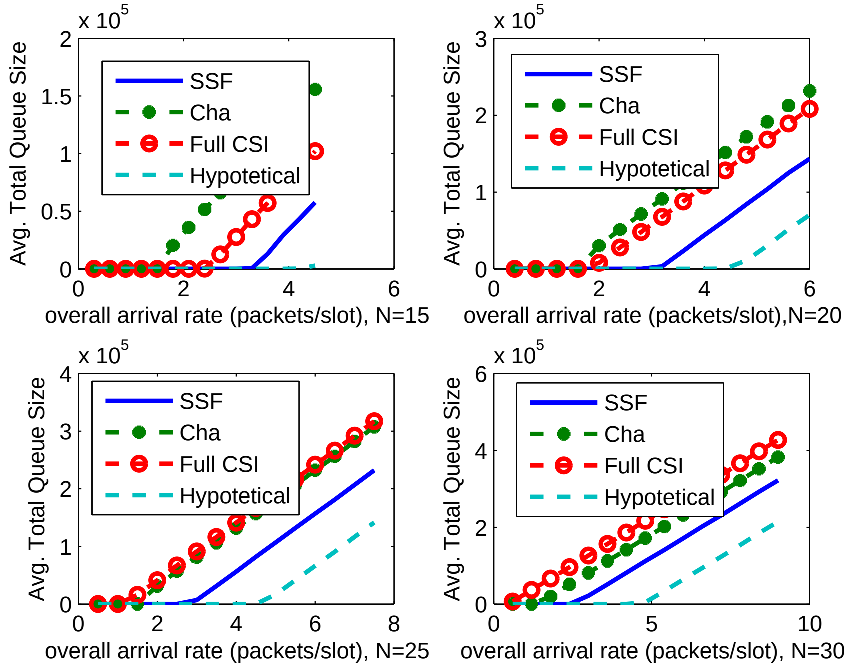

Figure 5.3: Performance of SSF and Cha vs. N.

the average total queue sizes in terms of packets vs. the overall arrival rate when $\beta = 0.02$, $\beta = 0.04$, $\beta = 0.06$ and $\beta = 0.08$. The maximum supportable arrival rate is achieved by a *hypothetical* Max-Weight algorithm where the feedback cost is zero, i.e., $\beta = 0$. Note that for $\beta = 0.02$ the lowest supportable rate is achieved with Cha, and it is achieved by Max-Weight acquiring channel states from all users for $\beta = 0.08$. The reason that Max-Weight is worse for larger values of β is that the decrease in throughput due to feedback costs becomes more prominent. SSF algorithm achieves a very similar performance as compared to the hypothetical Max-Weight algorithm for $\beta = 0.02$, and it outperforms both of the other algorithms for all values of β. The main reason that SSF outperforms Cha is *not* because it acquires feedback from fewer average number of users. On the contrary, in our numerical simulations, we observe that SSF has $\mathbb{E}[M(t)] = 7.63$, whereas Cha has a larger value of $\mathbb{E}[M(t)]$. It is because SSF *always* schedules the user with maximum weighted rate at every slot, while Cha does not. Note that the performance of Cha becomes similar to that of SSF as the condition on β violates, and it may be better than SSF for large values of β. However, Figure 5.2 shows that for consistence values of β SSF outperforms Cha.

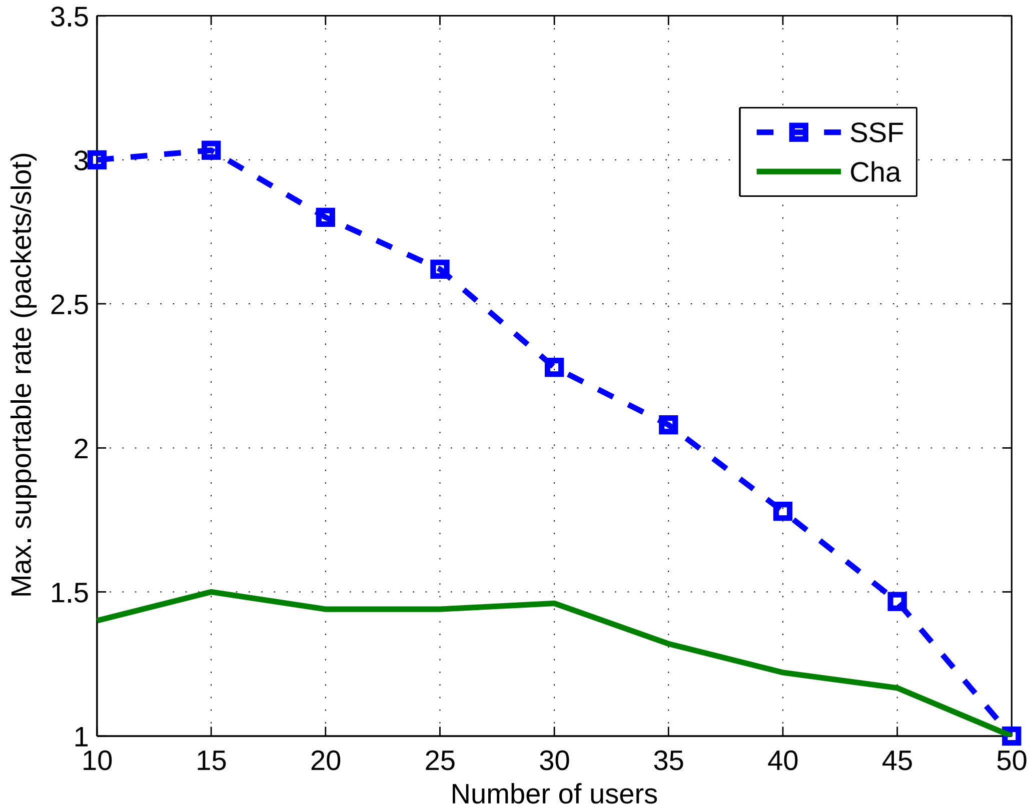

Figure 5.4: Maximum supportable rate vs. number of users.

By using (5.16) and setting $N = 15$, $L = 11$ and $\beta = 0.02$, one can find that $\epsilon = 0.19$. That result states that we can support an arrival rate which is 19% higher than that of Max-Weight algorithm with full CSI. From Figure 5.2, the maximum supportable rate by Max-Weight with full CSI is around 3 packets/slot whereas SSF can support up to 3.57 packets/slot, which is 19% higher than that of Max-Weight algorithm with full CSI, which confirms our theoretical result.

We have also evaluated the performance of SSF algorithm with respect to the number of users in the network. We set $\beta = 0.03$ so that the condition on β is satisfied for all values of N. Figure 5.3 depicts total average queue sizes in terms of packets vs. the overall arrival rate when $N = 15$, $N = 20$, $N = 25$ and $N = 30$. For all values of N, SSF can support arrival rates higher than that can be supported by Cha scheme. We have performed additional numerical studies to understand the performances of Cha and SSF as the number of users in the network increases. We choose $\beta = 0.03$,

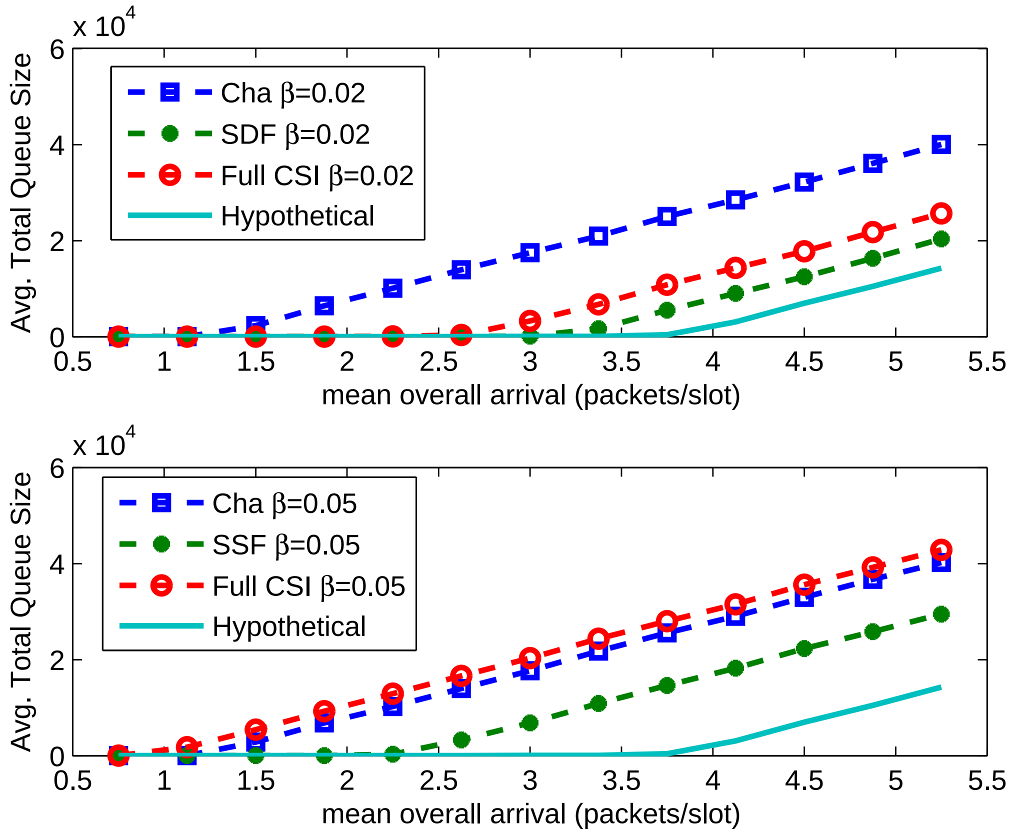

Figure 5.5: Performance of SSF algorithm with Non-Uniform channels.

and vary N between 10 and 50 users. Note that for large values of N, there may not be sufficient time for data transmission since the entire slot is used for probing. In such a case, data transmission rate equals to zero. Figure 5.4 depicts the maximum supportable arrival rate vs. the number of users in the network. As N increases, the maximum supportable rate of both SSF and Cha decreases since the remaining time for data transmission decreases as well. However, SSF outperforms Cha for all values of N.

Non-Uniform Channels

Here, we investigate the performance of SSF algorithm when channels are identical but the channel state distributions are not uniform. In this case, the channel state distribution p_l, $l = 1, \ldots, 11$ is given as $\{0.01, 0.01, 0.03, 0.08, 0.15, 0.24, 0.18, 0.09, 0.12, 0.05, 0.04\}$.

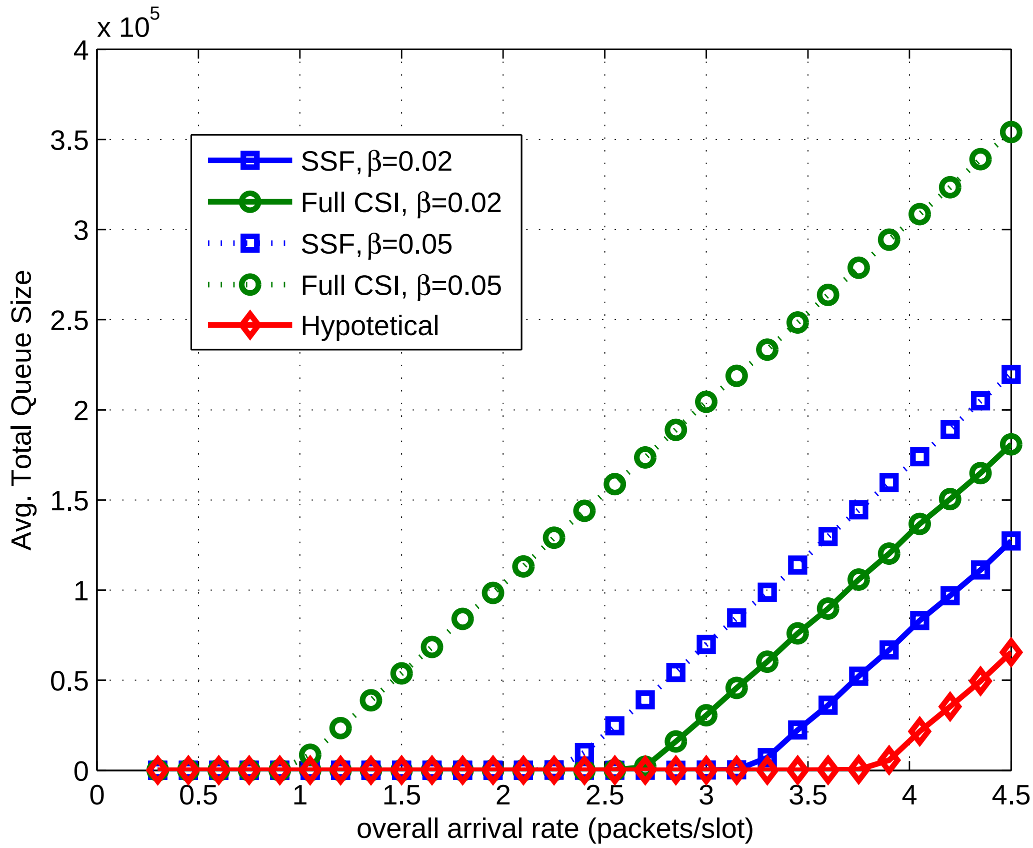

Figure 5.6: Performance of SSF algorithm with Heterogenous channels.

Figure 5.5 depicts the average total queue sizes in terms of packets vs. the overall arrival rate when $\beta = 0.02$ and $\beta = 0.05$. The maximum supportable arrival rate is achieved with hypothetical algorithm, and the lowest supportable rate is achieved by Max-Weight algorithm acquiring channel states from all users. We can again observe that SSF outperforms Cha when the channels have non-uniform distribution.

5.4.2 Heterogenous Channels

For the case of heterogeneous channels, we divide the users into three groups where there are five users in each group. The channel state distributions of each group are given as follows: for the first group, $n = \{1, 2, \dots, 5\}$, p_l^n, $l = \{1, \dots, 11\}$ is given as $\{0.01, 0.01, 0.03, 0.08, 0.15, 0.24, 0.18, 0.09, 0.12, 0.05, 0.04\}$, for the second group, $n = \{6, 7, \dots, 10\}$, $p_l^n = \{0.01, 0.02, 0.02, 0.07, 0.2, 0.24, 0.15, 0.05, 0.12, 0.08, 0.04\}$ and for the last group, $n = \{11, 12, \ldots, 15\}$, $p_l^n = \{0.02, 0.01, 0.02, 0.06, 0.21, 0.25, 0.18, 0.05, 0.11, 0.06, 0.03\}$.

Figure 5.6 depicts the average total queue sizes in terms of packets vs. the overall arrival rate for $\beta = 0.02$ and $\beta = 0.05$. The maximum supportable arrival rate is still achieved by hypothetical algorithm and the lowest supportable rate is achieved by Max-Weight algorithm acquiring channel states from all users. when $\beta = 0.05$ the supportable rate achieved by SSF algorithm is 2.25 packets/second whereas it is approximately 1.12 packets/second with full CSI. Hence, SSF achieves larger rate region than Max-Weight with full CSI has.

5.5 Chapter Summary

We have developed joint scheduling channel feedback algorithm in a single channel wireless downlink network. We have assumed that acquiring CSI of a user consumes a certain fraction of data slot, hence, decreases the achievable throughput. The set of channels is determined by considering the channel gain of the user with maximum queue size. We have shown that the proposed joint algorithm can support $1+\epsilon$ fraction of full rate region achieved when all CSIs are available at the scheduler. Then, we have proved the sufficient condition for $\epsilon > 0$ for both homogenous and heterogenous channels by determining the expected number of reported CSI within our algorithm. In simulation results, we show that by applying our algorithm the base station can stabilize the network and achieves larger rate region with collecting CSI from less than the half of users instead of full CSI.

Chapter 6

Entropy-based Active Learning for Scheduling

In Chapter 5, we have investigated Max-Weight algorithm without complete CSI at the base station. Our main assumptions in that Chapter are i) channels vary independently over time slots; ii) the base station can obtain the complete CSI by sacrificing throughput. In this Chapter, we go further and remove these assumptions by considering more practical channel and probing models. Specifically, in this Chapter, we address the joint design of scheduling and channel probing under general channel processes. Our method predicts the instantaneous channel rates, and calculates the uncertainty in the prediction to make a scheduling and probing decision. To explicitly quantify the uncertainty in the channel prediction that will be removed by channel probing we adopt entropy measure from information theory. In order to accurately predict instantaneous user channel states we employ a Bayesian approach and use Gaussian processes as a state-of-the-art regression technique. We analytically prove that our algorithm achieves a fraction ϵ of the full rate region when complete CSI is available. We demonstrate numerically under realistic assumptions that this rate region can be achieved by probing only less than 50% of all channels in a CDMA based cellular network utilizing high data rate protocol under practical channel conditions.

6.1 Overview

Due to their flexible and low-cost infrastructure, wireless networks have the potential to provide high-speed, ubiquitous communication to mobile users. One of the most challenging problems that remains to be solved is to allocate limited and time-varying resources among multiple users meeting their Quality-of-Service (QoS) requirements. The problem is exacerbated by the highly dynamic nature of wireless communications due to multiple superimposed random effects such as mobility and multi-path fading. In some cases, acquiring extensive information on wireless system characteristics may be simply infeasible due to prohibitive costs or hard constraints. In others, the wireless channel may be so non-stationary that by the time the information is obtained, it is already outdated due to channels fast-changing nature. Hence, in general, resource allocation decision should be made based on limited and outdated information.

In this Chapter, we develop a joint partial CSI acquisition and scheduling algorithm without requiring any a priori assumption on their channel state distributions. Our algorithm tracks the channel states according to a learning algorithm and by judiciously probing only those user channels for which there is a high level of uncertainity about their states. At each slot, a user among the set of probed users is scheduled which has the maximum queue length and rate product. Our algorithm is based on a recent machine learning and optimization framework developed in [85], wherein the exploration and exploitation trade-off is explicitly quantified as a multi-objective meta optimization problem. In our setting, trade-off is between scheduling of the user with the highest weighted rate (exploitation) and probing of the channels with outdated channel observations (exploration). The solution of this optimization problem requires the prediction of the instantaneous channel states, and the measurement of the level of uncertainty in this prediction. Instantaneous channel states are predicted by adopting a Bayesian approach, and using Gaussian processes as the state-of-the-art regression method. Gaussian process regression is a powerful nonlinear interpolation tool, where the inference of continuous values are made with respect to a Gaussian process prior [86]. We note that although the inference of instantaneous channel gains is with a Gaussian process prior, this does not assume that underlying channel model is Gaussian. In fact, as demonstrated by our numerical experiments, our approach is applicable to a wide range of channel models including time-correlated and even non-stationary channels. Another unique feature of our algorithm is that the uncertainty in the predicted channel state is quantified explicitly by the *entropy measure* from the information theory thanks to the Gaussian process regression method used in the prediction. Hence, our algorithm weighs the level of uncertainty eliminated by probing a channel against the aspiration to schedule the user with the maximum weight to determine a set of users probed at every slot.

Our contributions are summarized as follows:

- We define a class of Max-Weight-like policies which make scheduling decisions based on the estimated values of instantaneous channel rates rather than their exact values, and characterize the achievable rate region of these policies as compared to Max-Weight policy with exact channel state information.

- We quantify the information obtained at every probing of a channel with Shannon's entropy formula based on the past observations of the channel.

- We develop a Gaussian Process Regression based channel rate prediction algorithm, which does not require any a priori assumption on the channel distributions. The algorithm essentially learns the channel characteristics in an active learning framework, where the channel observations arrive sporadically over time.

- We develop a joint probing and scheduling algorithm, which selects the set of users to be probed, and a user in the same set scheduled for transmission. The selection is based on the solution of a multi-objective optimization problem aiming to balance the information obtained by each channel probe with the selection of the user with the maximum weight at each slot.

- We evaluate our algorithm in a realistic network setting, where users communicate with the base station using High Data Rate (HDR) protocol [4]. We compare the performance of our algorithm with that of the state-of-the-art channel prediction method based on Autoregression (AR) [87].

The inter-play between the channel probing/estimation with the achievable network throughput has been recognized early in the research community, and a plethora of work has appeared investigating the underlying trade-off.

