

# Online Algorithms for Delay Constrained Scheduling over a Fading Channel

A thesis submitted in partial fulfillment of  
the requirements for the degree of

Doctor of Philosophy

by

**Nitin Salodkar**  
(Roll No. 02429003)

Advisor: Prof. Abhay Karandikar



Department of Computer Science & Engineering

Indian Institute of Technology Bombay

Powai Mumbai 400076

May 2008



To my parents (Sau. Usha and Shri Dileep Salodkar),  
sister (Sayali) and wife (Devyani)



# Indian Institute of Technology Bombay

## Certificate of Course Work

This is to certify that **Nitin Salodkar** (Roll No. 02429003) was admitted to the candidacy of Ph.D. degree in January 2004, after successfully completing all the courses required for the Ph.D. programme. The details of the course work done are given below.

S.No	Course Code	Course Name	Credits
1	EE 764	Wireless and Mobile Communications	6
2	EE 706	Communication Networks	6
3	IT 612	Wireless Local Area Networks	6
4	EE 659	Linear and Nonlinear Optimization	6
5	EE 703	Digital Message Transmission	6
6	CS 681	Performance Analysis of Computer Systems and Network	6
7	ITS801	Seminar	4
8	IT 690	Mini Project	10
		<b>Total Credits</b>	<b>50</b>

IIT Bombay

Date:

Dy. Registrar (Academic)



# Abstract

In this thesis, we consider the problem of delay constrained scheduling over a fading wireless channel. We design cross layer scheduling algorithms that optimize various quantities such as power expenditure and throughput while satisfying the delay constraints. We formulate the cross layer scheduling problem as a multistage optimization problem. A well known solution technique for solving such a problem is to cast it as a Markov Decision Process (MDP) and then to utilize the traditional MDP solution techniques such as Linear Programming (LP) or other iterative techniques such as value iteration for determining the optimal policy. These techniques are computationally infeasible for a large state space. Moreover they require a knowledge of the transition probability mechanism of the underlying Markov chain, which, in turn, depends on the exact system model, i.e., a knowledge of the statistics of the channel gain and arrival processes. Since this knowledge is difficult to possess in practice, the central theme of the thesis is to develop efficient scheduling algorithms that do not require this knowledge. We demonstrate that stochastic approximation and reinforcement learning frameworks can be successfully employed for this purpose. We consider four scenarios: point-to-point, uplink, downlink and distributed transmission.

For the point-to-point scenario, we consider minimizing the long term average transmission power expenditure subject to average packet delay constraint. While this problem has been formulated in the literature within the Constrained Markov Decision Process (CMDP) framework, the issue of determining the optimal packet scheduling policy has largely remained untouched. We suggest an online algorithm for this problem based on the novel concept of a post decision state. This algorithm is a reformulation of the well known Relative Value Iteration Algorithm (RVIA). The constraint is naturally handled through the Lagrangian approach. The optimal value function and optimal Lagrange Multiplier

(LM) are determined using simultaneous iterations, albeit at different timescales. We prove that the algorithm asymptotically converges to the optimal scheduling policy. The simulation results demonstrate that the algorithm converges to a ‘near optimal’ regime in reasonable number of iterations and hence it is quite useful in practice.

For the multiuser uplink scenario, we consider minimizing the average power expenditure of each user subject to individual delay constraint. The primary issue in providing an efficient solution is that of the large state space. To address this issue, in our approach, each user’s queue evolution behaves as if it were controlled by a single user policy. Depending on each user’s channel state and queue size, the algorithm allocates a certain rate to each user in a slot using a single user algorithm. The algorithm then schedules the user with the highest rate in a slot. We argue that the delay constraints are satisfied and that the algorithm has a stabilizing structure, which is confirmed by the simulations within the IEEE 802.16 framework.

Next, we consider the problem of scheduling users on the downlink of a Time Division Multiplexing (TDM) system. Our objective is to maximize the sum throughput with constraints on the user delays. Due to the large state space and unknown system model, the traditional approaches based on LP within the CMDP framework are rendered infeasible. We, therefore, propose a sub-optimal scheduling algorithm which is based on computing appropriate indices and scheduling the user with the highest index. We prove that the proposed algorithm satisfies the delay constraints. The simulations within an IEEE 802.16 system indicate that the algorithm is throughput efficient.

Finally, for the single hop distributed transmission scenario, we consider the problem of designing a random access mechanism with an objective of minimizing transmission power while satisfying the user delay constraints. The users located in a geographical area are divided into source-destination pairs. The users are located in close vicinity such that only one source can transmit at any instant of time. A source regulates its transmission probability (or equivalently, channel access rate) such that it is just enough for satisfying the delay constraints. Accessing the channel at a higher rate may result in higher number of collisions with other users thus leading to wastage of bandwidth as well as energy, while accessing the channel at a lower rate may not be enough for satisfying the delay constraint. We formulate the problem as a constrained multistage optimization

problem. We propose two algorithms, both based on stochastic approximation. The first algorithm is based on the stochastic gradient approach. It is a three timescale algorithm where the transmission probability is tuned based on the gradient of the Lagrangian. The second algorithm is a single timescale stochastic approximation algorithm. We prove that it satisfies the delay constraints and that it converges to an equilibrium. Both algorithms are quite simple to implement in practice and have no communication overhead.



# Contents

<b>Abstract</b>	<b>v</b>
<b>List of Acronyms</b>	<b>xiii</b>
<b>List of Symbols</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Figures</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 QoS in Wireless Networks . . . . .	2
1.2 QoS Mechanisms . . . . .	5
1.3 Cross Layer Design . . . . .	5
1.3.1 Advantages of Cross Layer Design . . . . .	6
1.3.2 Cross Layer Scheduling - Implementation Issues . . . . .	7
1.4 Motivation for the Thesis . . . . .	8
1.5 Contributions and Organization of the Thesis . . . . .	11
<b>2 Cross Layer Scheduling: Approaches, Performance Limits and Open Issues</b>	<b>15</b>
2.1 Wireless Channel Characteristics . . . . .	16
2.1.1 Multipath Fading . . . . .	16
2.1.2 Wireless Channel Model . . . . .	18
2.2 Capacity of Fading Channel . . . . .	20
2.2.1 Point-to-Point Capacity with Full Transmitter CSI . . . . .	21

2.2.2	Multiuser Capacity with Full Transmitter CSI on the Uplink . . . . .	24
2.2.3	Multiuser Capacity with Full Transmitter CSI on the Downlink . . . . .	28
2.2.4	Towards a Framework for Cross Layer Scheduling . . . . .	28
2.3	Multiuser Diversity with Centralized Scheduling . . . . .	30
2.3.1	System Model . . . . .	30
2.3.2	Throughput-Fairness Tradeoff . . . . .	33
2.3.3	Throughput-Delay Tradeoff . . . . .	36
2.3.4	Power-Delay Tradeoff . . . . .	40
2.4	Multiuser Diversity with Distributed Scheduling . . . . .	44
2.5	Discussion and Open Problems . . . . .	46
<b>3</b>	<b>Energy Efficient Scheduling for a Point-to-Point Link</b>	<b>49</b>
3.1	System Model . . . . .	50
3.2	Formulation as a CMDP . . . . .	52
3.3	Lagrangian Approach . . . . .	53
3.4	Online Algorithm . . . . .	55
3.4.1	Post-Decision State Framework . . . . .	55
3.4.2	Reformulation of RVIA . . . . .	56
3.4.3	LM Update . . . . .	57
3.4.4	Complete Primal Dual Algorithm . . . . .	57
3.4.5	Implementation Details . . . . .	58
3.5	Convergence Analysis . . . . .	59
3.6	Simulation Results . . . . .	62
3.7	Conclusions . . . . .	71
<b>4</b>	<b>Energy Efficient Scheduling for Multiuser Uplink</b>	<b>73</b>
4.1	System Model . . . . .	74
4.2	Problem Formulation . . . . .	76
4.2.1	Formulation as a Constrained Optimization Problem . . . . .	76
4.2.2	Notion of an Optimal Solution . . . . .	77
4.2.3	Difficulties in Determining an Optimal Solution . . . . .	77
4.3	Transmission in the Presence of Transmitter Errors . . . . .	79

---

4.3.1	Primal Dual Approach . . . . .	80
4.3.2	Online Rate Allocation Algorithm . . . . .	80
4.3.3	Proof of Convergence . . . . .	81
4.4	An Online Primal Dual Algorithm for the Multiuser Problem . . . . .	82
4.4.1	Rate Allocation Algorithm for a User . . . . .	82
4.4.2	User Selection Algorithm . . . . .	82
4.4.3	Implementation Details . . . . .	83
4.4.4	Discussion . . . . .	85
4.5	Analysis of AA . . . . .	86
4.6	Experimental Evaluation . . . . .	89
4.6.1	The IEEE 802.16 System . . . . .	89
4.6.2	Simulation Results . . . . .	90
4.7	Conclusions . . . . .	100
<b>5</b>	<b>Throughput Efficient Scheduling for Multiuser Downlink</b>	<b>101</b>
5.1	System Model . . . . .	102
5.2	Problem Formulation . . . . .	103
5.2.1	Formulation as a Constrained Optimization Problem . . . . .	103
5.2.2	Formulation within CMDP Framework . . . . .	104
5.2.3	Issues in Determining the Optimal Policy . . . . .	105
5.3	Indexing Scheduler . . . . .	106
5.3.1	Determining Weights . . . . .	107
5.3.2	Implementation Details . . . . .	108
5.4	Experimental Evaluation . . . . .	109
5.4.1	The IEEE 802.16 System . . . . .	110
5.4.2	Simulation Results . . . . .	110
5.5	Conclusions . . . . .	120
<b>6</b>	<b>Energy Efficient Scheduling for Multiuser Distributed Channel Access</b>	<b>123</b>
6.1	Introduction . . . . .	123
6.2	System Model . . . . .	125
6.3	Problem Formulation . . . . .	127

6.4	An Algorithm based on the Stochastic Gradient Approach . . . . .	127
6.4.1	Lagrangian Approach . . . . .	128
6.4.2	Stochastic Gradient Approach . . . . .	129
6.4.3	Implementation Details . . . . .	130
6.4.4	Limitations of SGA . . . . .	131
6.5	A Single Timescale Stochastic Approximation Algorithm (STSAA) . . . . .	132
6.5.1	Implementation Details . . . . .	133
6.5.2	Convergence Analysis . . . . .	133
6.6	Simulation Results . . . . .	136
6.7	Conclusions . . . . .	141
<b>7</b>	<b>Conclusions and Future Work</b>	<b>143</b>
7.1	Conclusions . . . . .	143
7.2	Future Work . . . . .	148
<b>A</b>	<b>Markov Decision Process</b>	<b>151</b>
A.1	Markov Decision Process . . . . .	151
A.2	Constrained Markov Decision Process (CMDP) . . . . .	154
<b>B</b>	<b>Reinforcement Learning</b>	<b>157</b>
B.1	Reinforcement Learning . . . . .	157
B.2	Q-learning . . . . .	158
<b>C</b>	<b>Stochastic Approximation</b>	<b>159</b>
C.1	Stochastic Approximation and Stochastic Iterative Algorithm (SIA) . . . . .	159
C.2	Convergence Analysis . . . . .	160
C.3	Stochastic Approximation on Two Timescales . . . . .	161
<b>D</b>	<b>Properties of Auction Algorithm (4.19)</b>	<b>163</b>

# List of Acronyms

1G	First Generation
2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
AA	Auction Algorithm
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BE	Best Effort
BER	Bit Error Ratio
BPSK	Binary Phase Shift Keying
BS	Base Station
CDMA	Code Division Multiple Access
CMDP	Constrained Markov Decision Process
CSI	Channel State Information
DL	Downlink
DL-MAP	Downlink-Map
DP	Dynamic Programming
EXP	Exponential
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
HDR	High Data Rate
i.i.d.	independent and identically distributed
IS	Indexing Scheduler

LCQ	Longest Connected Queue
LM	Lagrange Multiplier
LP	Linear Programming
MAC	Medium Access Control
MDP	Markov Decision Process
N.A.	Not Applicable
nrtPS	non real time Polling Service
o.d.e.	ordinary differential equation
PCMA	Power Controlled Multiple Access
PMP	Point to Multipoint
PSD	Power Spectral Density
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase shift Keying
REP-REQ	Report-Request
REP-RSP	Report-Response
r.h.s.	right hand side
RL	Reinforcement Learning
RNG-REQ	Ranging-Request
rtPS	real time Polling Service
RVIA	Relative Value Iteration Algorithm
SA	Stochastic Approximation
SGA	Stochastic Gradient Algorithm
SIA	Stochastic Iterative Algorithm
SMS	Short Message Service
SNR	Signal to Noise Ratio
SS	Subscriber Station
STSAA	Single Timescale Stochastic Approximation Algorithm
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDM	Time Division Multiplexing

TDMA	Time Division Multiple Access
TOCA	Tradeoff Optimal Control Algorithm
UCD	Uplink Channel Descriptor
UGS	Unsolicited Grant Service
UL	Uplink
UL-MAP	Uplink-Map
VoIP	Voice over Internet Protocol
w.r.t.	with respect to
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WWW	World Wide Web