Saturated Systems: The authors in [48], [49], [50], [51] have studied the problem of joint scheduling and feedback by assuming infinitely backlogged user queues. In [48] users contend for the feedback channel if the channel state exceeds a pre-defined threshold. Similarly, in [50], multiple threshold levels are used to reduce the cost for obtaining the CSI. For uplink scheduling, the authors in [49] propose an optimization framework in OFDM systems. A random access based feedback protocol for achieving multiuser diversity with limited feedback was proposed in [51]. We refer the readers to [53] for a summary of different techniques used to reduce the overhead of obtaining CSI such as quantization of CSI, beamforming or precoding.

Unsaturated Systems: Recent body of work has investigated the problem of stabilizing the network of queues, when these queues have arbitrary arrival rates. In [73], the authors use the expected transmission rates weighed by the queue lengths to select a fixed number of users to be probed in each slot. After probing, a user within this subset of probed users is selected according to Max-Weight rule. In case of a multi-channel downlink system, the limited feedback resource is allocated at the beginning of each time slot according to the queue-lengths and the channel statistics in [75]. In [83], the problem of finding optimal joint scheduling and probing is transformed into an optimal stopping time problem. The resulting algorithm is found as the solution to a dynamic program defined over the underlying Markov Decision Process (MDP). All three aforementioned algorithms achieve throughput-optimality under the assumption that channel distributions are independent and identically distributed (iid), and they are known a priori.

In contrast, [77], [78], [79], and [88] presented joint scheduling and probing algorithms without requiring a priori channel statistics. In [77] and [78], the authors propose algorithms that estimate the channel statistics under the assumption that user channels have iid probability distribution. The problem of joint prediction of channel states and scheduling to optimize a long term metric under stability and other resource constraints was studied in [77]. Similarly, the authors in [78] probabilistically determine at every slot whether to explore a user channel state or exploit the time to transmit data. Based on the past observations, the user channel statistics is estimated, and it is shown that the resulting random scheduling algorithm is throughput optimal. In [79], the authors propose a joint scheduling and channel estimation algorithm for correlated ON/OFF Markovian channels.

Unlike these works, our algorithm does not aim to estimate the channel statistics, and thus, it is more suitable for realistic non-stationary channel models [81]. In [88], we develop an algorithm which probes only those users with sufficiently good channel quality and schedules the user with the maximum weight at each transmission opportunity. The underlying system model considered in [88] is completely different than the one used in this work, since in [88] feedback from as many users as needed can be obtained by tolerating a reduction in data transmission rates. Unlike all aforementioned approaches, our proposed approach assumes neither stationarity nor a particular distribution for channels, and hence, it has more general applicability. In our preliminary work [89], we apply a similar algorithm assuming the same model in [88] without giving any analytical performance guarantees in terms of rate region.

Learning: The advantages of using learning algorithms in wireless communication systems where the complete channel/network state information is not available has long been recognized. For example, learning algorithms are used in adaptive transmission techniques, such as adaptive modulation and channel coding to predict future channel states. Studies have shown that such learning based adaptive transmission techniques provide more efficient spectrum utilization [90]. Learning algorithms have also been applied to various problems in communication networks where there is limited information on network states such as routing [91], spectrum allocation [92], interference mitigation [93], multi-channel cognitive networks [94], combinatorial network optimization [95], multi-channel access [96] and localization [97]. These problems are solved via reinforcement learning [91], [92], Q-learning [93], multi-armed bandit [94], [96], [95] and

Gaussian Process Regression [97] techniques.

Note that by assuming the stationarity of the underlying stochastic processes (i.e., iid or Markovian channels), the focus in [91]- [96] is on finding a single, stationary solution which is once found, no longer need to be changed. However, the problems in stationary domains may be too large to be solved exactly. For instance, extending the analysis of the aforementioned works to correlated Rayleigh fading is in general difficult. Moreover, focusing only on the stationary solution is not adequate when dealing with a non-stationary channel process. If the underlying process changes, then all prior predictions become outdated. Thus, in this case the only option is to track the process. The need for tracking nonstationary environments via learning, although widely acknowledged, has not been extensively pursued in research community. In this work we focus on continuous tracking of the channel process rather than finding a single stationary solution which may not be sufficiently good even for stationary problems [98].

6.2 System Model

We consider a multiuser downlink network with N users and a single base station (BS). We assume that time is slotted, and we adopt a non-interference model, where only one user is transmitting at any given time slot. Each user channel experiences independent block fading, in which the channel gain is constant over the duration of a time slot, and it is varying *continuously* from slot to slot. The gain of the channel between the BS and node n, $n \in \{1, 2, \ldots, \mathcal{N}\}$ at time t is denoted by $c_n(t)$, and its value is determined according to an arbitrary probability distribution[1]. Similarly, we denote the instantaneous channel rate between the BS and node n by $R_n(t)$, which is defined as the mutual information between the output symbols of base station and the input symbols at node n over slot t. The maximum value of $R_n(t)$, is obtained when the input

[1]As described in the subsequent section, our algorithm which relies on Gaussian Process Regression (GPR), does not make any assumption on the channel distribution. GPR is a generic function estimation method that performs well in approximating any smooth and continuous nonlinear function [86].

symbols are chosen from a Gaussian-distributed input alphabet,

$$R_n(t) \leq BW \log_2 \left(1 + P \left|c_n(t)\right|^2\right) \text{ bits}, \tag{6.1}$$

where BW is the bandwidth of the channel, and P is the noise normalized transmit power. Note that we assume that both BW and P are exogenous processes over which we have no control, i.e., we do not consider bandwidth or power control.

The base station does not have the knowledge of the channel states of the receivers at the beginning of the slot, but it has to acquire this information. At the beginning of each time slot, t, base station broadcasts a pilot signal with a fixed and known power. Each node n determines its CSI $c_n(t)$ by measuring the received SNR. In this model, we assume that the base station receives feedback from at most $L < N$ users over a dedicated and fixed bandwidth uplink channel. We assume that L is fixed and remains constant throughout the system operation. Such a feedback channel model closely represents practical systems such as HDR and LTE [4]. Let $\mathcal{S}(t)$ be the set of users for which the channel state information is acquired at time slot t. Note that the cardinality of set $\mathcal{S}(t)$, is L, i.e., $|\mathcal{S}(t)| = L$, for all t.

The base station maintains a separate queue for each node n. Packets arrive according to a stationary arrival process that is independent across nodes and time slots. Let $A_n(t)$ be the amount of data arriving into the queue of node n at time slot t. Also let $\lambda_n = \mathbb{E}[A_n(t)]$ be the average arrival rate into the queue of node n. There is a departure from the queue of node n, whenever that node is selected for transmission. Let $\mathcal{J}_n(t)$ represent the scheduler decision, where $\mathcal{J}_n(t) = 1$ if user $n \in \mathcal{S}(t)$ is scheduled for transmission in slot t, and $\mathcal{J}_n(t) = 0$ otherwise. By definition, at most one user can be served at a time slot, i.e., $\sum_{n=1}^{N} \mathcal{J}_n(t) = 1$, for all t. The dynamics of queue length process of node n is given as follows:

$$Q_n(t+1) = [Q_n(t) + A_n(t) - R_n(t)\mathcal{J}_n(t)]^+, \tag{6.2}$$

where $[x]^+ = \max(x, 0)$. Let $\mathbf{Q}(t) = [Q_1(t), Q_2(t), \ldots, Q_N(t)]$ denote the vector of user queue lengths. We say that the network is stable if the mean length of user queues are all finite.

6.3 Scheduling Under Limited Channel State Information

We denote as π the joint scheduling and channel probing policy which selects the pair $(n, \mathcal{S}(t))$ at every slot t, where $n \in \mathcal{S}(t)$ is the scheduled user, and $\mathcal{S}(t)$ is the set of probed users. Given $\pi = (n, \mathcal{S}(t))$, n is determined according to Max-Weight rule, i.e.,

$$n = \underset{i \in \mathcal{S}(t)}{\operatorname{argmax}} \; Q_i(t) R_i(t). \tag{6.3}$$

Let $\mathcal{F}$ be the set of all feasible policies and $\pi \in \mathcal{F}$.

The convex hull of the set of arrival rate vectors $\Lambda = (\lambda_1, \ldots, \lambda_N)$ for which there exists an appropriate scheduling policy that stabilizes the network is called *achievable rate region*. When the exact channel information for all users is known, i.e., $L = N$, the achievable rate region is the largest. Let Λ_h denote this *hypothetical rate region*, the boundary of which can never be achieved in real systems [84]. It was shown that Max-Weight algorithm with full CSI stabilizes the network and achieves Λ_h [6]. In practice, the channel state information from only a subset of users can be obtained, i.e., $L < N$. Under limited channel state information, there is a possibility that the user with the maximum weighted rate cannot be selected, which in turn may reduce the achievable rate region. Our aim is to find a joint scheduling and channel probing policy that stabilizes the network for a given set of arrival rates by judiciously determining a subset of channels probed, $\mathcal{S}(t)$, and by scheduling a user from this subset at every time slot. Given the queue state $\boldsymbol{Q}(t)$, we consider the following optimization problem at each slot:

$$\underset{(n, \mathcal{S}(t)) \in \mathcal{F} : |\mathcal{S}(t)| = L}{\operatorname{argmax}} \; \mathbb{E}\left[\sum_{n \in \mathcal{S}(t)} Q_n(t) R_n(t) \mathcal{J}_n(t) \mid \mathbf{Q}(t) \right] \tag{6.4}$$

In [73] and [83], the authors proposed throughput-optimal algorithms solving (6.4) when channel distribution is known. In this work, we extend these works to address the case when the channel distributions are unavailable and/or time-varying.

In practice, it is not possible to accurately determine the exact channel distributions a priori to system operation. Hence, we consider a joint policy $\pi(\eta)$ which employs an arbitrary channel tracking algorithm η to predict channel states at each slot. We consider a class of policies where $\mathcal{S}(t)$ and n are determined as follows: Based on the predicted channel states by tracking algorithm η, L users with the highest *estimated* transmission rate and queue length product are added to $\mathcal{S}(t)$. After acquiring CSI from users in $\mathcal{S}(t)$, a user in $\mathcal{S}(t)$ is scheduled according to (6.3). Let $\hat{c}_n^{\pi(\eta)}(t)$ denote the estimated CSI of user n at the beginning of time t under policy $\pi(\eta)$. Also let $\hat{R}_n^{\pi(\eta)}(t)$ denote the estimated transmission rate of user n at time t which is defined according to (6.1) by replacing the $c_n(t)$ by $\hat{c}_n^{\pi(\eta)}(t)$. The estimation error is defined as $e_n^{\pi(\eta)}(t) = \left|\hat{R}_n^{\pi(\eta)}(t) - R_n(t)\right|$. We assume that $R_{min} < \hat{R}_n^{\pi(\eta)}(t) < R_{max}$, and $e_n^{\pi(\eta)}(t) < e_{max}$ for all n, t.

We first analyze the performance of a general joint scheduling and probing policy $\pi(\eta) \in \mathcal{F}$. Note that depending on the quality of employed estimation method, and the choice of users probed at each slot, policy $\pi(\eta)$ may or may not schedule the user with the actual highest weight at each slot. Given the backlog process, $\mathbf{Q}(t)$, we need to determine how often a policy $\pi(\eta)$ chooses the user with the *actual* maximum weighted rate. Let $\rho^{\pi(\eta)}(\mathbf{Q}(t))$ be this probability which is defined as:

$$\begin{aligned}\rho^{\pi(\eta)}(\mathbf{Q}(t)) = \Pr \Bigg[&\operatorname*{argmax}_n Q_n(t) R_n(t) = k \Bigg| \\ &\operatorname*{argmax}_n Q_n(t) \hat{R}_n^{\pi(\eta)}(t) = k, \mathbf{Q}(t) \Bigg].\end{aligned}$$

The following theorem characterizes the achievable rate region of policy $\pi(\eta) \in \mathcal{F}$, i.e., $\Lambda^{\pi(\eta)}$, as compared to that of Max-Weight algorithm, Λ_h.

Theorem 6.1. *For some given $0 < \epsilon < 1$, a fraction ϵ of the rate region, $\Lambda^{\pi(\eta)} \subseteq \ \epsilon \cdot \Lambda_h$*

can be achieved if

$$\rho^{\pi(\eta)}(\mathbf{Q}(t)) \geq \epsilon$$

for all $\mathbf{Q}(t)$*.*

Proof. We consider the worst case in which at most one user is probed at every slot, i.e., $L = 1$. In this case, the probed user is always the scheduled user. Note that when $L > 1$, we may achieve a larger rate region. We also drop η in $\pi(\eta)$ for notational simplicity. Let $\mathcal{J}^f(t) = 1$ if user n is scheduled when full CSI available, otherwise $\mathcal{J}^f(t) = 0$. Similarly, $\mathcal{J}_n^\pi(t) = 1$ if user n is scheduled with policy π. Otherwise, $\mathcal{J}_n^\pi(t) = 0$.

Consider also the following functions :

$$g_f(\mathbf{Q}(t)) = \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) R_n(t) \mathcal{J}_n^f(t) | \mathbf{Q}(t)\right], \tag{6.5}$$

$$g_\pi(\mathbf{Q}(t)) = \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) R_n(t) \mathcal{J}_n^\pi(t) | \mathbf{Q}(t)\right]. \tag{6.6}$$

The function (6.5) gives the expected weighted-sum rate according to Max-Weight algorithm with full CSI, whereas (6.6) is the expected-sum rate when at most $L = 1$ channel is probed with policy π. Our aim is to determine $\mathcal{J}_n^\pi(t)$ at every time slot so that $g_f(\mathbf{Q}(t))$ is close to $g_\pi(\mathbf{Q}(t))$. We define the event χ such that χ occurs if policy π and full CSI Max-Weight algorithm schedule the same user at a time slot, i.e.,

$$\operatorname*{argmax}_n Q_n(t) R_n(t) = \operatorname*{argmax}_n Q_n(t) \hat{R}_n^\pi(t).$$

We denote the probability of event χ as $\rho^\pi(\mathbf{Q}(t))$. We use the following theorem given in [40] to prove our main result Theorem 6.1.

Theorem 6.2. *[40] If for some* $\epsilon > 0$ *policy* π *guarantees*

$$g_\pi(\mathbf{Q}(t)) \geq \epsilon g_f(\mathbf{Q}(t)) \tag{6.7}$$

for all $\mathbf{Q}(t)$*, then policy* π *can achieve a fraction* ϵ *of hypothetical rate region,* Λ_h*.*

Note that $g_\pi(\mathbf{Q}(t))$ can be rewritten as follows,

$$\begin{aligned} g_\pi(\mathbf{Q}(t)) = \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)R_n(t)\mathcal{J}_n^\pi(t)|\mathbf{Q}(t),\chi\right]\rho^\pi(\mathbf{Q}(t)) \\ + \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)R_n(t)\mathcal{J}_n^\pi(t)|\mathbf{Q}(t),\chi'\right](1-\rho^\pi(\mathbf{Q}(t))). \end{aligned}$$

Note that when event χ occurs, the following equality is true,

$$g_f(\boldsymbol{Q}(t)) = \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)R_n(t)\mathcal{J}_n^\pi(t)|\mathbf{Q}(t),\chi\right].$$

Thus, we have

$$\begin{aligned} g_\pi(\mathbf{Q}(t)) = g_f(\mathbf{Q}(t))\rho^\pi(\mathbf{Q}(t)) \\ + \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)R_n(t)\mathcal{J}_n^\pi(t)|\mathbf{Q}(t),\chi'\right](1-\rho^\pi(\mathbf{Q}(t))). \end{aligned}$$

Note that,

$$\mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)R_n(t)\mathcal{J}_n^\pi(t)|\mathbf{Q}(t),\chi'\right](1-\rho^\pi(\mathbf{Q}(t))) \geq 0$$

Hence,

$$g_\pi(\mathbf{Q}(t)) \geq g_f(\mathbf{Q}(t))\rho^\pi(\mathbf{Q}(t)) \tag{6.8}$$

By dividing both sides of (6.8) by $g_f(\boldsymbol{Q}(t))$, we obtain,

$$\frac{g_\pi(\mathbf{Q}(t))}{g_f(\mathbf{Q}(t))} \geq \rho^\pi(\mathbf{Q}(t)) \tag{6.9}$$

Thus, if $\rho^\pi(\mathbf{Q}(t)) \geq \epsilon$, then $g_\pi(\mathbf{Q}(t))/g_f(\mathbf{Q}(t)) \geq \epsilon$. Hence, according to theorem [40], the scheduling policy with estimated channel rates can achieve as least ϵ fraction of

Λ_h. $\square$

We note that the largest value of ϵ that can be supported by an estimation policy depends on the channel statistics, and it cannot be obtained for the general case. However, to demonstrate the typical structure of ϵ, we consider a simple example where there are two users with identical channel distributions receiving service from a BS.

Example: The channel gain between the BS and each user is assumed to be iid Rayleigh fading channel with parameter μ. We assume that at most one user can be probed at each slot, i.e., $L = 1$.

Lemma 6.1. *Under high SNR assumption, ϵ is given by,*

$$\epsilon = \exp\left(-\frac{\mu}{P}\left[1 - e^{-e_{max}\left(1+\frac{R_{max}}{R_{min}}\right)}\right] e^{\frac{(R_{max})^2}{R_{min}}}\right)$$

Proof. Let transmission rates of users be defined as $R_1(t)$ and $R_2(t)$, and their queue sizes be defined as $Q_1(t)$, $Q_2(t)$, respectively. For analytical simplicity, we assume high SNR approximation, i.e., $R_n(t) \approx \ln\left(P\left|c_n(t)\right|^2\right)$ and we drop η in $\pi(\eta)$ and time index for notational simplicity. Then, we determine the probability that the same user is scheduled by policy π and full CSI Max-Weight algorithm. Formulary:

$$p^\pi = \Pr\left(Q_1R_1 \geq Q_2R_2 | Q_1\hat{R}_1^\pi \geq Q_2\hat{R}_2^\pi, Q_1, Q_2\right)$$

We assume the worst case scenario in which the estimation error is e_{max} and R_1 is overestimated whereas R_2 is underestimated. In this scenario we have,

$$Q_1\hat{R}_1^\pi \geq Q_2\hat{R}_2^\pi$$
$$Q_1(R_1 + e_{max}) \geq Q_2(R_2 - e_{max})$$
$$Q_1R_1 - Q_2R_2 \geq -e_{max}(Q_1 + Q_2)$$

Let us define $K \triangleq e_{max}(Q_1 + Q_2)$. Now, p^{π} can be rewritten as follows:

$$\begin{aligned}
p^{\pi} =& \Pr\left(Q_1R_1 \geq Q_2R_2 | Q_1R_1 - Q_2R_2 \geq -K, Q_1, Q_2\right)\\
=& \frac{\Pr\left(Q_1R_1 \geq Q_2R_2 \text{ and } Q_1R_1 - Q_2R_2 \geq -K | Q_1, Q_2\right)}{\Pr\left(Q_1R_1 - Q_2R_2 \geq -K | Q_1, Q_2\right)}\\
=& \frac{\Pr\left(Q_1R_1 - Q_2R_2 \geq 0 | Q_1, Q_2\right)}{\Pr\left(Q_1R_1 - Q_2R_2 \geq -K | Q_1, Q_2\right)}
\end{aligned}$$

We first calculate the following probability,

$$p^{\pi}_{num} = \Pr\left(Q_1R_1 - Q_2R_2 \geq 0 | Q_1, Q_2\right).$$

Since $|c_1(t)|^2$ and $|c_2(t)|^2$ are identically distributed exponential random variables with parameter μ, p^{π}_{num} is determined as follows:

$$\begin{aligned}
p^{\pi}_{num} &= \Pr\left(R_1 \geq \frac{Q_2R_2}{Q_1} \mid Q_1, Q_2\right)\\
&= \Pr\left(P\,|c_1|^2 \geq \frac{e^{aR_2}}{P} \mid Q_1, Q_2\right)\\
&= \int_0^{\infty} \Pr\left(|c_1|^2 \geq \frac{e^{ar_2}}{P} \;\mid\; Q_1, Q_2, R_2 = r_2\right) f_{R_2}(r_2) d_{r_2}
\end{aligned}$$

where $a = \frac{Q_2}{Q_1}$ and $f_{R_2}(r_2)$ is the pdf of R_2 which is given by,

$$f_{R_2}(r_2) = \frac{\mu}{P} e^{r_2} e^{-\frac{\mu}{P} e^{r_2}}, \text{ for } r_2 \geq 0.$$

Hence,

$$\begin{aligned}
&\Pr\left(|c_1|^2 \geq \frac{e^{ar_2}}{P} \;\mid\; Q_1, Q_2, R_2 = r_2\right)\\
&= \int_{\frac{e^{aR_2}}{P}}^{\infty} \mu e^{-\mu c_1} dc_1\\
&= e^{-\frac{\mu}{P} e^{ar_2}}
\end{aligned}$$

Then, p^{π}_{num} is given by,

$$p^{\pi}_{num} = \int_0^{\infty} e^{-\mu e^{\frac{ar_2}{P}}} f_{R_2}(r_2) d_{r_2}$$
$$= \int_0^{\infty} \frac{\mu}{P} e^{r_2} e^{-\frac{\mu}{P} e^{r_2}} e^{-\frac{\mu}{P} e^{ar_2}} d_{r_2}$$

Let $b = \frac{K}{Q_1}$. Next, we determine the following probability,

$$p^{\pi}_{denum} = \Pr\left(Q_1 R_1 - Q_2 R_2 \geq -K | Q_1, Q_2\right).$$

By following the same way, p^{π}_{denum} is given as,

$$p^{\pi}_{denum} = \int_0^{\infty} \frac{\mu}{P} e^{r_2} e^{-\frac{\mu}{P} e^{r_2}} e^{-\frac{\mu}{P} e^{ar_2 - b}} d_{r_2}$$

Hence, we have,

$$\begin{aligned} p^{\pi} &= \frac{p^{\pi}_{num}}{p^{\pi}_{denum}} \\ &= \frac{\int_0^{\infty} \frac{\mu}{P} e^{r_2} e^{-\frac{\mu}{P} e^{r_2}} e^{-\frac{\mu}{P} e^{ar_2}} d_{r_2}}{\int_0^{\infty} \frac{\mu}{P} e^{r_2} e^{-\frac{\mu}{P} e^{r_2}} e^{-\frac{\mu}{P} e^{ar_2 - b}} d_{r_2}} \end{aligned} \tag{6.10}$$

Since $R_{min} < r_2 < R_{max}$, p^{π} can be approximated as follows,

$$p^{\pi} \approx \frac{\int_{R_{min}}^{R_{max}} e^{r_2} e^{-\frac{\mu}{P} e^{r2}} e^{-\frac{\mu}{P} e^{ar_2}} d_{r_2}}{\int_{R_{min}}^{R_{max}} e^{r_2} e^{-\frac{\mu}{P} e^{r2}} e^{-\frac{\mu}{P} e^{-b} e^{ar_2}} d_{r_2}} \tag{6.11}$$

Note that the following inequality holds,

$$e^{-\frac{\mu}{P} e^{-b} e^{ar_2}} \leq e^{-\frac{\mu}{P} e^{ar_2} + \mu B}$$

where $B = e^{aR_{max}}(1 - e^{-b})$. Hence,

$$\int_0^{\infty} e^{r_2} e^{-\frac{\mu}{P} e^{r2}} e^{-\frac{\mu}{P} e^{-b} e^{ar_2}} d_{r_2} \leq$$
$$e^{\mu B} \int_0^{\infty} e^{r_2} e^{-\frac{\mu}{P} e^{r2}} e^{-\frac{\mu}{P} e^{ar_2}} d_{r_2}$$

By applying this result to (6.11), we have

$$p^{\pi} \geq e^{-\mu B} \tag{6.12}$$

We know also that

$$Q_2 \leq Q_1 \frac{\hat{R}_1^{\pi}}{\hat{R}_2^{\pi}}$$

Since $R_{min} \leq \hat{R}_i^{\pi} \leq R_{max}$, $i = 1, 2$ we have,

$$Q_2 \leq Q_1 \frac{R_{max}}{R_{min}}$$

By applying this result to (6.12), we have,

$$p^{\pi} \geq \exp\left(-\frac{\mu}{P}\left[1 - e^{-e_{max}\left(1+\frac{R_{max}}{R_{min}}\right)}\right] e^{\frac{(R_{max})^2}{R_{min}}}\right) \ \forall t.$$

Hence, $\epsilon = \exp\left(-\frac{\mu}{P}\left[1 - e^{-e_{max}\left(1+\frac{R_{max}}{R_{min}}\right)}\right] e^{\frac{(R_{max})^2}{R_{min}}}\right)$. Policy π can achieve ϵ fraction of rate region of Max-Weight algorithm with full CSI. □

Remark: Note that ϵ increases, (which in turn increases achievable rate region according to Theorem 6.1), as the maximum estimation error e_{max} decreases. Since the prediction is based on past channel observations, the quality of prediction not only depends on the method used in prediction but also the set of channel observations, i.e., $\mathcal{S}(t)$. Note that the joint policy $\pi(\eta)$ takes into account only the queue sizes with estimated transmission rates. Hence, it is possible that under policy $\pi(\eta)$ some channels may not be probed for a long time (i.e., the users with small queue sizes), as a result, the prediction accuracy on those channels will be poor especially with fast fading. To prevent this problem, a scheduling policy must efficiently balance the "exploitation-exploration tradeoff", i.e., acquiring channel states of users with high expected weights to achieve higher throughput (exploitation) vs. learning channel states of users which have not been probed for a long time (exploration). Therefore, the problem here is not just tracking the channels accurately (see [87]) but also optimizing channel probing process. The framework in this work is related to *active learning* schemes in the literature [99], where the decision maker is allowed to choose the data from which it learns, and the optimization of the observation of the process is coupled with learning.

6.4 Gaussian Process Regression for Channel Probing and Scheduling

6.4.1 Problem Formulation

Note that channel prediction is inherently error-prone. The degree of uncertainty in the estimate of the current channel state depends on the previous channel observations, and the dynamics of the channel. In this context, we define *information of an unexplored channel* as the uncertainty in the channel state given its past observations. Obtaining this information accurately is important since making any scheduling and probing decision in a principled manner on a given time slot necessitates first measuring this information. For instance, if the channels are probed uniformly randomly at each time slot each remaining unexplored channel provides equivalent amount of information. However, not all channels vary in the same fashion, and depending on the prediction method used and previous information collected different unexplored channels provide different amount of information. Thanks to Shannon's entropy formulation [5], uncertainty in channel states can be exactly quantified. Let $I_n^{\pi(\eta)}(t)$ denote the *information* of channel state of user n under policy $\pi(\eta)$ at the beginning of time slot t given past observations of the channel. For instance, the information obtained by probing a channel whose state was observed recently and many times before is less than the channel which has not been probed for a long time, since the uncertainty in the state of the latter is higher.

Hence, we have two closely related objectives: (1) To make a scheduling decision that stabilizes a network with the largest possible achievable region; (2) To probe a set of users that minimizes the channel prediction error.

- objective 1: $\max \sum_{n=1}^{N} Q_n(t)\hat{R}_n^{\pi(\eta)}(t)$

- objective 2: $\max \sum_{n=1}^{N} I_n^{\pi(\eta)}(t)$

We seek a joint feasible policy $\pi(\eta)$, which determines a subset of users probed by considering both objectives, and schedules a user out of this subset according to Max-Weight algorithm. The most common approach to find the solution of multi-objective optimization problems is the weighted sum method [100]. The problem is stated as follows:

$$\max_{\pi(\eta)\in\mathcal{F}} \sum_{n=1}^{N} \alpha_1 Q_n(t)\hat{R}_n^{\pi(\eta)}(t) + \alpha_2 I_n^{\pi(\eta)}(t), \tag{6.13}$$

where α_1 and α_2 are the weights assigned to each objective according to their relative importance. Note that the first term in the summation refers to the first objective that aims to schedule the user with maximum weighted throughput for the stability of the network. The second term in (6.13) refers to the second objective that aims to probe those users with maximum information to decrease the uncertainty about the channels. We also note that the scheduling and probing decision depends not only on the queue sizes and the estimated channel rates as in the original Max-Weight algorithm, but also on the uncertainty in each channel state given its past observations. The problem (6.13) also exhibits the well-known "*Exploration vs. Exploitation*" trade-off by exploiting the users with high weighted rate and exploring the current state of the channels with outdated CSI. In the following sections, we deal with a modified version of this problem, where we divide the objective function in (6.13) by α_1, and define a single weight $\xi = \frac{\alpha_2}{\alpha_1}$. Note that when ξ is tuned to higher (lower) values, we track the channels more (less) closely.

The problem given in (6.13) involves estimating $R_n(t)$ by employing a particular estimator η. This is known as the regression problem in machine learning which is also a supervised learning method since the observed data constitutes at the same time the learning data set. There is a plethora of work for carrying out regression analysis. In this work, we employ Gaussian Process Regression (GPR) to predict channel states. Before explaining how a channel state is predicted with GPR in detail, we first give the main reasons behind our choice.

Although prediction of fading channels is a well studied topic [87], the implementation of these prediction methods is difficult. This is mainly because linear regression, or autoregressive techniques used by these prediction methods involves parameters that should be chosen judiciously based on the channel statistics in order for these methods to perform well. GPR does not have such an issue, and past analysis showed that it performs very well in comparison to other techniques [101]. In addition to its good performance, GPR has the advantage that it is - its mathematical background left aside - easy to handle. Also, unlike conventional statistical methods, GPR is a model-based machine learning approach which adapts Bayesian model on channel process and learns while obtaining an on-line experience. Such methods are known as active learning in the literature [99].

In contrast to other regression models, GPR also provides a simple way to measure the expected uncertainty in the prediction for any given set of CSI observations, which is particularly important for our joint probing and scheduling policy. The prediction methods for fading channels mentioned in [87] can only predict the future channel state without giving any information on the expected uncertainty in the prediction.

Finally, GPR gives decisions based only on the most recent channel observations. This is especially important for non-stationary channels, since past channel observations may become outdated. In the following, we use the terms $\hat{c}_n(t)$, $\hat{R}_n(t)$ and $I_n(t)$ to denote that these values are obtained by employing GPR.

6.4.2 Prediction of Channel States with GPR

Let $\mathcal{D}_n(t) = (\mathbf{c}_n, \boldsymbol{\tau}_n)$ denote the set of observations for channel n at the beginning of time slot t, where $\mathbf{c}_n = \{c_n^1, c_n^2, \dots, c_n^w\}$ denotes the set of the latest w CSI values taken at times, $\boldsymbol{\tau}_n = \{\tau_n^1, \tau_n^2, \dots, \tau_n^w\}$, and $\tau_n^i < t$, for all $\tau_n^i \in \boldsymbol{\tau}_n$, $i \in \{1, 2, \dots, w\}$. We use GPR to predict the value of CSI, i.e., $\hat{c}_n(t)$ at the beginning of time slot t, given $\mathcal{D}_n(t)$.

Let $p(c_n(t)|t, \mathcal{D}_n(t))$ be a posterior distribution of channel n. Note that the foundation of the approach adopted GPR is Bayesian inference, where the main idea is to choose an a priori model and update it with actual experimental data observed. According to GPR, a posterior distribution is Gaussian with mean $\hat{c}_n(t)$ and variance $v_n(t)$. Specifically, Gaussian process is specified by the kernel function, $k_n(\tau_n^i, \tau_n^j)$, that describes the correlation of channel n between two of its measurements taken at times τ_n^i and τ_n^j. It is possible to choose any positive definite kernel function. However, the most widely used is the squared exponential, i.e., Gaussian, kernel:

$$k_n(\tau_n^i, \tau_n^j) = \exp\left[-\frac{1}{2}(\tau_n^i - \tau_n^j)^2\right]. \tag{6.14}$$

Given $\mathcal{D}_n(t)$, $\hat{c}_n(t)$ and variance $v_n(t)$ are determined as follows:

$$\hat{c}_n(t) = \mathbf{k}_n^T(t)\mathbf{K}_n^{-1}\mathbf{c}_n, \tag{6.15}$$

$$v_n(t) = k_n(t,t) - \mathbf{k}_n^T(t)\mathbf{K}_n^{-1}\mathbf{k}_n(t), \tag{6.16}$$

where $\mathbf{K}_n$ is a $w \times w$ matrix composed of elements $k_n(\tau_n^i, \tau_n^j)$ for $1 \leq i, j \leq w$ and $\mathbf{k}_n(t)$ is a vector with elements $k_n(\tau_n^i, t)$ for $\forall \tau_n^i \in \boldsymbol{\tau}_n$. Hence, the network scheduler can easily predict the CSI of users at time t by using (6.15). Furthermore, the variance $v_n(t)$ is used to measure the level of uncertainty in the predictions, i.e., $I_n(t)$ as discussed next.

6.4.3 Quantifying Information in GPR

The mean square error (MSE) has been a popular criterion in training of adaptive systems [102]. MSE is based on second order statistics, i.e., it is able to extract information successfully for linear systems whose statistics can be defined by their mean and variance. For nonlinear systems, entropy is the ideal measure for uncertainty and it extends MSE.

Recall that Shannon's **entropy** of a random variable A is defined as $H(A) = \sum_s p_s \log_s(\frac{1}{p_s})$, where $p(.)$ is the probability distribution function of A [5]. In our context, the current realization of CSI, i.e., $c_n(t)$, is a random variable. Accordingly, let

$H_n^0(c_n(t)|t, \mathcal{D}_n(t))$ and $H_n^1(c_n(t)|t, \mathcal{D}_n(t))$ denote the entropy of the random variable $c_n(t)$ before and after time slot t, respectively when $\mathcal{D}_n(t)$ is given. If channel n is probed at time t, then $H_n^1(c_n(t)|t, \mathcal{D}_n(t))$ will be zero since the channel state is known exactly. Otherwise, the uncertainty increases, i.e., $H_n^1(c_n(t)|t, \mathcal{D}_n(t)) > H_n^0(c_n(t)|t, \mathcal{D}_n(t))$. Hence, the information acquired by probing channel of user n is the reduction in its uncertainty, which is simply the difference between its entropies before and after the probing:

$$I_n(t) = H_n^0(c_n(t)|t, \mathcal{D}_n(t)) - H_n^1(c_n(t)|t, \mathcal{D}_n(t)).$$

The following Proposition is similar to the one given in [85], and establishes that information obtained by probing a channel is equal to the variance of the estimate of the state of that channel.

Lemma 6.2. *Given $\mathcal{D}_n(t), \forall n = 1, \ldots, N$, finding the channel that has the highest information at time slot t is equal to finding the channel which has the highest variance at that time slot, i.e.,*

$$i^* = \operatorname*{argmax}_{1 \le n \le N} I_n(t) = \operatorname*{argmax}_{1 \le n \le N} v_n(t). \tag{6.17}$$

Proof. Since $H_n^1(c_n(t)|t, \mathcal{D}_n(t)) = 0$ after probing, $I_n(t)$ is simply

$$I_n(t) = H_n^0(c_n(t)|t, \mathcal{D}_n(t)). \tag{6.18}$$

Note that according to GPR a posterior distribution of state of channel given $\mathcal{D}_n$ is

$$p(c_n(t)|t, \mathcal{D}_n) \sim \mathcal{N}(\hat{c}_n(t); v_n(t)). \tag{6.19}$$

Then, the entropy of this Gaussian distribution is given by,

$$H_n^0(c_n(t)|t, \mathcal{D}_n) = \frac{1}{2} \log(2\pi e v_n(t)). \tag{6.20}$$

Hence, $i^* = \operatorname{argmax}_{1 \le n \le N} I_n(t) = \operatorname{argmax}_{n \in \mathcal{N}} v_n(t)$. $\square$

6.4.4 Joint Scheduling and Probing Algorithm

Algorithm 2: MOSF Algorithm

(1) probing decision:

- Step 1: Sort

$$W_n \triangleq Q_n(t)\hat{R}_n(t) + \xi I_n(t),$$

  in a descending order. Tie is broken randomly.

- Step 2: Construct $\mathcal{S}(t)$ by selecting the first L users in this order.

(2) scheduling decision:
The base station acquires CSI of each user in $\mathcal{S}(t)$ and user $n^* \in \mathcal{S}(t)$ is scheduled to transmit,

$$n^* = \operatorname*{argmax}_{n \in \mathcal{S}(t)} Q_n(t)R_n(t). \tag{6.21}$$

i.e., $\mathcal{J}_{n^*}(t) = 1$, and updates queue lengths according to (6.2).

In this section, we give our Multi-Objective Scheduling and Feedback (MOSF) algorithm. MOSF takes into account both the estimated transmission rates of users and the information acquired from each user.

Lemma 6.3. *Given L, ξ, $\mathbf{Q}(t)$, $\hat{R}_n(t)$ and $I_n(t)$ determined according to GPR for each user at time slot t, MOSF algorithm solves* (6.13)*.*

Proof. Note that the solution of problem (6.13) requires $Q_n(t)$, $\hat{R}_n(t)$ and $I_n(t)$, which are available at the beginning of every time slot at the BS. Let U be the subset that contains arbitrary L users, i.e., $U \in \mathcal{G}$, where $\mathcal{G}$ is the union of all possible subsets. The solution of (6.13) is found by determining the optimal probing set which is the solution of the following optimization problem;

$$\max_{U \in \mathcal{G}} \max_{n \in U} Q_n(t)R_n(t)$$

Since MOSF algorithm acquires CSI from the top L users with the maximum $W_n(t)$, then $\mathcal{S}(t)$ is given by,

$$\mathcal{S}(t) = \max_{U \in O_L} \max_{n \in U} Q_n(t)R_n(t)$$

Hence, MOSF solves (6.13). $\square$

Next, we analyze the performance of MOSF algorithm by comparing it with Max-Weight algorithm. Let n^* be the user scheduled by Max-Weight algorithm with complete CSI at time t. Under the worst case scenario, i.e., $L = 1$, MOSF algorithm schedules user n^* if the maximum prediction error is below a certain threshold as given by the following Lemma. Note that when $L > 1$ the quality of the prediction improves.

Lemma 6.4. *For $L = 1$, MOSF algorithm schedules user n^* at time t, if the maximum prediction error e_{max} satisfies the following inequality for all $n \neq n^*$,*

$$e_{max} \leq \frac{Q_{n^*}(t)R_{n^*}(t) - Q_n(t)R_n(t) + \xi(I_{n^*}(t) - I_n(t))}{Q_{n^*}(t) + Q_n(t)}, \tag{6.22}$$

Proof. Since n^* is the scheduled user with Max-Weight algorithm with full CSI, the following inequality is true,

$$Q_{n^*}(t)R_{n^*}(t) \geq Q_n(t)R_n(t) \ \forall n \neq n^*$$

When MOSF algorithm decides to schedule user n^* then the following inequality must be satisfied,

$$Q_{n^*}(t)\hat{R}_{n^*}(t) + \xi I_{n^*}(t) \geq Q_n(t)\hat{R}_n(t) + \xi I_n(t), \ \forall n \neq n^*$$

We consider the worst case scenario in which $e_n(t) = e_{max}$, for all n and GPR underestimates for channel n^* whereas it overestimates for all other channels, i.e.,

$$\hat{R}_{n^*}(t) = R_{n^*}(t) - e_{max}$$
$$\hat{R}_n(t) = R_n(t) + e_{max}, \forall n \neq n^*$$

Hence, the following inequality must be true so that MOSF algorithm schedules user n^* in the worst case,

$$Q_{n^*}(t)(R_{n^*}(t) - e_{max}) + \xi I_{n^*}(t) \geq \\ Q_n(t)(R_n(t) + e_{max}) + \xi I_n(t), \forall n \neq n^*$$

Thus, e_{max} should satisfy the following condition,

$$e_{max} \leq \\ \frac{Q_{n^*}(t)R_{n^*}(t) - Q_n(t)R_n(t) + \xi(I_{n^*}(t) - I_n(t))}{Q_{n^*}(t) + Q_n(t)} \tag{6.23}$$

□

Remark: Lemma 6.4 gives the sufficient condition for the same user to be scheduled by MOSF and Max-Weight algorithms. Note that this condition always holds when $I_{n^*}(t) \geq I_n(t)$ for all $n \neq n^*$, and $\xi \to \infty$ since in this case the user which has the maximum weight and the maximum information is scheduled. On the other hand, the condition does not hold if there is at least one user such that $I_{n^*}(t) \leq I_n(t)$ and as ξ increases. This is because, as ξ increases MOSF algorithm emphasizes the information rather than the weighted rate.