# List of Symbols

$A_n^i$	Number of packets arriving into user $i$ queue in slot $n$
$\bar{a}^i$	Average arrival rate of a user $i$
$B$	Buffer size expressed in number of packets
$\bar{\delta}^i$	Queue length constraint of user $i$
$\bar{\delta}$	$N$ dimensional queue length constraint vector
$H_n^i$	Channel gain of user $i$ in slot $n$
$\mathbf{H}$	$N$ dimensional vector of channel gains
$i$	User index
$I_n^i$	Indicator variable, set to 1 if user $i$ is scheduled in slot $n$ , else set to 0
$\iota_n^i$	Index of user $i$ in slot $n$
$\lambda_n^i$	Lagrange Multiplier for user $i$ in slot $n$
$\boldsymbol{\lambda}_n$	$N$ dimensional Lagrange Multiplier vector in slot $n$
$\ell$	Packet length
$N$	Number of users in the system
$n$	Slot index
$\bar{P}^i$	Average power constraint for user $i$
$\bar{\mathbf{P}}$	$N$ dimensional average power constraint vector
$\hat{P}^i$	Peak power constraint for user $i$
$\hat{\mathbf{P}}$	$N$ dimensional peak power constraint vector
$\mathbf{Q}_n$	$N$ dimensional vector of queue lengths in slot $n$
$\bar{Q}^i$	Average queue length achieved by a user $i$
$Q_n^i$	Number of packets in user $i$ queue in slot $n$
$S_n^i$	State of user $i$ in slot $n$
$\mathbf{S}_n$	$N$ dimensional state vector in slot $n$

$\bar{T}$	Average sum throughput
$\theta_n^i$	Transmission probability for user $i$ in slot $n$
$U_n^i$	Number of packets transmitted by user $i$ in slot $n$
$\mathbf{U}_n$	$N$ dimensional transmission vector in slot $n$
$W$	Bandwidth
$\mathbf{X}$	$N$ dimensional vector of channel states
$X_n^i$	Channel state of a user $i$ in slot $n$

# List of Tables

1.1	QoS attributes for some applications (N.A. - Not Applicable, VoIP - Voice over Internet Protocol) . . . . .	2
2.1	Summary of relationship between types of fading and the channel and signal parameters . . . . .	18
3.1	Summary of parameters common for all scenarios . . . . .	64
3.2	Summary of parameters for Scenario 3.1, (Figure 3.2) . . . . .	65
3.3	Summary of parameters for Scenario 3.1, (Figure 3.3) . . . . .	66
3.4	Summary of parameters for Scenario 3.1, (Figure 3.4) . . . . .	67
3.5	Summary of parameters for Scenario 3.2, (Figures 3.5 and 3.6) . . . . .	68
3.6	Summary of parameters for Scenario 3.3, (Figures 3.7 and 3.8) . . . . .	70
3.7	Summary of parameters for Scenario 3.4, (Figures 3.9 and 3.10) . . . . .	71
4.1	Summary of parameters common for all scenarios . . . . .	91
4.2	Summary of parameters for Scenario 4.1 . . . . .	94
4.3	Summary of parameters for Scenario 4.2 . . . . .	94
4.4	Summary of parameters for Scenario 4.3 . . . . .	99
5.1	Summary of parameters common for all scenarios . . . . .	111
5.2	Summary of parameters for Scenario 5.1 . . . . .	112
5.3	Summary of parameters for Scenario 5.2 . . . . .	115
5.4	Summary of parameters for Scenario 5.3 . . . . .	117
5.5	Summary of parameters for Scenario 5.4 . . . . .	120
6.1	Summary of parameters common for all scenarios . . . . .	137

6.2	Summary of parameters for Scenario 6.1 . . . . .	137
6.3	Summary of parameters for Scenario 6.2 . . . . .	139
6.4	Summary of parameters for Scenario 6.3 . . . . .	141

# List of Figures

1.1	A generic wireless network . . . . .	3
1.2	A cellular network . . . . .	3
1.3	Channel quality variation with time . . . . .	9
2.1	Point-to-point transmission model . . . . .	22
2.2	Waterfilling power allocation . . . . .	23
2.3	Uplink transmission model, infinite backlog of bits at transmitters . . . . .	24
2.4	Downlink transmission model, infinite backlog of bits at base station for each user . . . . .	27
2.5	Uplink transmission model, finite buffer at each user . . . . .	30
2.6	Downlink transmission model, finite buffer for each user at base station . . . . .	31
2.7	Point-to-point transmission model with finite buffer . . . . .	40
3.1	Point-to-point transmission model with finite buffer . . . . .	50
3.2	Convergence of Lagrange multiplier for various average delay constraints . . . . .	64
3.3	Convergence of Lagrange multiplier for various average arrival rates . . . . .	65
3.4	Convergence of Lagrange multiplier for various average channel states . . . . .	66
3.5	Convergence of average delay for various average delay constraints . . . . .	67
3.6	Convergence of average power for various average delay constraints . . . . .	68
3.7	Achieved system delay for various average delay constraints . . . . .	69
3.8	Power-delay curve for various average channel states . . . . .	69
3.9	Achieved system delay for various average arrival rates . . . . .	70
3.10	Power-arrival rate curve . . . . .	71
4.1	System Model . . . . .	75

4.2	Hypothetical single user scenario . . . . .	78
4.3	Scheduling phases . . . . .	83
4.4	Achieved delay of a user with specified average delay constraints - symmetric case . . . . .	92
4.5	Power expended with specified average delay constraints - symmetric case . . . . .	92
4.6	Achieved delay of a user with specified average delay constraints - asymmetric case . . . . .	93
4.7	Power expended with specified average delay constraints - asymmetric case . . . . .	93
4.8	Achieved delay of a user with varying average channel states - symmetric case . . . . .	95
4.9	Power expended with varying average channel states - symmetric case . . . . .	95
4.10	Achieved delay of a user with varying average channel states - asymmetric case . . . . .	96
4.11	Power expended with varying average channel states - asymmetric case . . . . .	96
4.12	Achieved delay of a user with varying average arrival rates - symmetric case . . . . .	97
4.13	Power expended with variation in average arrival rate - symmetric case . . . . .	98
4.14	Achieved delay of a user with varying average arrival rates - asymmetric case . . . . .	98
4.15	Power expended with variation in average arrival rate - asymmetric case . . . . .	99
5.1	Downlink transmission schematic, finite buffer for each user at base station . . . . .	102
5.2	Delay experienced by a user selected at random for various average delay constraints - symmetric case . . . . .	113
5.3	Delay experienced by two users selected at random from Group 1 and Group 2 for various average delay constraints - asymmetric case . . . . .	113
5.4	Sum throughput for various average delay constraints - symmetric case . . . . .	114
5.5	Sum throughput for various average delay constraints - asymmetric case . . . . .	114
5.6	Delays experienced under IS with those under M-LWDF scheduler as constraints . . . . .	116
5.7	Comparison of the sum throughput under M-LWDF and IS . . . . .	116
5.8	Delay experienced by a user selected at random for various average channel states - symmetric case . . . . .	118