Note that as long as MOSF schedules the user with the actual maximum weight correctly at each slot (i.e., the condition in (6.22) is satisfied), then the rate region achieved by MOSF approaches that of the maximum achievable rate region, Λ_h. The achievable rate region of MOSF can be obtained based on Theorem 6.1, and Lemma 6.4. Let p_{min} be the probability that (6.22) holds for all t. Then, we have the following theorem.

Theorem 6.3. *Given queue sizes at each time slot, MOSF algorithm can achieve a fraction ϵ of the rate region, Λ_h, where $\epsilon = p_{min}$.*

Proof. Let $\mathcal{J}_n^m(t)$ represent the scheduling decision with MOSF algorithm. $\mathcal{J}_n^m(t) = 1$ if user n is scheduled with MOSF. Otherwise, $\mathcal{J}_n^m(t) = 0$. Let the condition in Lemma 6.4 hold with probability $\rho^m(\mathbf{Q}(t))$ at time slot t. We consider the following function:

$$g_m(\mathbf{Q}(t)) = \mathbb{E}\left[\sum_n Q_n(t)R_n(t)\mathcal{J}_n^m(t)|\mathbf{Q}(t)\right]$$

Using arguments similar to those in Theorem 6.1, we have

$$\frac{g_m(\mathbf{Q}(t))}{g_f(\mathbf{Q}(t))} \geq \rho^m(\mathbf{Q}(t))$$

If $\rho^m(\mathbf{Q}(t)) \geq p_{min}$ for all t, then MOSF can achieve a fraction $\epsilon = p_{min}$ of rate region Λ_h.

□

Note that the exact value of p_{min} depends on the channel characteristics, and can be calculated sin a similar fashion as we did in Lemma 6.1. As the number of probed users increases, i.e., as L increases, the prediction error decreases. This is due to the fact that channels are more frequently probed, which in turn helps track the channel states more closely. On the other hand, since the channel states are tracked more closely, the uncertainty in the states of the channels decrease. As a result, the information acquired from an unexplored channel decreases as well, i.e., $I_n(t)$ decreases.

6.5 Numerical Analysis

In our simulations, we model a single cell CDMA downlink transmission utilizing high data rate (HDR) [4]. The base station serves 16 users and keeps a separate queue for each user. Time is slotted with length $T_s = 1.67$ ms as defined in HDR specifications. Packets arrive at each slot according to Bernoulli distribution: The size of a packet is 128 bytes which corresponds to the size of an HDR packet. The wireless channel is

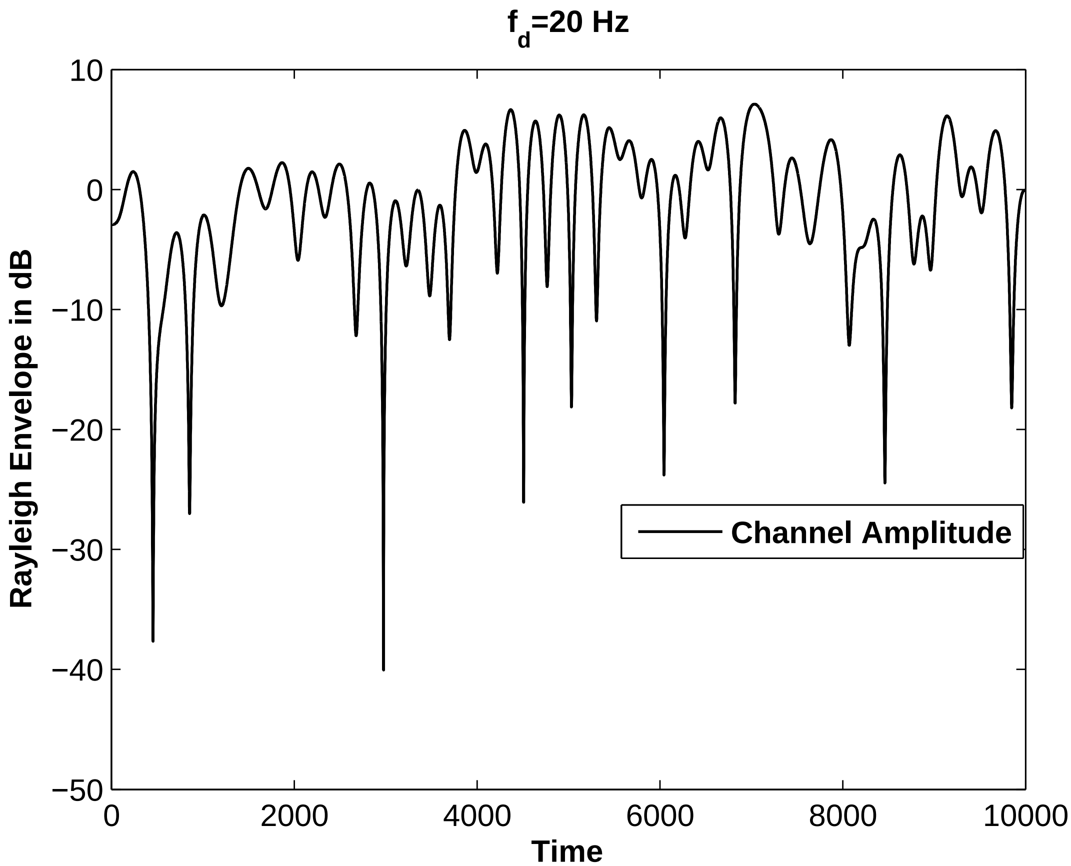

Figure 6.1: A typical Rayleigh fading channel.

modeled as *correlated* Rayleigh fading channel according to Jakes's model [103]. The bandwidth of the system is $BW = 1.25$ MHz and transmission power of the base station is $P = 10$ dB. The transmission rate of the user channel is taken as the maximum value given in (1). The channel sampling rate is 600 Hz (i.e., the sampling time $\mathcal{T}$ is 1.67 ms.), which also matches the Data Rate Request Channel (DRC) update rate used in HDR. Doppler frequency of each channel is randomly chosen in the range $f_d = [5, 20]$ Hz. We divide the users into two groups with eight users in each. The users in the first group experience slow fading, i.e., $f_d/f_s \leq 0.02$, and the users in the second group experience fast fading, i.e., $f_d/f_s \geq 0.02$, where $f_s = 600Hz$ is the sampling rate of the channel. We compare MOSF algorithm with the algorithm, named LAR in [87] that estimates future values of the fading coefficient of a time-correlated channel by employing the well-known autoregressive (AR) model. According to AR model, the current CSI of a user can be predicted when p previous CSIs of that user are given,

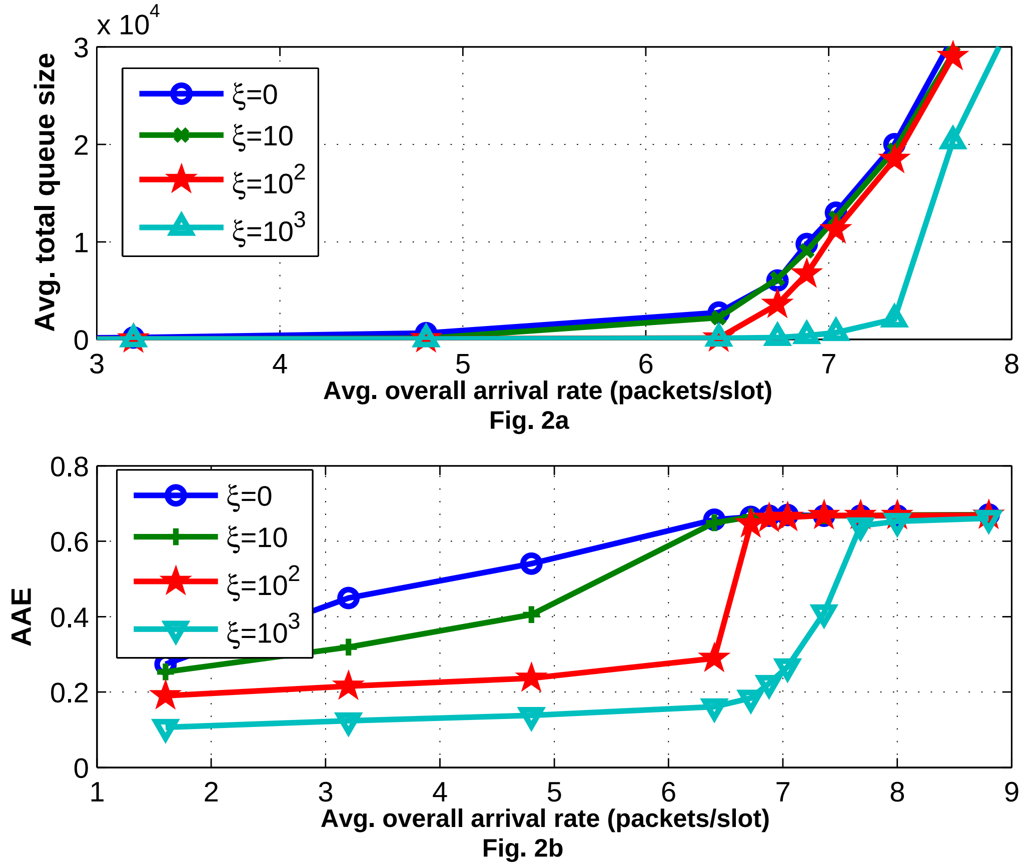

Figure 6.2: Average total queue backlogs and absolute channel estimation error.

where p is the order of AR model. After predicting the current channel states of all users, LAR algorithm probes L users with the highest estimated weighted rates and schedules the user with the maximum weight in the set of probed users at every slot. Recall that MOSF algorithm employs GPR for channel state prediction based on w most recent observations. We empirically observe that the minimum prediction error is achieved when $p = 2$ with AR and $w = 3$ with GPR, and thus, these values are taken throughout all experiments. The performance of the algorithms are measured in terms of the average queue sizes and average estimation error. Note that the average queue size is an indicator to the average delay experienced by the users. Also note that by inspecting the average rate of change of queue sizes, we can approximately determine the maximum arrival rates that can be supported by different algorithms. The lower bound for average queue sizes is given by Max-Weight algorithm which has full CSI at every time slot. The estimation error in LAR and MOSF is measured as average

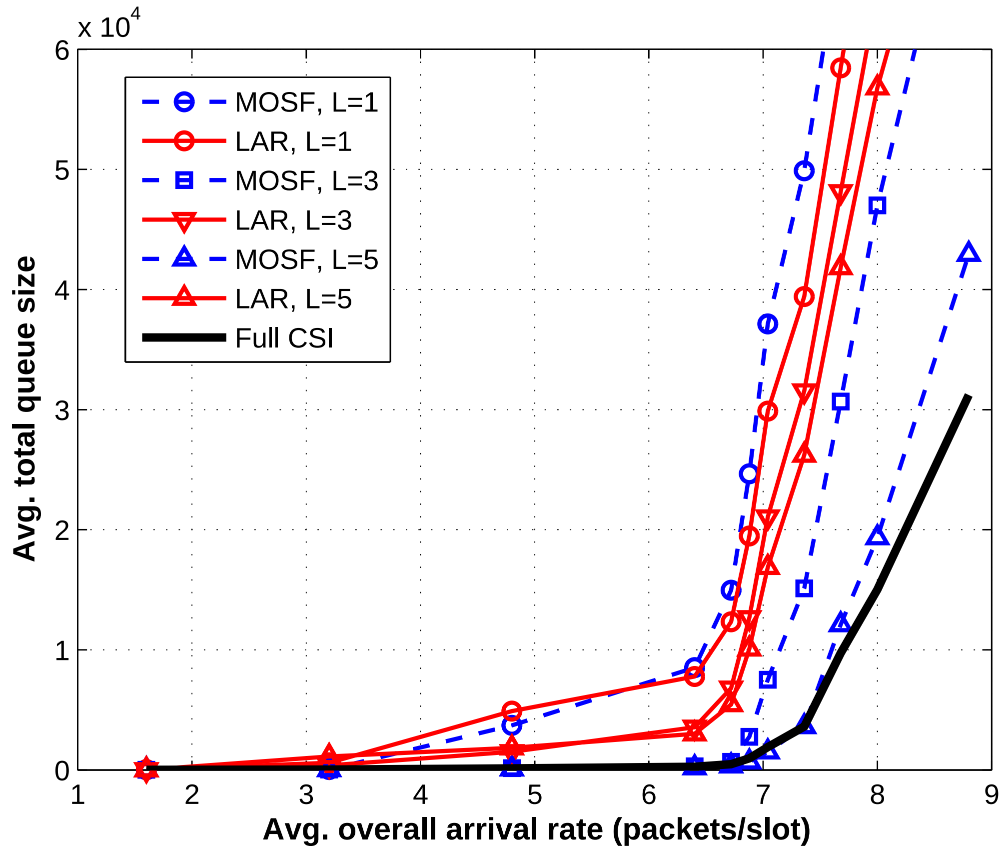

Figure 6.3: Average Total Queue Backlogs with MOSF and LAR.

absolute error (AAE):

$$AAE = \lim_{t\to\infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{n=1}^{N} \left| c_n(t) - \hat{c}_n^{\pi}(t) \right| .$$

We first conduct an experiment to show the effect of taking into account the uncertainty in channel states while making probing decisions. Hence, we compare MOSF algorithm for varying values of ξ. In order to keep the plots simple we only depict results for $L = 4$. Figure 6.2a shows the average total queue sizes with respect to arrival rates. When the information is not taken into account, i.e., when $\xi = 0$, the network can be stabilized for small arrival rates. For instance, for $\xi = 0$ the stabilizable arrival rate is approximately 5 packets/slot, whereas for $\xi = 10^3$, the network can be stabilized for arrival rates up to 7.5 packets/slot. Figure 6.2b depicts the average error in channel estimation. Clearly, as ξ increases, the estimation error decreases since the channels are tracked more accurately

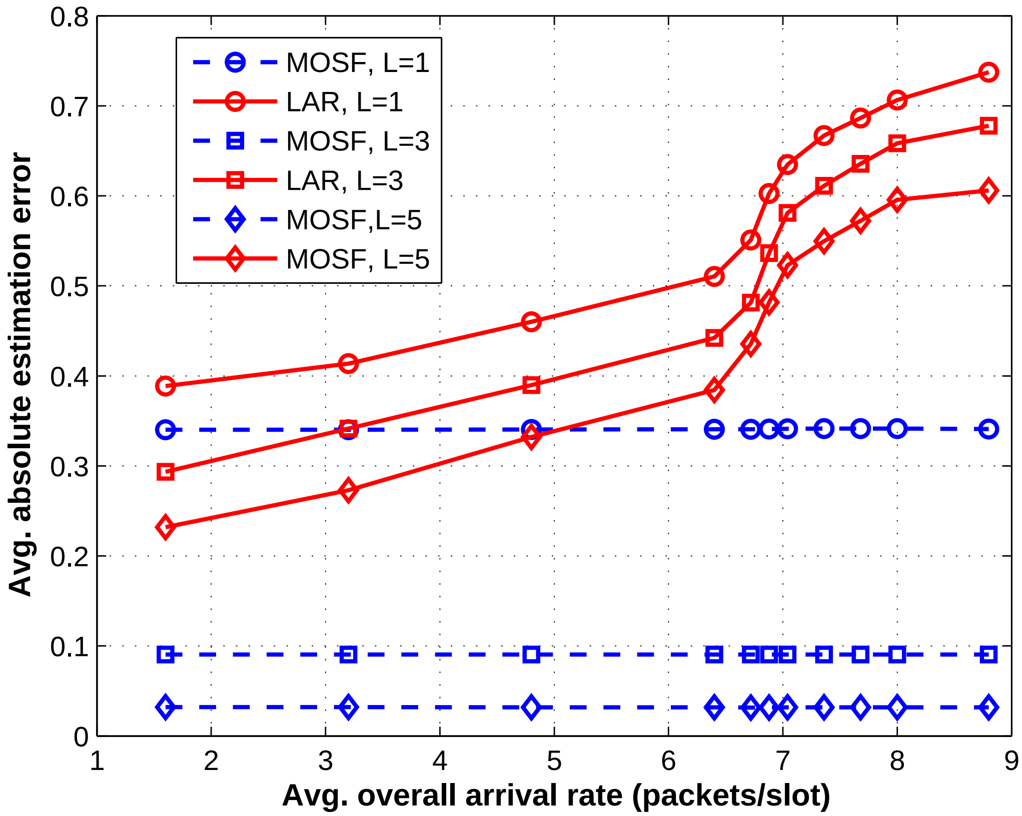

Figure 6.4: Average absolute channel estimation error with respect to arrival rates

for higher values of ξ. This experiment indicates that the information acquired from the channels must be taken into account for scheduling and probing decision to achieve larger rate region and smaller channel estimation error. Figure 6.3 depicts the average total queue sizes in terms of packets with respect to arrival rates for MOSF and LAR. The exploitation factor is set to be $\xi = 10^5$. As shown in Figure 6.3, for $L = 1$ MOSF algorithm is not throughput-optimal since it cannot to stabilize the network for all arrival rates. For instance, when the arrival rate is around 3 packets/slot, the average total queue size suddenly increases, which shows the instability of the network. This is because the learning algorithm does not have sufficiently frequent observations to accurately predict the channel state. However, as L is increased to $L = 3$ the network is stabilized for higher arrival rates (i.e., it stabilizes the network for 6.5 packets/slot), but MOSF still does not stabilize the network for all arrival rates. When we further increase the feedback to $L = 5$, then MOSF algorithm has a comparable performance

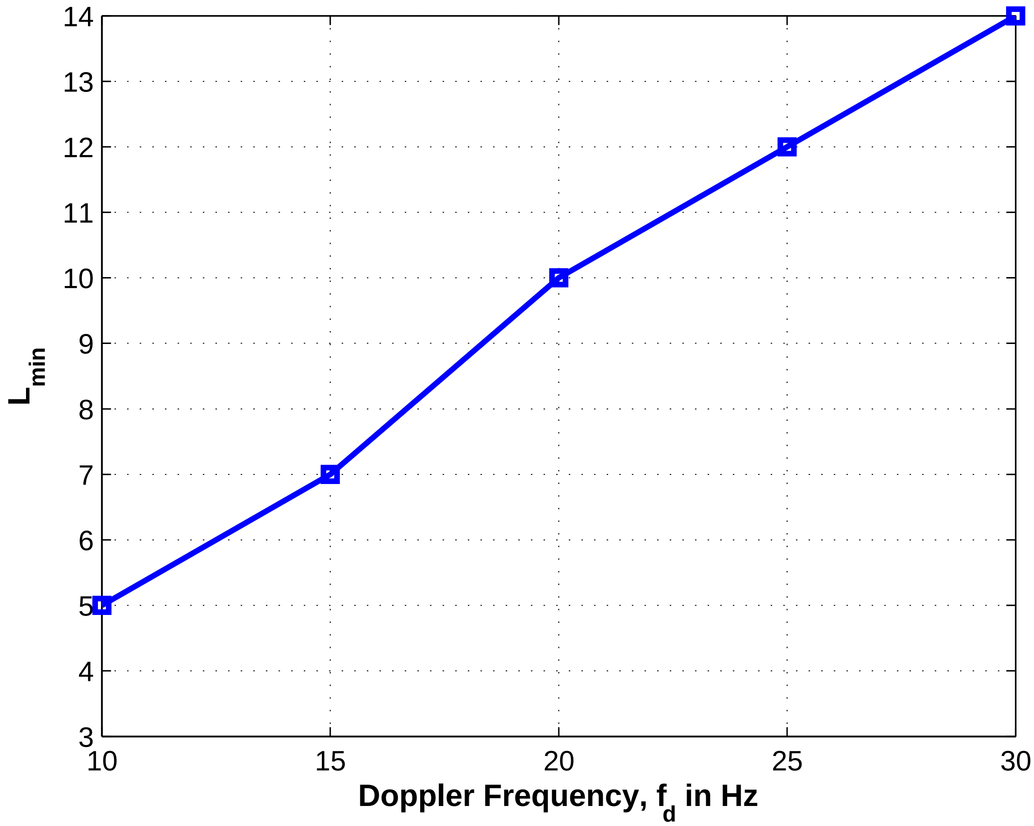

Figure 6.5: The required size of the feedback channel for MOSF to achieve Λ_h.

as that of Max-Weight algorithm with full CSI. On the other hand, LAR algorithm also achieves its best performance when $L = 5$, however, even in that case it cannot stabilize the network for all arrival rates. Hence, we conjecture that LAR achieves approximately smaller rate region as compared to MOSF algorithm.