---

5.9	Delay experienced by two users selected at random from Group 1 and Group 2 for various average channel states - asymmetric case . . . . .	118
5.10	Sum throughput for various average channel states - symmetric case . . . . .	119
5.11	Sum throughput for various average channel states - asymmetric case . . . . .	119
5.12	Delay experienced by a user selected at random for various average arrival rates - symmetric case . . . . .	121
5.13	Delay experienced by two users selected at random from Group 1 and Group 2 for various average average arrival rates - asymmetric case . . . . .	121
5.14	Sum throughput for various average arrival rates - symmetric case . . . . .	122
5.15	Sum throughput for various average arrival rates - asymmetric case . . . . .	122
6.1	Distributed transmission scenario . . . . .	125
6.2	Achieved delay for various average delay constraints . . . . .	138
6.3	Transmission probability (TP) for various average delay constraints . . . . .	138
6.4	Achieved delay for various average arrival rates . . . . .	140
6.5	Transmission probability (TP) for various average arrival rates . . . . .	140
6.6	Achieved delay for various average channel states . . . . .	141
6.7	Transmission probability (TP) for various average channel states . . . . .	142



# Chapter 1

## Introduction

Recent years have witnessed large scale proliferation of wireless communication technology. This has dramatically altered the way people communicate. The number of mobile subscribers stood at 3 billion worldwide by the end of August 2007 [1]. In India, by the end of March 2008, the number of cellular subscribers stood at 192 million [2]. Moreover, India adds about 7-8 million subscribers every month.

The first and second generation (1G and 2G) cellular systems concentrated primarily on providing better voice quality based on circuit switching. However, recently, applications such as email, Short Message Service (SMS), World Wide Web (WWW), multimedia applications, online gaming and peer-to-peer applications are gaining popularity with the users. Hence, in the third generation (3G) cellular standards such as those developed by Third Generation Partnership Project (3GPP) [3] and Third Generation Partnership Project 2 (3GPP2) [4], as well as other broadband wireless technologies such as Worldwide Interoperability for Microwave Access (WiMAX) based on the IEEE 802.16 standard [5] and Wireless Fidelity (WiFi) based on the IEEE 802.11 standard [6], the emphasis has been on providing satisfactory services to these applications based on packet switching.

Different applications mandate different kinds of packet delivery guarantees in order to perform satisfactorily. Real time applications such as streaming audio or video, typically, have a strict rate requirement [7]. Moreover, they also have an upper bound on the packet loss rate for satisfactory user experience. On the other hand, data applications, such as file downloads, are non-real time and do not have strict rate requirements. However, they expect zero packet loss. These and several other requirements such as worst

Application	Type	Rate	Delay	Packet Loss	Delay Jitter
VoIP	Real time	4 - 64 kbps	<100 msec	< 1%	< 20 msec
Interactive gaming	Real time	50 - 85 kbps	50-150 msec	Zero	N.A.
Web browsing, email, file downloads	Non-real time	0.01-100 Mbps	Flexible	Zero	N.A.
Streaming video	Real time	5 - 384 kbps	<250 msec	< 2%	<2 sec

Table 1.1: QoS attributes for some applications (N.A. - Not Applicable, VoIP - Voice over Internet Protocol)

case or average rate guarantees, worst case or average packet delay, probability of packet drop, maximum jitter between packets, among others are collectively termed as Quality of Service (QoS) attributes. Table 1.1 provides details regarding the QoS attributes for representative applications.

## 1.1 QoS in Wireless Networks

As depicted in Figure 1.1, a *wireless network*, in its most general form, comprises of source nodes<sup>1</sup> communicating with destination nodes possibly through multiple intermediate wireless nodes. In a cellular system, wireless network takes the form depicted in Figure 1.2. In this case, a wireless node located within a certain region called a *cell* communicates with an entity called the base station corresponding to that cell, over the wireless channel. Wireless networks, due to their unique characteristics, pose several challenges in providing QoS to the applications. We discuss some of these.

- *Wireless Channel*: One of the most important challenges in providing QoS over a wireless network comes from the channel itself. Wireless channel is characterized by decay of signal strength due to distance (path loss), obstructions due to objects such as buildings and hills (shadowing), and constructive and destructive interference caused by copies of the same signal received over multiple paths (multipath fading) [8]. These phenomena distort the signal in an unpredictable manner and can cause

---

<sup>1</sup>We use the terms ‘node’ and ‘user’ interchangeably in this thesis.