Next, we investigate the performance of MOSF and LAR algorithms in terms of average absolute error in channel estimation (AAE). As depicted in Figure 6.4, as the queue sizes increase, the estimation error increases with LAR. The increase in the error with MOSF algorithm is small and it is more robust to queue sizes than LAR. Moreover, the best error performance for both prediction algorithms is achieved with $L = 5$ and when the arrival rate is at its lowest value. In that case, the average absolute error with MOSF is 0.03 whereas it is equal to 0.23 with LAR.

Finally, we determine the minimum number of channels, L_{min}, required to stabilize the network for a given arrival rate and to achieve the similar delay performance as with

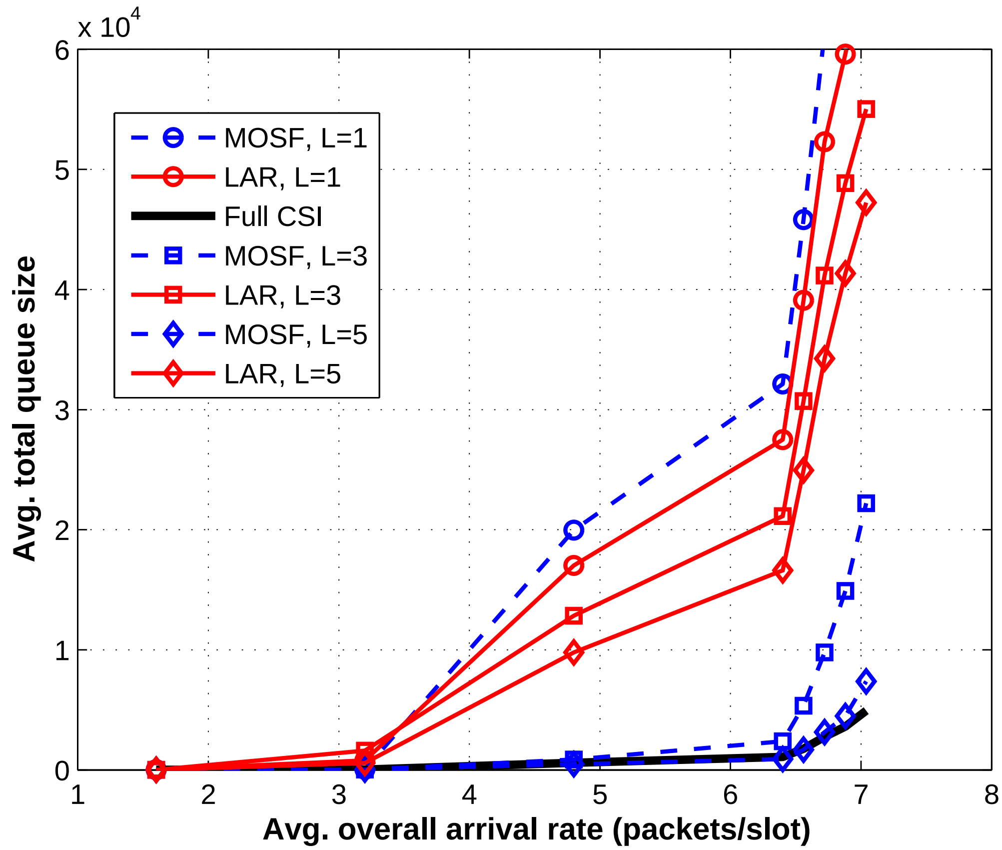

Figure 6.6: Performances of MOSF and LAR over non-stationary channels.

Max-Weight algorithm with full CSI. We set the average total arrival to the network 5 packets/slot. As depicted in Figure 6.5, when $f_d = 10$ Hz for all channels, the base station has to probe at least $L_{min} = 5$ channels to achieve the same rate region and the average delay performance. As f_d increases L_{min} increases as well. This is due to the fact that at a higher Doppler frequency the channel states vary faster, which in turn necessitates more observations for the learning algorithm to accurately predict the current channel state. Hence, the base station should collect CSI more frequently as f_d increases.

Performance over Non-stationary channels: In non-stationary environments, we assume that velocity of users (i.e., doppler frequencies) change after certain time slots and users do not have any knowledge about these changes. To simulate a non-stationary environment, up to $t = 5 \times 10^4$ we increase the normalized Doppler frequency f_d/f_s at a constant rate from 0.003 to 0.03. After time 5×10^4, we decrease f_d/f_s at a constant rate. This model approximately represents the movement of a mobile user such that the user velocity is increasing from $t = 0$ to $t = 5 \times 10^4$ and it is slowing from $t = 5 \times 10^4$ to $t = 10^5$. Figure 6.6 depicts that MOSF algorithm outperforms LAR in terms of rate region even in the dynamically changing channels. This is due to the fact that the quality of prediction of GPR depends on the most recent channel observations and how fast channels change rather than the distribution of channels. Thus, as long as channels are tracked closely, predicting future CSI will be more accurate.

6.6 Chapter Summary

We have developed a joint scheduling channel probing algorithm for general wireless channel models. In our model, the base station can only probe limited number of channels due to the bandwidth constraint on the feedback channel. The proposed algorithms first decide the set of channels that must be probed at the beginning of each time slot. The set of channels is determined by considering not only the queue sizes and estimated transmission rates but also by the information obtained by probing a channel. We apply Gaussian Process Regression technique to predict CSI at each time slot based on the previous actual CSI observed. In numerical results, we show that the base station using MOSF can stabilize the network and achieve a similar delay performance as compared to full CSI Max-Weight algorithm by probing less than half of the users at every slot.

Our work deviates from the current line of literature using machine learning for solving networking problems in the way that it addresses learning in a more general network setting. For this purpose, we develop an active learning framework that quantifies the reward of learning the current state of the system using the entropy measure. Based on this measure, we were able to make an intelligent trade off between having a more up-to-date picture of the system and maximizing the overall system throughput.

Review of Gaussian Process Regression

We provide here a brief introduction to Gaussian Process Regression for the sake of completeness. We refer the interested readers to [86] for further details. In GPR, the aim is to infer or learn $f(x)$ using the set of previously observed data points. Given a set of data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_w, y_w)$ which is also called training data or set, GPR attempts to find a function $\hat{f}(x)$ that provides a good approximation to $f(x)$. GPR approaches this problem by defining a probability distribution over a set of admissible functions and performing Bayesian inference over this set. A GPR is formally defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean function $m(x)$ and covariance function $C(x, \bar{x})$, where,

$$
\begin{aligned}
m(x) &= E[\hat{f}(x)] \\
C(x, \bar{x}) =& E[\hat{f}(x) - m(x)(\hat{f}(\bar{x}) - m(\bar{x}))], \forall x, \bar{x} \in \mathcal{D}
\end{aligned}
$$

Let us choose $m(x) = 0$ as a simplifying assumption. Then, GPR can be defined solely by its covariance function $C(x, \bar{x})$. This covariance matrix is also know as the kernel. Hence, The Gaussian process is denoted by,

$$
\hat{f}(x) \sim \mathcal{GP}(m(x), C(x, \bar{x})).
$$

The covariance function can be also defined as sum of kernel function and the diagonal noise variance such that,

$$
C(x, \bar{x}) = K(x, \bar{x}) + \sigma \mathrm{I}, \forall x, \bar{x} \in \mathcal{D}, \tag{6.24}
$$

where I is the identity matrix. It is possible to choose any positive definite kernel function. In this work, we choose the following function:

$$
K(x, \bar{x}) = \exp\left[-\frac{1}{2}(x - \bar{x})^2\right]. \tag{6.25}
$$

Let x^* be the new test point outside the training set and $f(x^*) = y^*$. The conditional Gaussian distribution of y^* given the previous data set $\mathcal{D}$ can be computed as follows:

First, define the vector

$$k(x^*) = [K(x_1, x^*), K(x_2, x^*), ..., K(x_M, x^*)]. \tag{6.26}$$

and the scalar

$$\kappa(x^*) = K(x^*, x^*) + \sigma. \tag{6.27}$$

Then, $p(y^*|\mathcal{D})$ that characterizes the $\mathcal{GP}(0, C)$ is a Gaussian $\mathcal{N}(\hat{f}, v)$ with mean $\hat{f}$ and variance v,

$$\hat{f}(x^*) = k^T(x^*)C^{-1}(x^*)y, \tag{6.28}$$

$$v(x^*) = \kappa(x^*) - k^T(x^*)C^{-1}(x^*)k(x^*). \tag{6.29}$$

This is a key result that defines GPR as the mean function $\hat{f}(x)$ of the Gaussian distribution provides a prediction of the objective function $f(x)$. Furthermore, the variance function $v(x)$ can be used to measure the uncertainty level of the predictions provided by $\hat{f}$.

Chapter 7

Throughput-Optimal Distributed Algorithm

In Chapter 5 and Chapter 6, we have considered downlink communication where users receive data from the base station. In this Chapter, we consider uplink communication where users are willing to transmit to the base station. For uplink communication the scheduler has to know not only CSI of users but also queue length information to make the best decision, which brings much more cost than that of downlink system. Hence, recently, research comminty have focused on CSMA-type random access protocols for uplink communication, and it has been shown that CSMA-type random access algorithms are throughput optimal since they achieve the maximum throughput while maintaining the network stability. However, the optimality is established with the following unrealistic assumptions; i) the underlaying Markov chain reaches a stationary distribution immediately, which causes large delay in practice; ii) the channel is static and does not change over time. In this Chapter, we design fully distributed scheduling algorithms which are provably throughput optimal for general fading channels. When arbitrary backoff time is allowed, the proposed distributed algorithm achieves the same performance in terms of rate region and delay as that of a centralized system without requiring any message passing. For the case where backoff time is discrete, we show that our algorithm still maintains throughput-optimality and achieves good delay performance at the expense of low overhead for collision resolution.

7.1 Overview

Recall that the optimality of Max-Weight algorithm requires full global knowledge of network states (i.e., complete channel states and queue sizes information from all users) at every time slot. The availability of complete network information depends on the type of communication. Septically, for *downlink* network the base station is already aware of queue size information of users, and the main difficulty in implementing Max-Weight type scheduling policies is having access to channel condition of users. On the other hand, for uplink communication the base station must acquire not only complete CSI but also queue length information to employ Max-Weight scheduling ,which brings much more overhead than that of a downlink system. Hence, the overhead of Max-Weight policy has motivated many researchers to develop distributed algorithms for the practical implementation of Max-Weight policy.

One of most popular random access protocols in practice is Carrier Sense Multiple Access (CSMA) in which each user senses the medium and transmits a packet only if the medium is sensed idle. Hence, this simple nature makes it suitable for distributed implementations, and as a result it has been regarded as one of the most attractive MAC protocols in practical wireless systems such as for IEEE 802.11 wireless networks. Thus, a large body of work in the literature has been devoted to the development of CSMA based distributed algorithms under different performance criteria and constraints.

In this work we study distributed scheduling algorithms based on CSMA with the aim of achieving the largest set of arrival rates under which the algorithms can keep the queues in the network stable. Since many wireless network applications have stringent delay requirements, designing distributed scheduling algorithms is important not only to achieve the maximum throughout but also to provide low delay, which is the another objective of this work. Also, designing distributed algorithms with low overhead is crucial since many wireless systems have limited resources.

Our contributions are summarized as follows:

- When continuous time backoff is allowed, we propose a distributed algorithm which is provably throughput-optimal. For this distributed algorithm:

  - Each user only uses its local information (e.g., its queue backlog and channel condition). No explicit control messages are required among the users.
  - It is based on CSMA random access which similar to IEEE 802.11 and easy to implement.
  - We show that the proposed distributed algorithm can achieve the same performance as the centralized Max-Weight in terms of both throughput and delay.

- Then, we develop distributed algorithm for practical system where collision is possible. We show that the practical algorithm maintains throughput-optimality.
  - We analytically determine the overhead incurred under the practical algorithm by using the underlying Markov chain property.

7.2 Related Works

There has been a lot of interests in developing distributed scheduling algorithms. However, many aforementioned algorithms still require heavy message passing or computations. Therefore, it has been a research challenge to find simple random access achieving full optimality without any message passing. Besides the optimality, other important research challenges can be summarized as follows:

1. implementable, low-complex and requires only local information (no message passing).
2. good delay performance.
3. perform well over general channel conditions.

In the following, we classify the works along this direction. In [58], it was first shown that CSMA-based distributed algorithm which uses only local information, i.e., no message passing is provably optimal in terms of throughput and utility. However, the results in [58] assumed that the convergence time of the underlying Markov Chain to its steady-state is done at every time slot, which is called *time-scale separation* assumption. In practice, this is not a realistic assumption since the convergence of Markov chain requires some time depending on the network topology. In fact, CSMA algorithms with time scale separation assumption have been observed to have large delay that may grow with the network size, which cheapens the throughput-optimality because most applications in practice require some level of low delay. An idea to mitigate the effect of time-scale assumption was presented in [104] where the link weights are chosen to be a specific function of the queue lengths. However, the result in [104] requires the maximum queue length to be known by each user in the network, which is estimated via a distributed message-passing procedure. In [57] a queue length based distributed algorithm which can perform without message passing was proposed by assuming time scale separation.

The queue-based approaches without time-scale separation were presented in [105], [106], where the main idea is to choose specific functions which are slowly increasing function of queue lengths. However, the algorithm in [105] requires a slight message passing to broadcast the maximum queue-size over the network. In [106] using a certain distributed algorithm in which users estimate the maximum queue-size information without explicit message passing. The authors in [107] an algorithm without time scale separation assumption. This work updates CSMA parameters by dividing the time axis into frames. However, delay performance under the above throughput-optimal scheme can be very poor.

Regarding channel models, all aforementioned works assume that channel capacity is fixed. Wireless channels, however, are time-varying in practice, where the results on optimal CSMA may significantly change, since the time-varying fading creates significant variations on the CSMA parameters. In other words, these algorithms may not converge over fading channels. The first attempt to this problems was proposed in [108] where the authors proposed a distributed algorithm for fading channel. It was assumed that there are flows with short-term deadline constraints and long-term drop rate requirements. The authors did not investigate the network stability problem and throughput-optimal algorithm was not proposed. Further, the work in [108] assumes continuous backoff time which is not practical. In practice, the systems are actually discrete, where the systems evolve over discretized time slots (e.g., 20 μ sec in IEEE 802.11b) and collisions will inevitably occur, when two users contend at a same time slot. In [109] the authors consider a specific channel model where the channel capacity randomly varies between 0 and 1.

Unlike these works, we develop a provably throughput-optimal distributed algorithm which requires only local information (no message passing) and can perform over general fading channels for complete interference graphs where each link interferes with every other link. Remarkably, the performance of our algorithm in terms of both throughput and delay is the same as that of a centralized system when the arbitrary backoff time is allowed. Our algorithm is also optimal with discrete back time at the expense of low number of mini-slots for collision resolution.

7.3 System Model

We consider a typical IEEE 802.11x network where N users are transmitting to a single access point (AP) (or base station) over a single channel. Time is slotted, $t \in \{0, 1, 2, \ldots\}$, and wireless channel between the AP and each user is assumed to be independent across users and slots. The gain of the channel is constant over the duration of a time slot but varies between slots. We assume that if two or more users transmit at the same time, there will be a collision. A schedule is represented by a vector x in which the i^{th} element of x denoted by x_i is equal to 1 (i.e., $x_i = 1$) if user i is scheduled, otherwise $x_i = 0$. Let $\mathcal{S}$ be the set of all feasible schedules of the network.

Let $A_n(t)$ be the amount of data arriving into the queue of user n at time slot t. Let $Q_n(t)$ and $R_n(t)$ denote the queue length and transmission rate of user n at time t, respectively, and $\mathbf{Q}(t) = (Q_1(t), Q_2(t), \ldots, Q_N(t))$ be the queue length vector. The dynamics of queue length for node n is given as

$$Q_n(t+1) = [Q_n(t) + A_n(t) - R_n(t)x_n(t)]^+, \tag{7.1}$$

where $[y]^+ = \max(y, 0)$. We say that the system is stable if the mean queue length for all transmitting users is finite.

A scheduling algorithm is a procedure to decide which schedule to be used in every time slot for data transmission. In this work we focus on the MAC layer, and thus we only consider single-hop traffic. The rate region of the network is the set of all arrival rates λ for which there exists a scheduling algorithm that can stabilize the queues. Let Λ be the rate region of the network. We say that a scheduling algorithm is throughput-optimal or achieves the maximum throughput, if it can keep the network stable for all arrival rates in Λ.

We define *weighted-throughput* (or shortly weight) of each user as $w_n(t) = Q_n(t)R_n(t)$, and let $w_{max}(t)$ denote the maximum weighted-throughput at time slot t i.e., $w_{max}(t) = \max_n\{w_n(t)\}$. It is well know that the *centralized Max-Weight* (C-MW) algorithm is throughput-optimal and achieves largest rate region by always scheduling the user with maximum weighted-throughput at every time slot. However, C-MW requires queue size and channel state information from all users at every scheduling time. Our aim is to design a fully distributed algorithm whose performance in terms of rate region and delay is close to that of C-MW.

7.4 Idealized Distributed Algorithm, I-DALG

Here, we present our CSMA based randomized algorithm in which transmission probabilities of users are determined both considering their queue sizes and channel conditions at each time slot. The key idea is as follows: each user determines a randomized backoff time based on their weighted-throughput. Then, the user whose backoff time expires first is allowed to access to the channel. First, we consider an idealized CSMA algorithm (I-DALG) where the backoff time of users is continuous. Under I-DALG , at the beginning of each time slot, each user generates an exponentially distributed random variable denoted by X_n, which represents the randomized backoff time of user n, with rate $r_n(t) = f(w_n(t))$, where $f(v)$ is an increasing function of v. A user starts transmitting after this random duration unless it senses another transmission before.

Algorithm 3: Idealized CSMA based Distributed Algorithm, I-DALG

1: User n generates exponentially distributed random variable $X_n(t)$ with rate $f(w_n(t))$ and waits for $X_n(t)$ seconds.
2: If user n hears a INTENT message before its timer expires, sets $x_n(t) = 0$. Otherwise, $x_n(t) = 1$.
3: If $x_n(t) = 1$, user n transmits its data at time slot t.

Next, we prove that I-DALG is throughput-optimal under certain conditions. Note that since X_n's are exponential random variables, the following equation is true:

$$P(X_n(t) \leq X_k(t)) = \frac{f(w_n(t))}{f(w_n(t)) + f(w_k(t))}$$

for all $k \neq n$. Then, the successful transmission of user n at time slot t denoted as $\pi_n(t)$ can be given as follows:

$$\pi_n(t) = \frac{f(w_n(t))}{\sum_{i=1}^{N} f(w_i(t))}$$

The following theorem is useful to prove the throughput-optimality of I-DALG:

Theorem 7.1. *[40] (Theorem 1 (see Proposition 1 and Claim 1.)) For a scheduling algorithm, any given ϵ and δ satisfying $0 < \epsilon, \delta < 1$, we can find a constant $B(\epsilon, \delta) > 0$ such that: at any time slot t, with probability greater than $1-\delta$, the scheduling algorithm chooses a user n that satisfies*

$$\Pr\left[w_n(t) \geq (1-\epsilon) w_{max}(t)\right] \geq 1-\delta, \quad \textit{whenever} \parallel \mathbf{Q}(t) \parallel > B$$

Then the scheduling algorithm is throughput-optimal.