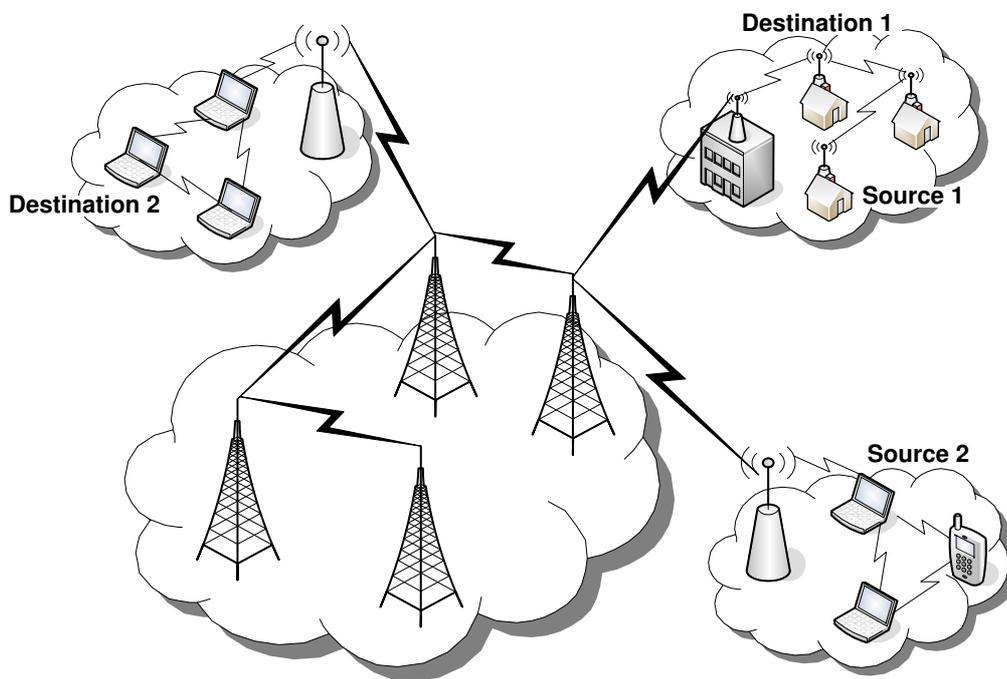


Figure 1.1: A generic wireless network

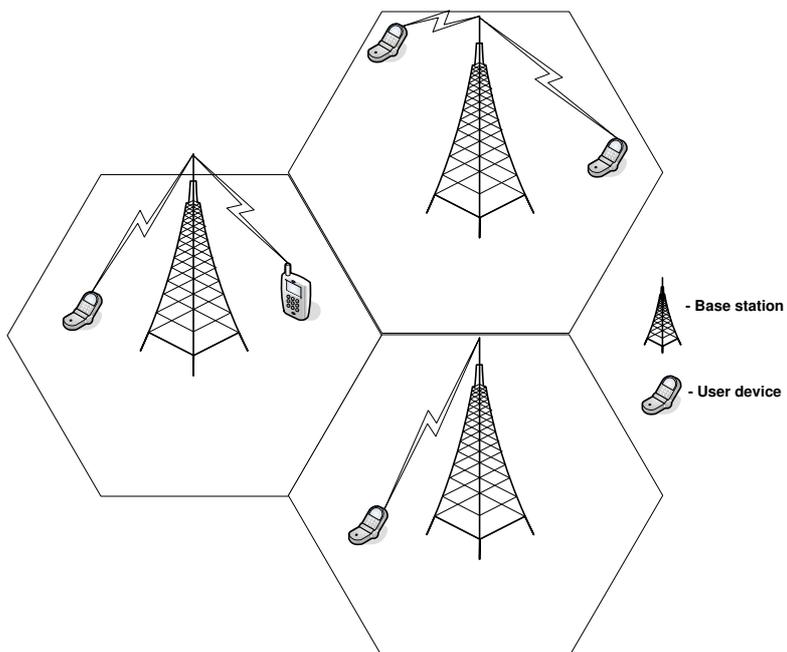


Figure 1.2: A cellular network

packet errors at the receiver. Thus, ensuring reliable packet delivery is a challenge. Moreover, packets in error need to be retransmitted, thus mandating the source (and possibly intermediate nodes) in the network to store the packets until they are successfully delivered at the destination. This makes the buffer space allocation a difficult task since the buffer space in a wireless node is typically limited.

- *Spectrum Scarcity*: Wireless spectrum regulations limit the amount of spectrum availability. This motivates the need for spectrally efficient coding and modulation schemes. Moreover, arbitration mechanisms should share the spectrum among various nodes efficiently.
- *Energy Efficiency*: There is an intrinsic relationship between QoS and energy expenditure. Intuitively, transmission at higher power results in the receiver perceiving a higher Signal to Noise Ratio (SNR) thus reducing the probability of packet error [9]. Consequently, it facilitates high rate ‘reliable’ transmission. Wireless devices, being battery powered, are energy constrained and have a limit on the power they can expend on transmission. This poses a significant challenge for providing QoS.
- *Mobility*: Reliable packet delivery in a wireless network with mobile nodes is a challenge because nodes may move out of the transmission range of each other. In a cellular network, nodes may move from the transmission range of one base station to that of another base station. Thus, the network has to divert the packets corresponding to that node to the appropriate base station, thereby introducing additional delays. These issues introduced by mobility make the problem of providing QoS a challenging one.

The network resources such as the wireless spectrum and energy are essential for communication. In order to provide QoS, these resources must be allocated in an efficient manner. Thus, the problem of providing QoS is a resource allocation problem.

## 1.2 QoS Mechanisms

Resource allocation involves a notion of *capacity*<sup>2</sup>, i.e., a quantification of the amount of resources available with the network. Each time the network promises certain QoS guarantees to an application, some amount of capacity is used. Hence, before admitting a user which requires a certain QoS, the network has to ensure that it has sufficient capacity available to satisfy the requirement. If there is sufficient capacity available, the user is admitted into the network, else it is denied access. This task is performed by the *admission control* component [7]. Once a user is admitted into the network, it has to ensure that the promised QoS is indeed delivered to it. This task is accomplished by means of *packet scheduling* algorithms. These algorithms decide the order in which packets corresponding to the user are transmitted such that their QoS requirements are met. Delivering QoS also needs to take care of mobility. This is addressed using the *mobility management* component.

In this thesis, we focus on the scheduling aspect. Specifically, we design scheduling strategies for providing average packet delay guarantees. Before studying the role of scheduling algorithms in providing QoS, we study an important recent paradigm in the wireless networks termed as the cross layer paradigm that influences the design of such algorithms.

## 1.3 Cross Layer Design

Traditional communication models such as the Open Systems Interconnection (OSI) model adopt a *layered* architecture where the entire networking task is divided into hierarchy of services provided by individual layers [10]. Services at a layer are provided by means of a communication protocol defined at that layer. Non-adjacent layers are not allowed to communicate, while adjacent layers communicate by means of procedure calls and responses. The *cross layer* design paradigm [11, 12, 13] advocates violating the layered architecture paradigm in one or more of the following ways:

- Creating interfaces and sharing or tuning parameters between non-adjacent layers.

---

<sup>2</sup>Different notions of capacity have been defined in the literature including information theoretic notions. We will define the precise notion of capacity, relevant to this thesis, in Chapter 2.

- Joint design of protocols at two or more layers.

### 1.3.1 Advantages of Cross Layer Design

In recent years, the cross layer design paradigm has been strongly advocated for the wireless networks as it has resulted in substantial improvement in system performance. For example, variants of Transmission Control Protocol (TCP) that obtain information from physical layer regarding channel condition can distinguish between packet losses due to congestion and poor channel condition. These physical layer aware TCP variants have substantially improved performance [14, 15].

The cross layer design paradigm has also resulted in improving the performance of scheduling algorithms at the Medium Access Control (MAC) layer by making use of channel related information from the physical layer. Such scheduling algorithms are termed as *channel aware* or *cross layer* scheduling algorithms. The following example from [16] demonstrates the benefits obtained by cross layer scheduling. Consider a cellular system where the base station schedules two users on the downlink. Suppose that the users' channel quality (indicated by the received signal power) is characterized by two states - 'good' (indicates higher received power) and 'bad' (indicates lower received power) respectively. Assume that the base station can transmit at a reliable rate of 100 kbps to a user 1 when its channel quality is good and at 50 kbps when it is bad. For user 2, the base station can transmit at a reliable rate of 200 kbps and 100 kbps when its channel quality is good and bad respectively. Let us assume that both users perceive good and bad channel quality with equal probability of 0.5. Thus, on an average, user 2 perceives a better channel quality than user 1. If the base station uses a simple round robin scheduler then the average throughput obtained over the long run by user 1 can be calculated as:

$$0.5 \times (0.5 \times 100 + 0.5 \times 50) = 75 \text{ kbps.} \quad (1.1)$$

Similarly, the average throughput obtained over the long run by user 2 can be calculated as:

$$0.5 \times (0.5 \times 200 + 0.5 \times 100) = 150 \text{ kbps.} \quad (1.2)$$

On the other hand, consider a scheduler that uses a channel aware scheduling scheme where the base station schedules the user that perceives a 'relatively' better channel

quality (user 1 transmits at 100 kbps and user 2 transmits at 200 kbps). If there is a tie, i.e., both users perceive a relatively better or worse channel quality, a user is selected randomly with equal probability. In this case, the average throughput obtained over the long run by user 1 can be calculated as:

$$0.25 \times 0.5 \times 100 + 0.25 \times 100 + 0.25 \times 0.5 \times 50 = 87.5 \text{ kbps}, \quad (1.3)$$

while that obtained over the long run by user 2 can be calculated as:

$$0.25 \times 0.5 \times 200 + 0.25 \times 200 + 0.25 \times 0.5 \times 100 = 175 \text{ kbps}. \quad (1.4)$$

It is clear that channel aware or cross layer scheduling improves performance as compared to channel unaware schemes.

### 1.3.2 Cross Layer Scheduling - Implementation Issues

Note that cross layer scheduling assumes a knowledge of the channel quality as perceived by the receiver. In order for cross layer scheduling to work in a practical system, several operations need to be performed:

- The receiver has to estimate the channel quality and inform the estimate to the transmitter.
- An entity such as the base station scheduling the packets for the users on the *uplink* has to determine a particular user who should transmit at any instant of time. Moreover, it also has to determine the rate and power at which the user should transmit. On the *downlink*, the transmission is typically at a fixed power. However, the base station can vary its transmission rate based on the channel quality perceived by the user to which it transmits. This variable rate transmission is possible through adaptive modulation and coding schemes [17].

In a practical system such as WiMAX [5], the ranging request (RNG-REQ) messages can be used to convey channel quality information to the base station for downlink scheduling. Moreover, the base station can also transmit a Channel Measurement Report Request (REP-REQ) to obtain channel related information. The nodes then respond to this message using the Channel Measurement Report Response (REP-RSP) messages. The

base station informs the scheduling decision to users using the Downlink Map (DL-MAP) transmitted at the beginning of each frame. For the uplink scheduling, the base station has to convey channel quality information to users. This information can be conveyed using Uplink Channel Descriptor (UCD) messages which is a part of Uplink Map (UL-MAP) transmitted at the beginning of each frame following the DL-MAP. Moreover, the UL-MAP is used to inform the scheduling decision to users on the uplink.

## 1.4 Motivation for the Thesis

Having introduced cross layer paradigm, let us now study three important aspects that illustrate how information from the physical layer can be exploited for improving the performance of scheduling schemes at the MAC layer. First is the convex rate-power relationship. In wireless communication, the power required for transmitting at a rate is a convex and increasing function of the rate [18]. Let the power required for transmission at a rate  $r$  be denoted by  $F_p(r)$ , where  $F_p(r)$  is a convex and increasing function of  $r$ . Consider a slotted transmission system serving a single user that generates a total of  $a$  packets of unit size in alternate slots. Let the slot duration be normalized to unity. Consider the performance of the following two transmission schemes:

- *Scheme A*: The scheduler sends packets for transmission in the same slot as they arrive. The packets, then, do not experience any queuing delays and the average power consumed by the scheme is  $F_p(a)$ .
- *Scheme B*: The scheduler sends half the packets, i.e.,  $\frac{a}{2}$  packets for transmission in the same slot as they arrive and sends the remaining half for transmission in the next slot that has no arrivals. In this scheme, a packet experiences a delay of half a slot on an average. The average power requirement, however, is now  $F_p(\frac{a}{2})$  which can be substantially less than that required by Scheme A, due to the convexity of  $F_p(\cdot)$ .

It is clear that power can be saved by transmitting at lower rates, albeit by incurring higher delay.

The second aspect that influences the scheduler design is related to the channel quality perceived by the receiver. Reliable transmission at a certain rate under better

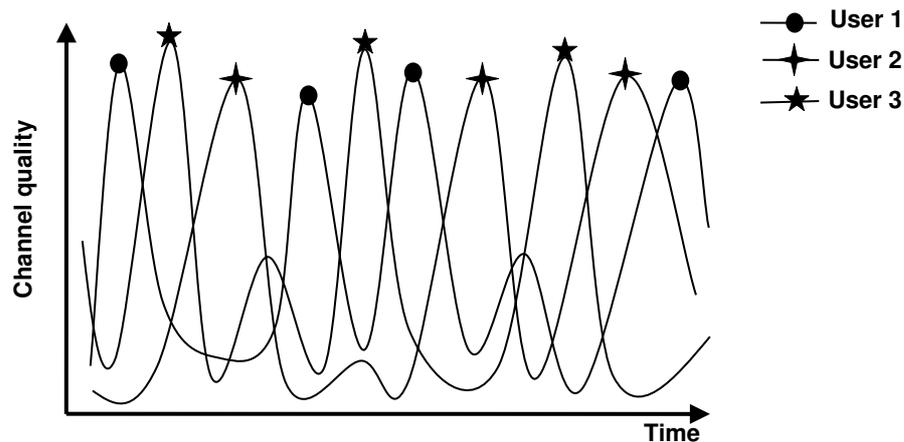


Figure 1.3: Channel quality variation with time

channel quality requires much less power than what is required under poor channel quality [19]. Thus, the transmitter can wait for the channel to become better in order to transmit at a certain rate and thereby save power. This, again points to the fact that power can be saved at an expense of higher delay.

The third aspect is related to the relative channel quality perceived by different users in a multiuser setting. Consider a cellular system where the base station schedules transmissions to users on the downlink. As depicted in Figure 1.3, the channel quality perceived by each user is time varying. The peaks indicate that the user perceives the ‘best’ channel quality, while valleys indicate that it perceives the ‘worst’ channel quality. In each slot, the channel quality perceived by different users is independent of each other and is diverse; a phenomenon referred to as *multiuser diversity* [20]. The base station can, thus, schedule the users based on the channel quality perceived by them. In each slot, if the base station schedules the user with the best channel quality, the sum throughput (i.e., the sum of the throughput obtained by all users) is maximized [21]. Thus, multiuser diversity can be exploited in order to improve the sum throughput. However, it can result in higher delay for a user who perennially perceives poor channel quality. On the other hand, scheduling a user not perceiving the best channel quality results in a reduction in sum throughput.