The scheduler that obtains $w_{max}(t)$ in each time slot t is usually centralized and computation-prohibitive but can lead to optimal throughput/delay. Theorem 7.1 states that the proposed scheduler can find a schedule using weight ϵ-close to $w_{max}(t)$ with high probability $1-\delta$ when weights $w_n(t)$ are large enough. Note that when ϵ and δ tend to 0, then the scheduling algorithm achieves the maximum throughout i.e., it is *throughput-optimal.*

Theorem 7.2. *I-DALG is throughput-optimal if $f(x) = b^x$ and $R_n(t) > R_{min}$ where $R_{min} > 0$ for all n, t and $b > 1$.*

Proof. We prove Theorem 7.2 by using Theorem 7.1. Let $w_{max}(t) = \max_{x \in \mathcal{S}} \sum_{n \in x} w_n(t)$ at time slot t. Given any $0 < \epsilon, \delta < 1$, we define

$$\chi = \{x \in \mathcal{S} : \quad \sum_{k \in x} w_k(t) < (1-\epsilon) w_{max}(t)\}$$

The probability of event χ is given by,

$$\pi(\chi) = \sum_{k \in \chi} \pi_k(t)$$

Now, in order to show that $\pi(\chi') \geq 1 - \delta$ which implies Theorem 1, we need to show that $\pi(\chi) < \delta$. Hence,

$$\begin{aligned} \pi(\chi) = \sum_{k \in \chi} \pi_k(t) &= \sum_{k \in \chi} \frac{b^{w_k(t)}}{\sum_{i=1}^{N} b^{w_i(t)}} \\ &\leq \frac{N b^{(1-\epsilon)w^*(t)}}{\sum_{i=1}^{N} b^{w_i(t)}} \end{aligned}$$

Note that

$$\sum_{i=1}^{N} b^{w_i(t)} \geq b^{w_{max}(t)}$$

Therefore, we have

$$\begin{aligned} \pi(\chi) &\leq \frac{N b^{(1-\epsilon)w_{max}(t)}}{\sum_{i=1}^{N} b^{w_i(t)}} \\ &\leq \frac{N b^{(1-\epsilon)w_{max}(t)}}{b^{w_{max}(t)}} \end{aligned}$$

Therefore if

$$w_{max}(t) > \frac{\log_b(N) + \log_b(\frac{1}{\delta})}{(\epsilon)} \tag{7.2}$$

then $\pi(\chi) < \delta$. Since $w_{max}(t)$ is a continuous nondecreasing function of $Q_n(t)$'s and $R_n(t)$ is arbitrarily large, with $\lim_{\mathbf{Q}(t)\to\infty} w_{max}(t) = \infty$, there exists $B > 0$ such that

$$w_{max}(t) > \max_n Q_n(t)R_{min} > \frac{\log_b(N) + \log_b(\frac{1}{\delta})}{(\epsilon)} \tag{7.3}$$

Hence, If queue sizes are large enough we can find a constant B.

$$\max_n Q_n(t) > \frac{\log_b(N) + \log_b(\frac{1}{\delta})}{(R_{min})\epsilon} \triangleq B \tag{7.4}$$

Hence, Theorem 7.1 holds and $\pi(\chi) < \delta$. Thus, I-DALG is throughput-optimal. $\square$

Remark: Theorem 7.2 establishes the fact that I-DALG selects the user with the weight-throughput close to maximum weighted-throughput with high probability when the maximum weighted-throughput $w_{max}(t)$ is large enough. In other words, the weight of users should be sufficiently high (i.e., ϵ and δ are sufficiently small) to achieve the throughput optimality. In fact, this requirement leads to large delays, and it is expected that the delay performance of I-DALG is poor. This fact also can be seen from the condition on $w_{max}(t)$ in (7.4), where $w_{max}(t)$ should be higher than a constant B which can be very large since ϵ and δ are sufficiently small. Note that under I-DALG B depends on the value of b which is the design parameter that enable us to reduce the delay. Specifically, if we choose b large, then B becomes small due to the term $\log_b(.)$. Hence, the condition in (7.4) can be satisfied with small queue sizes by I-DALG, which yields better delay performance. Moreover, we show next that by increasing b the delay performance of I-DALG can be pushed to the delay performance of C-MW. Before proving this result we explain it in an example: consider that there are two users with weights $w_1(t)$ and $w_2(t)$ at time slot t. Without loss of generality, assume that $w_1(t) > w_2(t)$ (i.e., user 1 has the maximum weight at time slot t), and user 1 must be scheduled according to C-MW. Let $\pi_1(t)$ and $\pi_2(t)$ be transmission probabilities of user 1 and user 2, respectively. With I-DALG, these probabilities are given by as follows:

$$\pi_1(t) = \frac{b^{w_1(t)}}{b^{w_1(t)} + b^{w_2(t)}},$$
$$\pi_2(t) = \frac{b^{w_2(t)}}{b^{w_1(t)} + b^{w_2(t)}}$$

Note that when the centralized Max-Weight algorithm is employed $\pi_1(t) = 1$ and $\pi_2(t) = 0$. Now, we allow that b becomes large, i.e., $b \to \infty$. Then, one can show that $\pi_1(t) \to 1$ and $\pi_2(t) \to 0$. Therefore, under I-DALG with large values of b, the user with maximum weighted throughput is scheduled with probability one, and I-DALG and C-MW becomes the same algorithm. Therefore, the delay performance of the network approaches to that of C-MW and throughput-optimality is still maintained.

Lemma 7.1. *As $b \to \infty$, I-DALG schedules the user with the maximum weighted throughout at every time slot with probability 1.*

Proof. First, without loss of generality, let n^* be the user which obtains $w_{max}(t)$ at time slot t. Then, user n^* transmits with the following probability at time slot t;

$$\pi_{n^*}(t) = \frac{b^{w_{n^*}(t)}}{\sum_{i=1}^{N} b^{w_i(t)}}$$

By diving both nominator and dominator by $b^{w_{n^*}(t)}$, we have,

$$\pi_{n^*}(t) = \frac{1}{1 + \sum_{i=1}^{N} b^{w_i(t) - w_{n^*}(t)}}, \forall i \neq n^*$$

We know that $w_{n^*}(t) > w_i(t)$ $\forall i \neq n^*$. Hence, taking $b \to \infty$ yields that

$$\lim_{b \to \infty} \pi_{n^*} = \frac{1}{1+0} = 1$$

One can also show that the success probability of user k, where $k \neq n^*$ goes to zero as $b \to \infty$. Then, user $k \neq n^*$ for all k transmits with the following probability;

$$\pi_k(t) = \frac{b^{w_k(t)}}{\sum_{i=1}^{N} b^{w_i(t)}}$$

By diving both nominator and dominator $b^{w_{n^*}(t)}$, we have,

$$\pi_k = \frac{b^{w_k(t)-w_{n^*}(t)}}{1 + \sum_{i=1}^{N} b^{w_i(t)-w_{n^*}(t)}}, k \neq n^*$$

Hence, taking $b \to \infty$ yields that

$$\lim_{b\to\infty} \pi_k = \frac{0}{1+0} = 0$$

Hence, I-DALG guarantees to schedule the user with maximum weighted throughput at every time slot. This completes the proof. □

Note that Lemma 7.1 states that I-DALG can achieve the performance of the centralized Max-Weight algorithm in terms of both rate region and delay by scheduling the user with the maximum weighted throughput at every time slot. In other words, with large value of b, I-DALG and the centralized MW are the same algorithms. Note that this is possible only when continuous backoff time is allowed. Specifically, I-DALG assumes that the sensing time is negligible, then with the continuous distributions of the backoff time, the probability for two conflicting users to start transmission at the same time is zero. So, collisions are ignored. In practice, backoff time is discrete and only takes a finite values. Hence, there is always a positive collision probability. Next, we turn our attention to the design of practical distributed algorithm which can perform well with discrete backoff time.

7.5 Practical and Throughput-Optimal Distributed Algorithm, P-DALG

Here, our aim is to develop a throughput-optimal algorithm that can be easily implemented in practical systems. In this regard, we divide each time slot into a control slot and a data slot. The control slot consists of mini-slots that enable to generate a collision-free data transmission as shown in Figure 7.1. Let $M(t)$ be the number of mini-slots used at time slot t. $M(t)$ is a random variable depending on the weighted

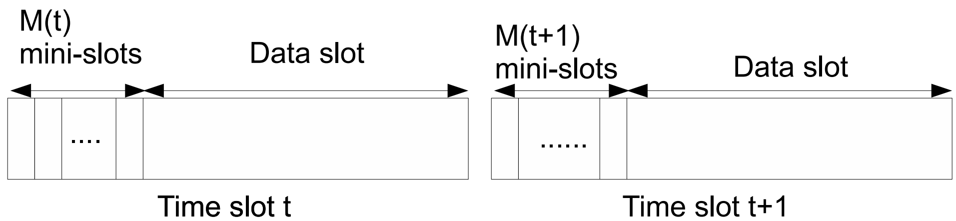

Figure 7.1: Mini-slots and data slot.

throughput of users and changes from time slot to time slot. We assume that all users are synchronized and start a data slot at the same time. A user can contend during a data slot if it has non-empty queue. If $Q_n(t) = 0$, then user n remains idle, skipping to the next slot. We define τ as the length of *virtual time*. Also, we define $a_n(m,t)$ is the attempt decision of user n at mini-slot m and time t. Then, P-DALG is implemented as follow: at each *mini-slot*, user n, $n \in \{1, 2, \cdots, N\}$, generates an exponential distributed random variable $X_n(t)$, with rate $r_n(t) = b^{w_n(t)}$. If $X_n(t)$ is less than τ, then user n attempts to transmit at time slot t, i.e., $a_n(m,t) = 1$, otherwise that user remains idle and waits for the next mini slot, i.e., $a_n(m,t) = 0$. If there are more than one users attempting at mini slot m, i.e., $\sum_{n=1}^{N} a_n(m,t) > 1$, then there will be a collision at that mini slot and users content in the next mini-slot. Similarly, if there is no user transmitting at a mini-slot, i.e., $\sum_{n=1}^{N} a_n(m,t) = 0$, then there will be an idle slot and users content in the next mini-slot. This process continuous until there is only one user attempting to transmit at a mini slot, i.e., $\sum_{n=1}^{N} a_n(m,t) = 1$.

Theorem 7.3. *P-DALG is throughput-optimal.*

Proof. Recall that I-DALG selects the user whose backoff time expires first. This is basically the main reason behind the throughput optimality of I-DALG. P-DALG waits until only a single user attempting to transmit. Hence, clearly, that user is the user whose backoff time expires first. Thus, like I-DALG, P-DALG schedules the user whose backoff time expires first and is throughput-optimal. □

Algorithm 4: Practical CSMA based Distributed Algorithm with fixed τ, P-DALG

- Step 1: At the beginning of each time slot, every user sets its virtual time to τ.
- Step 2: Then, each user generates exponential random variable $X_n(t)$ with rate, $r_n(t) = b^{w_n(t)}$, where $b > 1$.
- Step 3: If $X_n(t) < \tau$, the attempt decision $x_n(t) = 1$. Otherwise, $x_n(t) = 0$.
  - If there are more than one user attempting, i.e., $\sum_{n=1}^{N} a_n(m,t) > 1$, then there will be a collision at that mini slot. Then, go Step 2.
  - If $\sum_{n=1}^{N} a_n(m,t) = 0$, then that slot is an idle slot. Go Step 2.
- Step 4: If there is only one user attempting i.e., $\sum_{n=1}^{N} a_n(m,t) = 1$, then that user transmits at the data slot. Let user k is only user attempting at a given mini slot. Then, $x_k(t) = 1$ and $x_n(t) = 0$, for all $n \neq k$.

Note that unlike I-DALG, P-DALG requires some number of mini-slots for collision free transmission. Next, we determine the expected number of mini-slots required under P-DALG.

7.5.1 Expected number of mini-slot under P-DALG

The successful transmission during any mini-slot is given by,

$$P_{succ}(t) = \sum_{n=1}^{N} \left(1 - e^{-\tau r_n(t)}\right) \prod_{k \neq n} e^{-\tau r_k(t)}$$

Let $P_{coll}(t)$ and $P_{idle}(t)$ be the collision and idle probabilities at time slot t, respectively. Note that under P-DALG , $P_{succ}(t)$, $P_{coll}(t)$ and $P_{idle}(t)$ do not depend on m since τ and b do not change over mini-slots.

Lemma 7.2. *Under P-DALG, the expected number of mini slot,* $\mathbb{E}[M(t)]$, *is given by,*

$$\mathbb{E}[M(t)] = \frac{1}{P_{succ}(t)}.$$

Proof. Note that the probability that $M(t) = 1$ is equal to the probability of successful transmission at the first mini-slot,

$$\Pr[M(t) = 1] = P_{succ}(t)$$

If $M(t) = 2$, collision free transmission occurs at the second mini slot. That means the first mini slot is either a collision or idle slot. Thus,

$$\Pr[M(t) = 2] = (P_{coll}(t) + P_{idle}(t))P_{succ}(t)$$

By iterating, we have,

$$\Pr[M(t) = m] = (P_{coll}(t) + P_{idle}(t))^{m-1}P_{succ}(t)$$

and $\mathbb{E}[M(t)]$ is given as,

$$\mathbb{E}[M(t)] = \sum_{m=1}^{\infty} m(P_{coll}(t) + P_{idle}(t))^{m-1}P_{succ}(t)$$

Clearly, $M(t)$ has geometric distributed random variable with parameter $P_{succ}(t)$. Hence, $\mathbb{E}[M(t)]$ is given by,

$$\mathbb{E}[M(t)] = \frac{1}{P_{succ}(t)} \tag{7.5}$$

$\square$

Next, we turn our attention to the practical solutions to reduce $\mathbb{E}[M(t)]$ as low as possible. Note that when $\vec{w}(t) = (w_1(t, w_2(t), \cdots, w_N(t))$ is given $P_{succ}(t)$ is a function of only b and τ. Hence, we consider the following optimization problem:

$$\max_{b,\tau} \quad \sum_{n=1}^{N} \left(1 - e^{-\tau r_n(t)}\right) \prod_{k \neq n} e^{-\tau r_k(t)}$$

The following Lemmas state that in order to maximize $P_{succ}(t)$, b and τ must be optimized.

Lemma 7.3. *For given τ, as $\mathbf{Q}(t)$ increases (or $\vec{w}(t)$ increases), the optimal value of b maximizing $P_{succ}(t)$ goes 1, i.e., $b^* \rightarrow 1$.*

Proof. One can show that $P_{succ}(t)$ is a concave function of b by employing the second derivative test. Then the optimal value of b denoted by b^* can be determined by solving the following equation;

$$\begin{aligned}&\left(\sum_{n=1}^{N} e^{\tau r_n(t)}\right)\left(\sum_{n=1}^{N} w_n(t) r_n(t)\right) - \sum_{n=1}^{N} e^{\tau r_n(t)} w_n(t) r_n(t)\\&= N \sum_{n=1}^{N} w_n(t) r_n(t)\end{aligned}$$

Let us consider the worst case where $w_n(t) = w_{max}(t)$ for all n at time slot t, and $w_{max}(t) = \max_n\{w_n(t)\}$. Then, b^* can be given by,

$$b^* = \left[\frac{1}{\tau} \ln\left(\frac{N}{N-1}\right)\right]^{1/w_{max}(t)}$$

Clearly, as $w_{max}(t) \rightarrow \infty$ b^* goes 1. □

This result can be also observed intuitively as follows. As the queue sizes increase, the backoff time of users decreases, and the number of users attempting in a mini slot increases as well. As a result, collision probability increases. When τ is fixed and queue sizes are large, to mitigate the effect of large queue sizes, b^* must become small, i.e., $b^* \rightarrow 1$.

Lemma 7.4. *For given b, as $\mathbf{Q}(t)$ increases the optimal value of τ maximizing $P_{succ}(t)$ goes 0, i.e., $\tau^* \rightarrow 0$.*

Proof. One can show that $P_{succ}(t)$ is a concave function of τ by employing the second derivative test. Then, when $w_n(t) = w_{max}(t)$ for all n at time slot t the optimal value of τ denoted by τ^* can be given by,

$$\tau^* = \left[\frac{1}{b^{w_{max}(t)}} \ln\left(\frac{N}{N-1}\right)\right]$$

Clearly, as $w_{max}(t) \rightarrow \infty$ τ^* goes to zero, i.e., $\tau^* \rightarrow 0$. □

Lemma 7.3 points that when queue sizes are large b must be small to decrease $\mathbb{E}[M(t)]$. However, we know from Theorem 7.2 and Lemma 7.1 that as b increases the delay performance of the system improves. Hence, one can conclude that there is *trade-off* between $\mathbb{E}[M(t)]$ and the delay. On the other hand, Lemma 7.4 points that τ must be adjusted properly to decrease $\mathbb{E}[M(t)]$. However, in order to find the optimal τ, the global knowledge of the network is required, i.e., $w_n(t)$ for all n. Next, we propose a modified version of P-DALG which determines τ dynamically by changing its value according to collision or idle slot. To do that, unlike P-DALG we allow that the length of virtual time can depend on m. Hence, we use the notation of $\tau(m)$ to denote the length of virtual time at the beginning of mini-slot m. Specifically, at each mini-slot, user sends a short message to announce its intention for data transmission. If an idle slot occurs, users increase virtual time by a factor of $g(\alpha)$, where $g(.)$ is a increasing function of α and $\alpha > 1$. If there is a collision (i.e., if there is another user transmitting an INTENT message at the same mini-slot), then users decrease virtual time by a factor of $g(\alpha)$. This process continues until only one user attempts.

7.5.2 Modified P-DALG, MP-DALG

Algorithm 5: Modified P-DALG, MP-DALG

- Step 1: At the beginning of each time slot t, every user sets it virtual time to τ, i.e, $\tau(1) = \tau$ for all t.
- Step 2: Then, at each mini slot m, every user generates exponential random variable $X_n(t)$ with rate, $r_n(t) = b^{w_n(t)}$.
- Step 3: If $X_n(t) < \tau(m)$, $a_n(m,t) = 1$. Otherwise $a_n(m,t) = 0$.
  - If $\sum_{n=1}^{N} a_n(m,t) > 1$, there will be a collision at that mini slot. Then, update $\tau(m) \to \frac{\tau(m)}{g(\alpha)}$ where $g(\alpha) > 1$. Go Step 2.
  - If $\sum_{n=1}^{N} a_n(m,t) = 0$, update $\tau(m) \to g(\alpha)\tau(m)$. Go Step 2.
- Step 4: If $\sum_{n=1}^{N} a_n(m,t) = 1$, then that user transmits at time slot t.

Next, we investigate the overhead introduced in terms of expected mini slot when

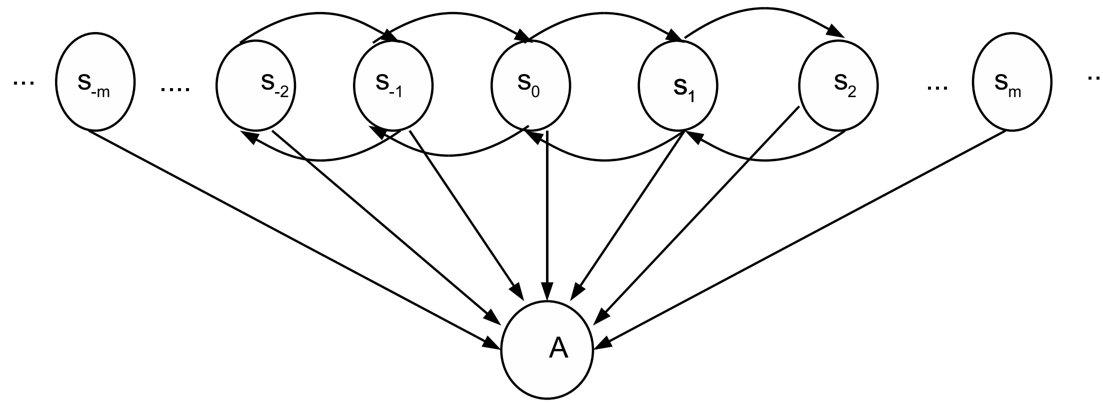

Figure 7.2: Markov Process.

MP-DALG is applied. Recall that virtual time $\tau(m)$ is decreased when a collision occurs at mini-slot m and is increased if no user transmits. Hence, this process can be explicitly given as follow:

$$\tau(m+1) = \begin{cases} \tau(m)/g(\alpha) & \text{; if collision at } \tau(m) \\ \tau(m)g(\alpha) & \text{; if idle at } \tau(m) \end{cases} \tag{7.6}$$

We will show that $\{\tau(m)\} \in \mathbb{R}_+$ is a Markov chain.