The first two aspects indicate a power-delay tradeoff. This tradeoff reaffirms the observation made previously that the objectives of energy efficiency and providing QoS

are two contrasting objectives. One needs to seek an appropriate balance between these objectives based on application requirements. The third aspect indicates a throughput-delay tradeoff. While the network operator would like to have a high sum throughput, the users would like to have either lower delays, or delays below some desired limit. We exploit these tradeoffs for designing scheduling schemes that optimize quantities such as power expenditure and sum throughput while providing QoS to the users.

Cross layer scheduling problems have been formulated as optimization problems where the objective is to optimize a certain utility function such as throughput or power subject to QoS constraints. These problems can be cast as control problems wherein the scheduler can be viewed as a controller that has the objective of determining the user to be scheduled in a slot, its transmission rate and transmission power. A well known approach for determining an optimal solution is to cast the problem as a Markov Decision Process (MDP) [22] and then determine the optimal packet scheduling policy. However, this approach has two major issues. Firstly, an MDP faces the problem of curse of dimensionality, i.e., numerical approaches for determining an optimal scheduling policy become computationally infeasible for moderate to large state space. Secondly, these approaches require a knowledge of the transition probability mechanism of the underlying Markov chain. Determining this transition probability mechanism requires a knowledge of the *system model*, i.e., statistical characteristics of the channel and the packet arrival processes. In practice, it is difficult to know the exact system model. To overcome this issue, one can assume a model, e.g, assume Rayleigh channel and Poisson arrival process for a user. However, performance of the schemes designed with these assumptions is limited by the modeling accuracy. This provides us the motivation to devise packet scheduling algorithms that do not require this knowledge and hence are not limited by the accuracy of the system model. Towards this goal, we design computationally efficient scheduling schemes for optimizing the utility functions such as average power or sum throughput while providing the required average delay guarantees. These schemes do not require any knowledge of the statistics of the channel or arrival processes of the users. As will be discussed in subsequent chapters, this is achieved by exploiting Reinforcement Learning (RL) [23, 24] and Stochastic Approximation (SA) [25, 26] frameworks in a novel manner.

## 1.5 Contributions and Organization of the Thesis

In this section, we outline some of the salient contributions of the thesis. The thesis is organized into seven chapters. Chapter 2 presents a review of relevant literature. Chapters 3-6 present our contributions. In each of these chapters, our objective is to optimize a certain utility function such as power or throughput subject to average delay constraints. The chapter wise contributions are outlined below.

- In Chapter 2, we provide a review of representative approaches in the literature for scheduling over a fading channel. Information theoretic approaches, that put a fundamental limit on the achievable performance under fading, are reviewed first. These approaches provide key insight for efficient communication over the fading channel. Moreover, they lay the foundation for development of scheduling algorithms at the MAC or network layer. A variety of scheduling algorithms catering to various optimization objectives such as power minimization, sum throughput maximization subject to QoS constraints such as average delay and fairness have been proposed in the literature. These algorithms are reviewed subsequently. We conclude the chapter with some open issues.
- We consider the point to point communication scenario over a fading channel in Chapter 3. The problem considered for this scenario is that of average power minimization subject to maintaining average packet delay below a prescribed bound. This problem has been formulated in the literature as a Constrained Markov Decision Problem (CMDP) [27, 28, 29, 30]. There are numerous papers that prove results regarding structural properties of the optimal policy. However, an important issue of *computing* the optimal packet scheduling policy has not been addressed satisfactorily in the sense that most schemes assume a knowledge of the system model. To address this issue, we develop a model unaware online scheduling algorithm that iteratively computes the optimal scheduling policy. It is an online version of the Relative Value Iteration Algorithm (RVIA) [31], a well known algorithm in the MDP literature. Our approach is based on a novel idea of reformulating the RVIA based on introducing a virtual state referred to as the post decision state. The resultant algorithm has a nice structure that lends itself naturally to an online implementa-

tion based on stochastic approximation. We prove that the algorithm determines the optimal policy asymptotically. While analytical results are asymptotic, the simulation results demonstrate that the algorithm is quite useful in practice in the sense that it converges to near optimal values in a reasonable number of iterations.

- An extension of the scheduling framework developed in Chapter 3 for multiuser uplink communication is presented in Chapter 4. The specific problem studied is that of minimizing the average power expenditure of each user subject to satisfying the average delay constraint of each user. This problem has not been previously considered in the literature primarily because of the large state space. To address this issue, we propose a novel extension of the single user algorithm. In the proposed approach, each user's queue evolution behaves as if it were controlled by a single user policy. Depending on a user's channel state and queue size, the algorithm allocates a certain rate to the user in a slot. This rate allocation is performed using a variation of the single user algorithm developed in Chapter 3. The algorithm then schedules the user with the highest rate in a slot. The proposed algorithm does not require the knowledge of the system model. We argue that this algorithm has a stabilizing behavior. We demonstrate the efficacy of our algorithm through simulations within an IEEE 802.16 system.
- In Chapter 5, we consider the problem of sum throughput maximization subject to satisfying individual user average delay constraints. While there is abundant literature on downlink scheduling with various other objectives and QoS constraints, this specific problem has not been considered previously. We formulate the problem within the CMDP framework. However, our basic argument of not imposing model related restrictions coupled with large state space, render the traditional approaches infeasible for determining the optimal policy. Hence, we suggest a sub-optimal, albeit, efficient approach based on generating appropriate indices for scheduling the users. The resultant scheme, referred to as the Indexing Scheduler (IS), schedules the user having the highest index in a slot. We prove that the scheme satisfies the QoS requirements. Our simulation studies involving comparisons with other schemes in the literature within IEEE 802.16 framework indicate that our scheduler is highly throughput efficient.

- Chapters 4 and 5 focus on centralized scheduling aspects, while Chapter 6 investigates distributed scheduling. We consider pairs of nodes (sources and destinations) communicating with each other in a vicinity. These are so located that all nodes are within transmission range of all other nodes and hence only one node can transmit at any instant of time. We consider the problem of determining the minimum channel access rate or transmission probability for a source such that it is just sufficient to satisfy the average packet delay constraint. We suggest two iterative schemes referred to as Stochastic Gradient Algorithm (SGA) and Single Timescale Stochastic Approximation Algorithm (STSAA) for determining the steady state transmission probability. We prove that STSAA satisfies the delay constraints and converges to an equilibrium. We present simulation results to demonstrate that the schemes indeed satisfy the delay constraints.

We conclude with directions for future work in Chapter 7. In order to make the thesis self contained, we include three appendices that present relevant results and approaches from the literature related to Markov Decision Process, Reinforcement Learning and Stochastic Approximation.