Corollary 7.1. *For every slot t, $\tau(m) \in \mathbb{R}_+$ is a Markov process with a countably infinite state space.*

Proof. Let Y_m be the random variable representing the length of virtual time at mini-slot m. Then,

$$\begin{aligned} P(Y_{m+1} = y | Y_1 = y_1, Y_2 = y_2, \cdots, Y_m = y_m) \\ = P(Y_{m+1} = y | Y_m = y_m) \end{aligned} \tag{7.7}$$

Hence, $\{Y_m\}$ is a Markov process. □

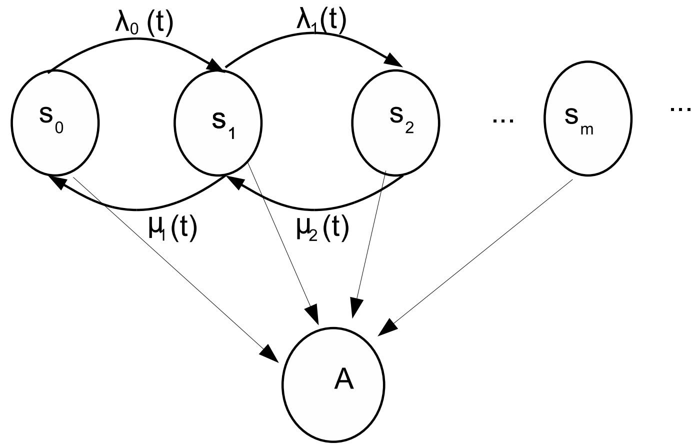

Figure 7.3: Markov Process for high loaded system.

We describe the Markov chain as follows. We have a set of states as follows,

$$S = \{\cdots, s_{-m}, s_{-1}, s_{-2}, s_0, s_1, s_2, \cdots, s_m, \cdots\}, \tag{7.8}$$

in which each state represents the length of virtual time. Specifically, at state s_m, the length of virtual time is equal to $\tau(m) = \frac{\tau}{g(\alpha)^m}$ whereas at state s_{-m}, it is equal to $\tau g(\alpha)^m$. We assume that at the beginning of each time slot t, the chain starts with $s_0 = \tau(0) = \tau$ for all t, and then moves from this state to other states, depending on collision or idle event. When only a single user attempts, the process reaches an end state which refers the absorption state denoted by A. This Markov Chain is shown in Figure 7.2.

For high loaded system and sufficiently large value of τ , the idle probability at the states $(\cdots, s_{-m}, \cdots, s_0)$ can be ignored. Then, for this system, the Markov chain is slightly changed as shown in Figure 7.3. Next, we determine the expected number of mini-slots used under MP-DALG for high loaded system, and we only consider the Markov chain in Figure 7.3. Let $\lambda_m(t)$ be the transition probability, which also represents the collision probability at state m, from state s_m to state s_{m+1}. Let $\mu_m(t)$ be the transition probability, which also refers the idle probability at state s_m, from state s_m

to s_{m-1}. Note that under this setup, the expected number of mini-slots is equal to the expected number of steps to the absorbtion state A, starting from state s_0. Let $\psi(m)$ be the expected number of steps from an arbitrary state m until absorbtion occurs. Then, by using first-step analysis we have a set of equations:

$$\begin{aligned}
\psi(0) &= 1 + \lambda_0 \psi(1), \\
\psi(1) &= 1 + \lambda_1 \psi(2) + \mu_1 \psi(0), \\
&\vdots \\
\psi(m) &= 1 + \lambda_m \psi(m+1) + \mu_m \psi(m-1) \\
&\vdots
\end{aligned}$$

We define T_∞ as expected number of steps for absorbtion within this infinite model. Let us consider a finite version of this Markov chain where the last state is denoted as s_m. Then, we have the following set of equations:

$$\begin{aligned}
\psi(0) &= 1 + \lambda_0 \psi(1), \\
\psi(1) &= 1 + \lambda_1 \psi(2) + \mu_1 \psi(0), \\
&\vdots \\
\psi(m) &= 1 + \lambda_m \psi(m) + \mu_m \psi(m-1)
\end{aligned}$$

Let the solution of this finite system be T_m. By using monotone convergence theorem, one can show that $\lim_{m\to\infty} T_m = T_\infty$. Hence, we next determine the expected number of mini-slot for finite linear system.

Lemma 7.5. *For high load and the number of user is large then* $\mathbb{E}[M(t)]$ *is upper bounded as:*

$$\mathbb{E}[M(t)] \leq \frac{1}{1-\lambda_0(t)} \tag{7.9}$$

Proof. Under high traffic load and the number of user is large, the idle probability can be ignored, i.e., $\mu_m = 0$ for all m. By using first-step analysis we have the following set of equations:

$$\begin{aligned}
\psi(0) &= 1 + \lambda_0 \psi(1), \\
\psi(1) &= 1 + \lambda_1 \psi(2), \\
&\vdots \\
\psi(m) &= 1 + \lambda_m \psi(m)
\end{aligned}$$

After rearranging the terms yields that,

$$\psi(0) = 1 + \sum_{s=1}^{m-1} \prod_{k=0}^{s-1} \lambda_k + \psi(m) \left(\prod_{i=1}^{m} \lambda_i \right)$$

Then, as $m \to \infty$,

$$\psi(0) = 1 + \lim_{m\to\infty} \sum_{s=1}^{m-1} \prod_{k=0}^{s-1} \lambda_k + \lim_{m\to\infty} \psi(m) \left(\prod_{i=1}^{m} \lambda_i \right)$$

Since $\lim_{m\to\infty} \lambda_m = 0$, the second limit goes to zero. Then, the expected number of step starting from state s_0 is given by

$$\mathbb{E}[M(t)] = 1 + \sum_{s=1}^{\infty} \prod_{k=0}^{s-1} \lambda_k(t). \tag{7.10}$$

Next, we show that the Markov chain is stable, and under MP-DALG the absorption eventually occurs by finding an finite upper bound on $\mathbb{E}[M(t)]$. Note that the maximum value of λ_m occurs at state 0, i.e., $\max_{k\in(0,1,\cdots,m)}\{\lambda_m\} = \lambda_0$. Hence,

$$\begin{aligned}
\mathbb{E}[M(t)] &\leq 1 + \sum_{s=1}^{\infty} \prod_{k=0}^{s-1} \lambda_0(t) \\
&= \frac{1}{1-\lambda_0(t)}
\end{aligned}$$

□

7.5.3 Improved MP-DALG

Note that MP-DALG algorithm sets the value of τ at the beginning of each time slot, i.e., $\tau(1) = \tau$ for all t. Depending on collision and idle event, τ decreases or increases, and eventually only a single user attempts at a mini slot (e.g., absorption occurs). Unlike MP-DALG, we now allow that τ depends on time t, hence, we use the notation $\tau(m,t)$ to denote that the value of virtual time at mini slot m and time slot t. This motivation is explained as follows: Let $m^*(t)$ be the state at which absorption occurs at time slot t with MP-DALG. Then, the corresponding virtual time is equal to $\tau(m^*)$. Let us assume that in the next slot $t+1$, the network dynamics changes slowly or never changes (e.g., channel states and queue process change slowly from time slot t to time slot $t+1$). Then, the state at which absorption occurs at time $t+1$, $m^*(t+1)$, will be at the vicinity of the state $m^*(t)$. In other words, $\tau(m^*, t+1)$ will be close to $\tau(m^*, t)$ with high probability. With this observation, we now present a modified version of MP-DALG namely MP-DALG2 as follows:

Algorithm 6: Improved MP-DALG

- Step 0: At time slot $t = 1$ and $m = 1$, every user sets it virtual time to τ.
- Step 1: At time slot t user n does:
  - Step 2: At mini-slot m, generate exponential random variable $X_n(t)$ with rate $r_n(t) = b^{w_n(t)}$.
  - Step 3: If $X_n(t) < \tau(m,t)$, $a_n(m,t) = 1$. Otherwise $a_n(m,t) = 0$.
    * If $\sum_{n=1}^{N} a_n(m,t) > 1$, there will be a collision at that mini slot. Then,
    * $m \leftarrow m + 1$,
    * $\tau(m,t) \leftarrow \frac{\tau(m,t)}{g(\alpha)}$. Go Step 2.
    * If $\sum_{n=1}^{N} a_n(m,t) = 0$, update
    * $m \leftarrow m + 1$, and $\tau(m,t) \leftarrow g(\alpha)\tau(m,t)$. Go Step 2.
  - Step 3: If $\sum_{n=1}^{N} a_n(m,t) = 1$, then that user transmits at time slot t. Let $m^*(t)$ be the state at which absorbtion occurs at time t.
- Step 4: $t \leftarrow t + 1$ and $\tau(1, t+1) = \tau(m^*, t)$. Go Step 1.

Note that like MP-DALG, MP-DALG2 is also throughput optimal. The expected

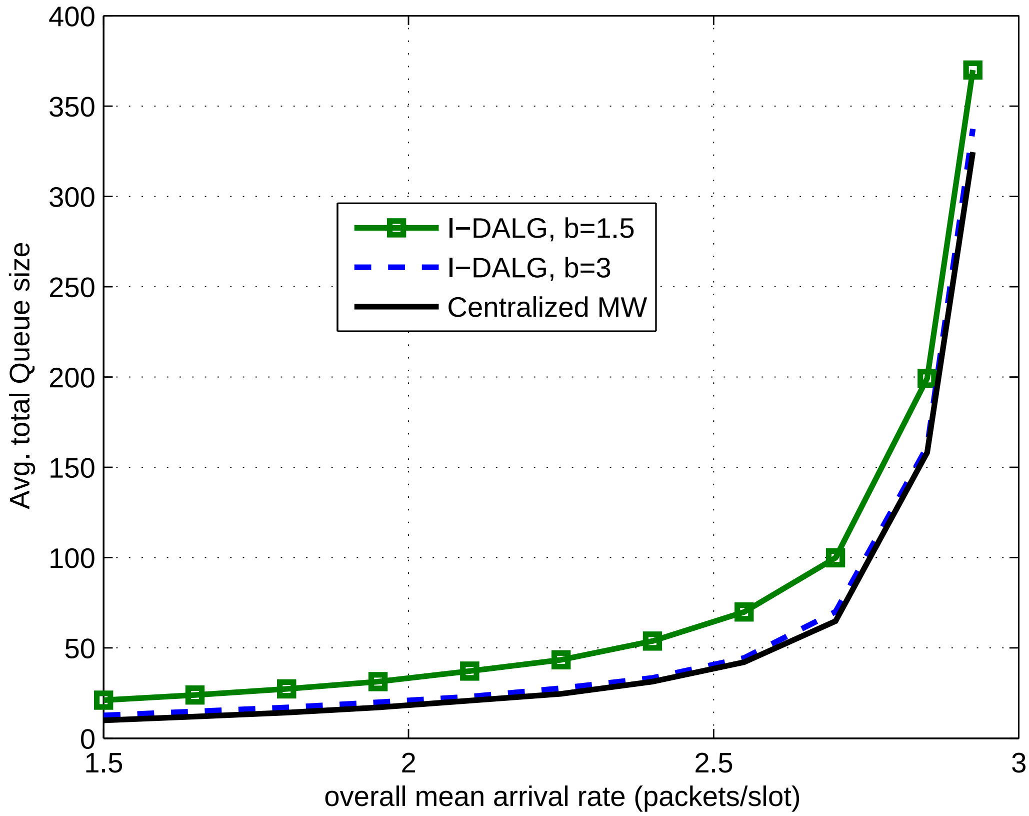

Figure 7.4: Average total queue sizes vs. overall mean arrival rate.

number of mini-slots used under MP-DALG2 can be determined by modeling the problem as an *embedded* Markov chain with absorption. However, the analysis is difficult and hence, we present the performance of MP-DALG2 in terms of expected mini-slot in our simulations.

7.6 Numerical Results

In our simulations, we model a single cell CDMA downlink transmission utilizing high data rate (HDR). The base station serves 10 users and keeps a separate queue for each user. Time is slotted with length $T_s = 1.67$ ms. Packets arrive at each slot according to Poisson distribution for each users with mean λ_n. The size of a packet is set to 128 bytes which corresponds to the size of an HDR packet. Each channel has 5 possible states with rates as given in Table 7.1. We compare I-DALG algorithm with the

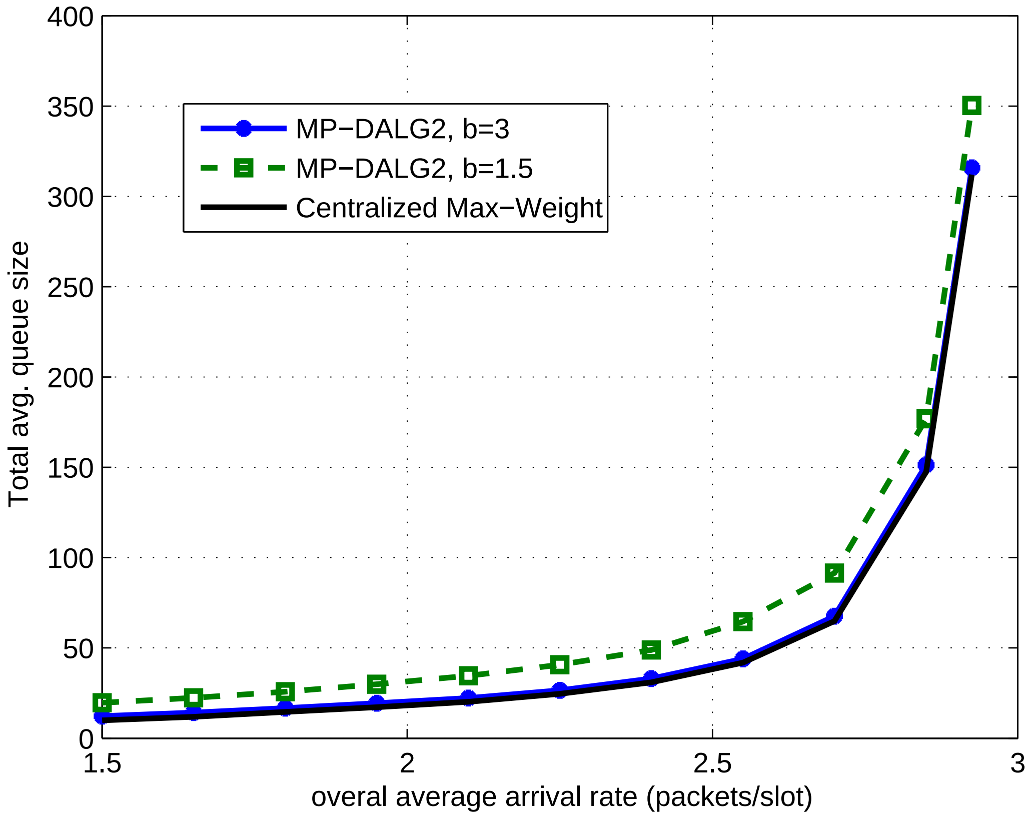

Figure 7.5: Performance of MP-DALG2

Table 7.1: Possible Physical Rates

Rates	r_1	r_2	r_3	r_4	r_5
kb/s	1843.2	1228.8	614.4	307.2	76.8

centralized Max-Weight algorithm in terms of rate region and delay. Figure 7.4 depicts the average total queue sizes in terms of packets vs. the overall arrival rate when $b = 1.5$ and $b = 3$. Note that the average queue size is an indicator to the average delay experienced by the users. Also note that by inspecting the average rate of change of queue sizes, we can approximately determine the maximum arrival rates that can be supported by different algorithms. Clearly, from Figure 7.4, the maximum supportable arrival rate is around the point where total arrival rate is 2.85 packets/slot. For all values of b, I-DALG can stabilize the network since the queue sizes increase suddenly, which indicates the instability, at the same point for all values of b, which shows that

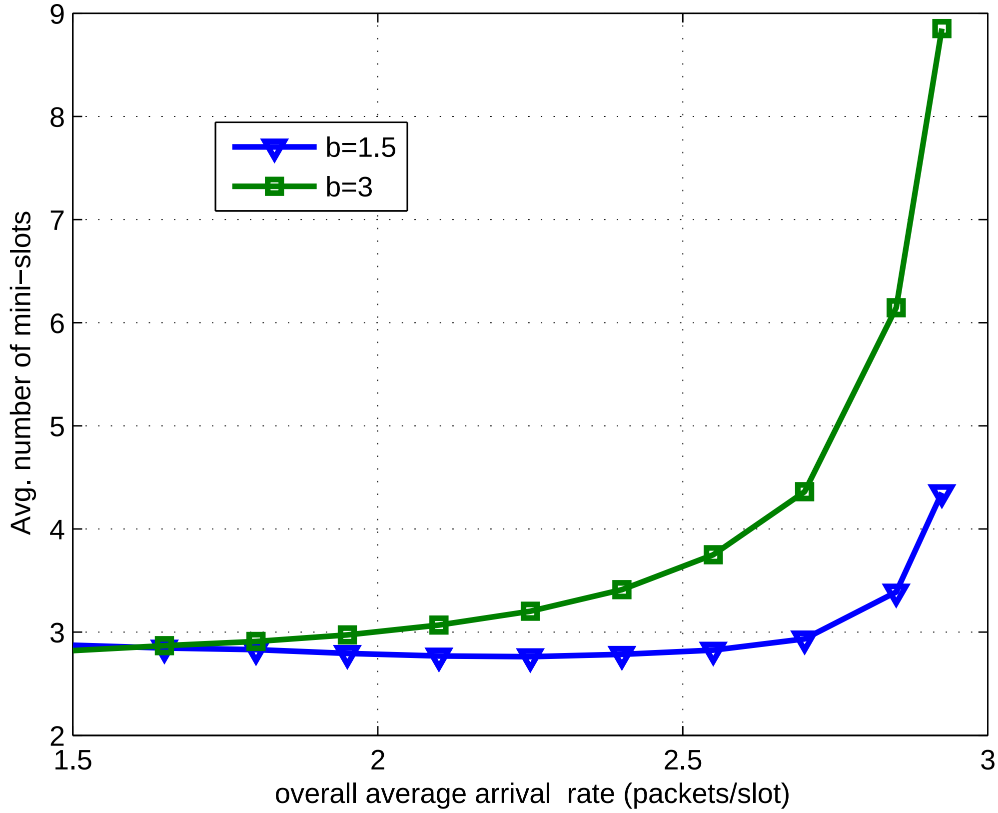

Figure 7.6: Average number of mini-slots vs. overall mean arrival rate.

I-DALG is throughput-optimal. Although the optimality of I-DALG is maintained for all b, the delay performance is different. Clearly, as b increases the delay performance of I-DALG is getting close to that of C-MW which has the optimal delay performance. This is because the probability that the user with the maximum weighted throughput is scheduled increases, and goes to one as b goes to infinity. Next, we investigate the performance of MP-DALG2 in terms of rate region and average delay. In Figure 7.5 we plot the mean total queue backlog summed over all users of the network, as the offered load λ increases when $b = 3$, $b = 1.5$ and $\alpha = 2$. When λ approaches a certain limit, the average total backlog will increase to infinity. This limit can then be viewed as the boundary of the rate region. Clearly this limit is same for centralized Max-Weight and MP-DALG2 with $b = 3$ and $b = 1.5$. Hence, for both values of b MP-DALG2 is throughput-optimal. However, the average delay performance of MP-DALG2 with $b = 3$ is better than that of MP-DALG2 with $b = 1.5$, which confirms our theoretical results

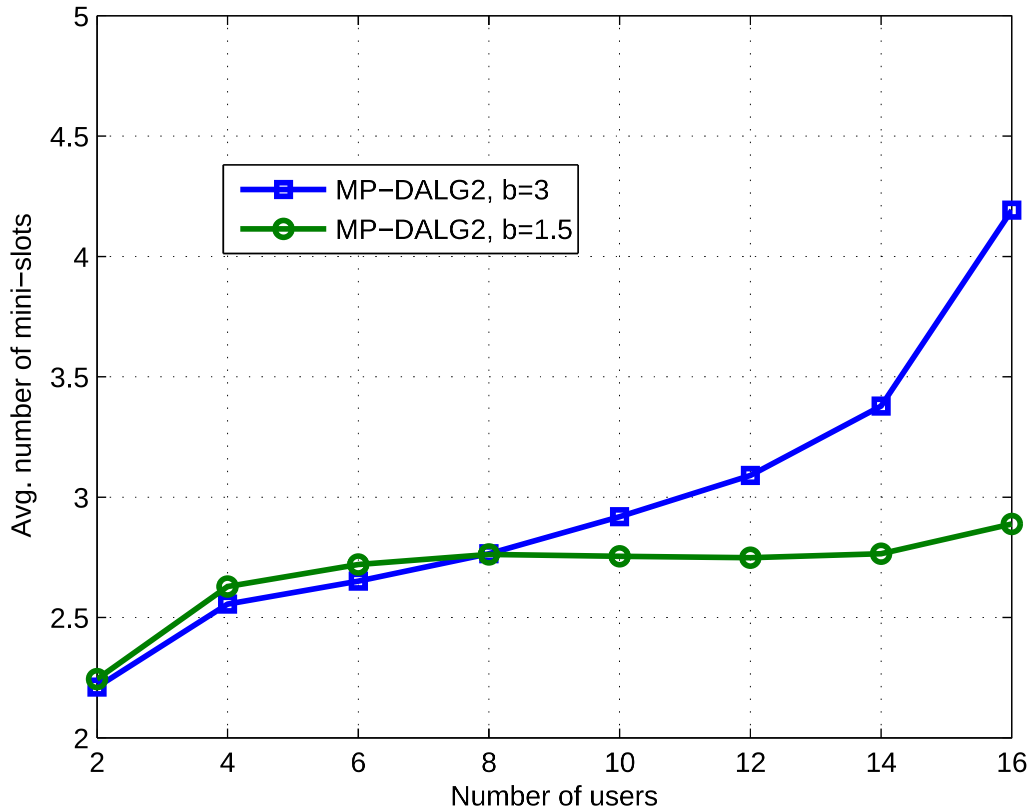

Figure 7.7: Average number of mini-slots vs. b.

in Theorem 3. On the other hand, from Figure 7.6 we can see that the expected number of mini-slots required by MP-DALG2 with $b = 1.5$ is less than that of one when $b = 3$. b actually controls the tradeoff between the average delay and the expected number of mini-slots. The penalty is decreasing with b and the queuing backlog is increasing with b. If more emphasis is placed for minimizing $\mathbb{E}[M(t)]$, we should choose a smaller b. If more strict delay is required, we should chose a larger b.

Lastly, we investigate the performance of MP-DALG2 in terms of required mini slot vs. number of users for different values of b. We set the overall mean arrival rate to 2.55 packets/slot, which is close to the boundary of rate region. In Figure 7.7 we plot the average number of mini-slots under MP-DALG2 vs. number of users in the system. When $b = 1.5$, as N increases, $\mathbb{E}[M(t)]$ stays almost constant around 2.7. When we increase b to 3, $\mathbb{E}[M(t)]$ increases as well. However, the overall required mini-slot is less than 5 in both cases.

7.7 Chapter Summary

In this Chapter, we have proposed two new distributed scheduling policies for a fully-connected wireless network over fading channels. For the first algorithm by allowing arbitrary backoff time we have developed fully distributed algorithm which achieves the same performance as that of centralized Max-Weight algorithm in terms of rate region and delay. We then have developed more practical distributed algorithm that can performs with discrete backoff time. For this algorithm we have quantified the expected number of mini-slots required for collision resolution. We also showed that we were able to make an intelligent trade off between having a low number of mini-slots and minimizing average delay by tuning system parameter b.

Chapter 8

Conclusions and Future Works

This thesis has studied several important scheduling problems in wireless networks with time-varying channel conditions, random packets arrivals, and different user constraints. Specifically, scheduling algorithms developed in the first part of this thesis including Chapter 3 and Chapter 4 contribute to the possibility of further improvement of wireless network performance. The algorithms in Chapter 5, Chapter 6 and Chapter 7 serve to highlight the value of network state information for scheduling in wireless systems.

In this thesis, we have studied opportunistic scheduling for time-slotted systems. When complete network state information including channel condition and queue length information of all users are available at the scheduler the performance of a network can be further increased with efficient communication techniques. In that sense, we first investigated a cognitive radio network with primary and secondary users where the primary users are licensed owners of spectrum while the secondary users do no have any such licensed spectrum. We further incorporated a cooperation between primary and secondary users by using a time-sharing policy. Through Lyapunov optimization technique, we developed optimal scheduling policies that achieve maximum utility and provide fairness among primary and secondary users. By considering a cognitive radio network with a possible cooperation between primary and secondary users, unused portion of spectrum can be utilized more efficiently, and hence we can improve spectrum efficiency. Another solution to improve spectrum efficiency of a wireless system with complete network state information is to allow more than one packets transmission in a given time slot instead of single packet transmission. We then developed throughput-optimal scheduling algorithm which provides multi-packet transmission by employing a specific modulation at the physical layer. By using Lyapunov analysis, we showed that the resulting rate region of developed policy is larger than that of single user scheduling. The results in Chapter 3 and Chapter 4 suggest that when complete network information is available at the scheduler, simple opportunistic scheduling algorithms with key communication techniques can provide better spectrum efficiency.

In the second part of this thesis including Chapter 5, Chapter 6 and Chapter 7 the first thing that we leant is that the network state information for opportunistic scheduling is a key component for developing optimal policies. However, obtaining it is very costly. By being motivated this fact, we studied this problem for both downlink and uplink network where channel state information is not immediately available at the base station but it has to be acquired from users. For the downlink system, we developed different scheduling algorithm that can perform without complete channel state information at the base station over different channel conditions. Specifically, we first considered a channel model where channel gains vary independently over time slots. Under this model, we developed a scheduling policy that can be easily implemented in practical systems. Then, we considered a more general channel model where channels gains continuously vary over slot and the channel process can be non-stationary. Within this channel model, we showed that a learning based scheduling algorithm can efficiently perform with limited channel state information. The results in Chapter 5 and Chapter 6 show that efficient scheduling algorithm can be designed with careful consideration of channel characteristics together with users traffic loads. For uplink network, we observed that the cost associated for acquiring channel state and queue length information of users is much higher than that of a downlink system. Hence, we designed a fully distributed algorithm that does not require any message passing between users. We showed that the proposed distributed algorithm is throughput-optimal and it achieves the same rate region and delay performance as that of centralized scheduling algorithm when continuous backoff is available. When discrete backoff time is available, the proposed policy can be implemented in a distributed manner with low overhead.

Before concluding this thesis, we will briefly present some ideas which will motivate future studies on this topic. In almost all the analysis in Chapter 3, we assumed that the feasibility region is known in advance. In other words, the supportable minimum rate requirements are known. However, in practice, the scheduler cannot determine whether a rate requirement can be supportable by the network or not beforehand and hence it has to be determined or estimated at least. It would be interesting to characterize the feasibility region. Also, we note that the complexity of the scheduling algorithms increases as the number of primary and secondary users increases. Hence, designing distributed algorithms will be another research direction along this topic.

In the second part of this thesis we have focused on the design of scheduling algorithms without complete network information. In Chapter 5 and Chapter 6, we have studied such scheduling problems with different assumptions on channel model. Specifically, In Chapter 5, the analytical and simulation results are given by assuming channels are i.i.d. However, the analytical result may not be directly applied for correlated channel. As a future work, we will provide the result by considering more general channel models, i.e., correlated or Markovian channels. Similarly, in Chapter 6 the performance gain is analyzed for a special case where there are two users. For a general case, the problem is not easy to solve since we need to know estimation statistics. Hence, the general case will be very challenging, and we believe it is an interesting problem to be addressed in the future. Also, we want to extend our works for the multichannel wireless system, i.e., OFDM networks with considering fairness among users. Lastly, we have shown that the proposed distributed algorithms in Chapter 7 operate under full connected network topologies where users can hear each other. As a future work, we will explore scheduling algorithms for more general network topologies. Also, we assumed that the network operator can allocate as many as mini-slots, which is not always the case in practical wireless networks. We will relax this assumption in our future work.

Bibliography

[1] 3GPP, "3GPP TS 36.300-870," in *3rd Generation Partnership Project.*

[2] IEEE, "IEEE std 802.16-2009," in *IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems*, 2009.

[3] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, no. 10, pp. 1374–1396, Oct. 1995.

[4] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi, "CDMA/HDR: a bandwidth efficient high speed wireless data service for nomadic users," *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70–77, 2000.

[5] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech.*, vol. 28, no. 3, pp. 565–715, 1949.

[6] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[7] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 30, pp. 466–478, 1993.

[8] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks.* Foundations and Trends in Networking, 2006.

[9] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Comput. Netw.*, vol. 41, 2003.

[10] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *Proc. IEEE INFOCOM'03*, vol. 2, 2003, pp. 1106–1115.

[11] S. Borst and P. Whiting, "Dynamic rate control algorithms for HDR throughput optimization," in *Proc. IEEE INFOCOM'01.*

[12] R. Agrawal, A. Bedekar, R. J. La, and V. Subramanian, "Class and channel condition based weighted proportional fair scheduler." in *In International Test Conference, 2001.*

[13] H. Kushner and P. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

[14] F. Kelly, A.Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *Journ. of the Operational Res. Society*, vol. 49, no. 3, pp. 237–252, 1998.

[15] S. H. Low and D. E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–875, Dec. 1999.

[16] M. Chiang, "Balancing transport and physical layer in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.

[17] A. R. C. M. Chiang, S. H. Low and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures." *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.

[18] C. Li and X.Wang, "Adaptive opportunistic fair scheduling over multiuser spatial channels," vol. 53, no. 10, pp. 1708–1717, Apr. 2005.

[19] F. C. Commission, "Notice of proposed rulemaking, in the matter of facilitating opportunities for flexible, efficient and reliable spectrum use employing cognitive radio technologies (et docket no. 03-108) and authorization and use of software defined radios (et docket no. 00-47), fcc 03-322," Dec. 2003.

[20] ——, "Spectrum policy task force report," Nov. 2002.

[21] J. M. Peha, "Approaches to spectrum sharing," *IEEE Commun. Mag.*, vol. 43, no. 2, pp. 10–12, Feb. 2005.

[22] J. O. Neel, "Analysis and design of cognitive radio networks and distributed radio resource management algorithms," PhD Dissertation, Virginia Polytechnic Institute, Sep. 2006.

[23] I. F. Akyildiz, W. Y. Lee, M. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[24] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-part I: System description," vol. 51, pp. 1927–1938, Nov. 2003.

[25] ——, "User cooperation diversity-part II:implementation aspects and performance analysis," vol. 51, pp. 1939–1948, Nov. 2003.

[26] T. M. Cover and A. A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. 25, pp. 572–584, 1979.

[27] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3037–3063, 2005.

[28] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 203–213, Jan. 2008.

[29] S. K. Jayaweera and T. Li, "Dynamic spectrum leasing in cognitive radio networks via primary-secondary user power control games," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3300–3310, 2009.

[30] S. K. Jayaweera, G. Vazquez-Vilar, and C. Mosquera, "Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2328–2339, 2010.

[31] H. I. Kim, "Two-best user scheduling for high-speed downlink multicode CDMA with code constraint," in *Proc. IEEE Conf. Global Commun. (Globecomm'04)*, pp. 2659–2663.

[32] H. Jiang and P. A. Wilford, "A hierarchical modulation for upgrading digital broadcast systems," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 223–229, 2005.

[33] A. Jdidi, T. Chahed, S.-E. Elayoubi, and H. Besbes, "On hierarchical modulation to increase flow-level capacity in OFDMA-based networks," in *In IEEE ICC*, May 2010.

[34] A. Jdidi and T. Chahed, "Impact of hierarchical modulation on proportional fair in OFDMA-based networks," in *In ComNet*, 2010.

[35] T. Cover, "Broadcast channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, 1972.

[36] M. J. Hosssain, M.-S. Alouini, and V. K. Bhargava, "Channel access delay and buffer distribution of two-user opportunistic scheduling schemes in wireless networks," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 2077–2087, 2010.

[37] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proceedings of IEEE INFOCOM 2009 Mini-Conference,*, 2009.

[38] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.

[39] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for timevarying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[40] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 411–424, 2005.

[41] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[42] ——, "Optimal energy and delay tradeoffs for multiuser wireless downlinks," *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3095–3113, 2007.

[43] ——, "Super-fast delay trade-offs for utility optimal fair scheduling in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.

[44] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[45] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. IEEE INFOCOM'05*, vol. 3, 2005, pp. 1794–1803.

[46] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE INFOCOM'05*, vol. 3, Mar. 2005, pp. 1723–1734.

[47] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a timevarying channel: The exponential rule," *American Mathematical Society Translations*, vol. 207, 2002.

[48] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *INFOCOM*, vol. 6, Mar. 2004, pp. 1662–1672.

[49] J. Huang, V. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink ofdm systems for broadband wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 5226–234, Feb. 2009.

[50] V. Hassel, D. Gesbert, M.-S. Alouini, and G. Oien, “A threshold-based channel state feedback algorithm for modern cellular systems,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2422–2426, 2007.

[51] T. Tang and R. Heath, “Opportunistic feedback for downlink multiuser diversity,” *IEEE Commun. Lett.*, vol. 9, no. 10, pp. 948–950, 2005.

[52] T. Samarasinghe, H. Inaltekin, and J. Evans, “The feedback-capacity tradeoff for opportunistic beamforming,” in *ICC*, 2011.

[53] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, “An overview of limited feedback in wireless communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[54] K. Kar, X. Luo, and S. Sarkar, “Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements,” in *INFOCOM*, 2007.

[55] L. Tassiulas, “Linear complexity algorithms for maximum throughput in radio networks and input queued switches,” in *Proc. IEEE INFOCOM’98*, 1998.

[56] E. Modiano, D. Shah, and G. Zussman, “Maximizing throughput in wireless networks via gossiping,” in *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, 2006, pp. 27–38.

[57] J. Ni, B. Tan, and R. Sarikant, “Q-CSMA: Queue length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” in *INFOCOM’10*, 2010.

[58] L. Jiang and J. Walrand, “A CSMA distributed algorithm for throughput and utility maximization in wireless networks,” in *Allerton 2008*, 2008.

[59] ——, “Approaching throughput-optimality in distributed CSMA scheduling algorithms with collisions,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 816–829, 2011.

[60] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: Making software radios more personal," *IEEE Personal Commun. Mag.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[61] S. Shakkottai and A. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in HDR." in *Bell Laboratories Technical Report*, 2000.

[62] I. H. Hou and P. R. Kumar, "Utility-optimal scheduling in time-varying wireless networks with delay constraints," in *Mobihoc*, 2010, pp. 31–40.

[63] A. Bhorkar, A. Karandikar, and V. Borka, "Power optimal opportunistic scheduling," in *Proc. IEEE GLOBECOM'06*, 2006.

[64] M. Xiao, N. B. Shroff, and E. Chong, "Utility-based power control in cellular wireless systems," in *Proc. IEEE INFOCOM'01*.

[65] R. Urgaonkar and M. J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 766–777, Jun. 2009.

[66] M. Lotfinezhad, B. Liang, and E. S. Sousa, "Optimal control of constrained cognitive radio networks with dynamic population size," in *Proc. IEEE INFOCOM'10*, Mar. 2010, pp. 1–9.

[67] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[68] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[69] J. Schiller, *Mobile Communication*. Addison-Wesley Press, 2000.

[70] S. Boyd and L. Vandenbeghe, *Convex Optimization*. Cambridge University Press, 2004.

[71] M. Karaca and O. Ercetin, "Optimal scheduling and resource allocation using hierarchical modulation in wireless networks," in *Proceedings of the IEEE/IFIP Eighth International Conference on Wireless and Optical Communications Networks (WOCN 2011)*, May 2011.

[72] B. Liu and H. Liu, "DPC-based hierarchical broadcasting: Design and implementation," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3895–3900, Nov. 2008.

[73] A. Gopalan, C. Caramanis, and S. Shakkottai, "On wireless scheduling with partial channel-state information," in *Proc. of the 45th Allerton Conference*, 2007.

[74] ——, "Low-delay wireless scheduling with partial channel-state information," in *INFOCOM*, 2012.

[75] M. Ouyang and L. Ying, "On optimal feedback allocation in multichannel wireless downlinks," in *Mobihoc*, 2011.

[76] C.-P. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 4, pp. 527–539, Apr. 2010.

[77] M. J. Neely, "Max weight learning algorithms with application to scheduling in unknown environments," in *arXiv:0902.0630v1*, Feb. 2009.

[78] W. Ouyang, S. Murugesan, A. Eryilmaz, and N. B. Shroff, "Scheduling with rate adaptation under incomplete knowledge of channel/estimator statistics," in *48th Allerton Conference on Communication, Control, and Computing*, 2010.

[79] ——, "Exploiting channel memory for joint estimation and scheduling in downlink networks," in *INFOCOM*, 2011.

[80] C. Tekin and M. Liu, "Approximately optimal adaptive learning in opportunistic spectrum access," in *INCOFOM*, 2012.

[81] B. Radunovic, A. Proutiere, D. Gunawardena, and P. Key, *Exploiting Channel Diversity in White Spaces.* Technical Report MSR-TR-2011-53, 2011.

[82] N. Chang and M. Liu, "Optimal channel probing and transmission scheduling for opportunistic spectrum access." in *Mobicom*, 2007.

[83] P. Chaporkar, A. Proutiere, H. Asnani, and A. Karandikar, "Scheduling with limited information in wireless systems," in *Mobihoc*, 2009.

[84] J. G. Andrews, S. Shakkottai, R. Heath, N. Jindal, M. Haenggi, R. Berry, D. Guo, M. Neely, S. Weber, S. Jafar, and A. Yener, "Rethinking information theory for mobile ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 12, pp. 94–101, Dec. 2008.

[85] T. Alpcan, "A framework for optimization under limited information," in *5th Intl. Conf. on Performance Evaluation Methodologies and Tools (ValueTools)*, 2011.

[86] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.

[87] A. Duel-Hallen, "Fading channel prediction and estimation for mobile radio adaptive transmission systems," *Proc. of IEEE*, vol. 95, no. 12, pp. 2299–2313, 2007.

[88] M. Karaca, Y. Sarikaya, O. Ercetin, T. Alpcan, and H. Boche, "Throughput optimal scheduling with dynamic channel feedback," in *PIMRC'12 Workshop-WDN*, 2012.

[89] M. Karaca, T. Alpcan, and O. Ercetin, "Smart scheduling and feedback allocation over non-stationary wireless channels," in *ICC'12 WS - SCPA*, 2012.

[90] A. Duel-Hallen and H. H. S. Hu, "Long-range prediction of fading signals: Enabling adaptive transmission for mobile radio channels," *IEEE Signal Process. Mag.*, vol. 17, no. 3, pp. 62–75, 2000.

[91] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in Neural Information Processing Systems*, 1994.

[92] F. Fu and M. van der Schaar, "Learning to compete for resources in wireless stochastic games," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1904–1919, May 2009.

[93] M. Bennis and D. Niyato, "A Q-learning based approach to interference avoidance in self-organized femtocell networks," in *IEEE FEMnet (co-located with IEEE GLOBECOM)*, Dec. 2010.

[94] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5667–5681, Nov. 2010.

[95] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed banditswith linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, 2012.

[96] K. Liu, Q. Zhao, and B. Krishnamachari, "Dynamic multichannel access with imperfect channel state detection," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2795–2808, May 2010.

[97] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A gaussian process positioning system for cellular networks," in *Advances in Neural Information Processing Systems*, 2004.

[98] R. S. Sutton, A. Koop, and D. Silver, "On the role of tracking in stationary environments," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007.

[99] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, New York, 2006.

[100] R. T. Marler and J. S. Arora, *Survey of multi-objective optimization methods for engineering.* Structural and Multidisciplinary Optimization, 2004.

[101] C. E. Rasmussen, *Evaluation of gaussian processes and other methods for non-linear regression.* PhD thesis, University of Toronto, 1996.

[102] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications.* The MIT Press, 1949.

[103] W. C. Jakes, *Microwave Mobile Communications.* Piscataway NJ IEEE Press, 1994.

[104] S. Rajagopalan, D. Shah, and J. Shin, "Aloha that works," *Submitted*, 2008.

[105] ——, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proceedings of ACM Sigmetrics*, 2009.

[106] D. Shah and J. Shin, "Randomized scheduling algorithm for queueing networks," vol. 22, pp. 128–171, 2012.

[107] A. Proutiere, Y. Yi, T. Lan, and M. Chiang, "Resource allocation over network dynamics without timescale separation," in *INFOCOM*, 2010.

[108] B. Li and A. Eryilmaz, "A fast-CSMA algorithm for deadline-constrained scheduling over wireless fading channels," in *RAWNET 2011*, 2011.

[109] S.-Y. Yun, J. Shin, and Y. Yi, "Medium access over time-varying channels with limited sensing cost," in *ArXiv e-prints*, 2012.