## Chapter 2

# Cross Layer Scheduling: Approaches, Performance Limits and Open Issues

Traditionally, fading has been viewed as a hindrance to communication over the wireless channel. Recent results from information theory [21, 32, 33] have provided key insights concerning efficient transmission of information over fading channels. As pointed out in the previous chapter, these results suggest that fading can also be considered as an opportunity for improving performance instead of it being always viewed as an adversary. In this chapter, we review some of these information theoretic results for the fast fading channel and emphasize their impact on the design of scheduling algorithms at the MAC layer. As outlined in the previous chapter, the exploitation of physical layer characteristics to obtain performance gains at higher layers of the protocol stack has now come to be known as the cross layer paradigm. Various cross layer scheduling schemes catering to different QoS objectives such as maximizing throughput, minimizing delay, or minimizing energy have been proposed in the literature. Our focus in this chapter is not to provide an exhaustive review of all these algorithms. Rather, we examine some of the representative work in this area with a view to elucidate the nature of the problems being considered in the literature. Towards the end of this chapter, we attempt to capture some of the key open issues in cross layer scheduling that form the basis for our investigations in the subsequent chapters.

## 2.1 Wireless Channel Characteristics

We begin this chapter by first providing a brief account of the characteristics of wireless channel that makes it different from wired channel, thereby posing several design challenges. Though the material in this section about fading characterization is somewhat standard, the treatment here is intended to introduce some terminology and assumptions that have been used in later part of the thesis. A more comprehensive account of fading in wireless channel in a general setting is available elsewhere [17, 19, 20].

### 2.1.1 Multipath Fading

Wireless users perceive time varying channel quality. The variations of the received signal strength at the receiver can be roughly attributed to two different phenomena:

- **Large Scale Fading:** There is an average signal strength attenuation or path loss depending on the distance between the transmitter and receiver. Moreover, shadowing due to large objects such as buildings and hills also causes signal strength attenuation. This phenomenon is termed as large scale fading and is typically frequency independent [8, 34].
- **Small Scale Fading:** The transmitted signal can reach the receiver over multiple paths (multipath propagation). The received signal is a vector addition of multipath components arriving over different paths. The relative motion between the transmitter and receiver and/or movement of the reflecting objects results in random path length changes; consequently the different multipath components have random amplitudes and phases. These fluctuations in the received signal's amplitude and phase can be large even for movement over very short distances. These fluctuations depend on the signal wavelength (and thereby frequency) and are referred to as small scale fading [8, 34].

In this thesis, we concentrate primarily on small scale fading and its impact on the design of scheduling strategies at the MAC layer. Henceforth, fading in this thesis refers to small scale fading.

Multiple copies of the transmitted signal reach the receiver at various instants of time depending on the length of the path over which the signal traverses. The *excess*

*delay* ( $\tau$ ) corresponding to a path is the difference between the signal's propagation delay along that path and the delay of the first signal arrival at the receiver. The *multipath intensity profile* captures the variation of the received average power  $S(\tau)$  as a function of the excess delay  $\tau$ . For a single transmitted *impulse*, let us define a threshold relative to the strongest multipath component. A multipath component is called the *last* component if its power falls below this threshold. The time  $T_m$  between the first and the last received components represents the *maximum excess delay*. Let  $T_s$  represent the symbol time. If  $T_m > T_s$  then the received multipath components of a transmitted symbol extend beyond the symbol's time duration. This leads to *Inter Symbol Interference (ISI)* and the channel is said to exhibit *frequency selective fading*. On the other hand, if  $T_m < T_s$ , all the received multipath components corresponding to a symbol arrive within the symbol time duration. In this case, the channel is said to exhibit *frequency non-selective* or *flat fading*.

The *coherence bandwidth*  $f_0$  of the channel is a statistical measure of the range of frequencies over which the channel distorts all spectral components of the signal in a similar fashion, i.e., with equal gain and linear phase. An approximate relationship between the coherence bandwidth and the maximum excess delay can be expressed as:

$$f_0 \approx 1/T_m. \quad (2.1)$$

The transmission bandwidth  $W$  of the signal can be approximately expressed as:

$$W \approx 1/T_s. \quad (2.2)$$

Hence, a channel is said to exhibit frequency selective fading if the coherence bandwidth is less than the signal transmission bandwidth, i.e.,  $f_0 < W$ , while it is frequency non-selective or flat if  $f_0 > W$ . Note that frequency selective fading causes ISI in the time domain.

The receiver perceives a time varying channel primarily because the motion between the transmitter and receiver results in changes in propagation paths. The expected time over which the channel's response to two *impulses* sent by the transmitter at times  $t_1$  and  $t_2$  is invariant, is referred to as the *channel coherence time*  $T_0$ . A channel is said to exhibit *fast fading* if the channel coherence time  $T_0$  is less than the symbol transmission time  $T_s$ , i.e.,  $T_0 < T_s$ . On the other hand, a channel is said to exhibit *slow fading* if the channel coherence time  $T_0$  is greater than the symbol transmission time  $T_s$ , i.e.,  $T_0 > T_s$ .

Type of Fading	Fast	Slow
Frequency selective	$T_0 < T_s, W < f_d$	$T_0 > T_s, W > f_d$
	$T_m > T_s, W > f_0$	$T_m > T_s, W > f_0$
Flat	$T_0 < T_s, W < f_d$	$T_0 > T_s, W > f_d$
	$T_m < T_s, W < f_0$	$T_m < T_s, W < f_0$

Table 2.1: Summary of relationship between types of fading and the channel and signal parameters

Fast/slow fading can be explained in the frequency domain as well. Due to multipath components reaching the receiver, each having a different amplitude, phase and angle of arrival, the receiver perceives a spectral broadening in the frequency domain. The width of this signal power spectrum is called the Doppler spread  $f_d$ . The Doppler spread and the coherence time are approximately related as [8]:

$$T_0 \approx \frac{1}{f_d}. \quad (2.3)$$

Hence, the Doppler spread can be considered as the *fading rate* of the channel. Approximating the symbol rate  $1/T_s$  by the signal bandwidth  $W$ , a channel is said to exhibit fast fading if the signal bandwidth is less than the fading rate, i.e.,  $W < f_d$  and is said to exhibit slow fading if  $W > f_d$ .

These relationships between the types of fading, and the channel and signal parameters have been summarized in Table 2.1.

### 2.1.2 Wireless Channel Model

Since the wireless channel has a time varying characteristic, it can be viewed as a filter with a time varying transfer function. In discrete time representation, the channel is modeled as a tapped delay line filter with finite, say,  $L$  taps. Under this model, each tap can be assumed to correspond to a delay window during which different multipath components arrive at the receiver. The number of taps,  $L$ , thus depends on the maximum excess delay  $T_m$ , and the tap gain depends on the amplitude and phase of the multipath components arriving during the corresponding time interval. Let  $H_{l,m}$  denote the  $l^{\text{th}}$  complex channel filter tap gain at time  $m$ . Let  $\chi_m$  denote the signal transmitted at time  $m$ . The received

signal  $Y_m$  can be expressed as:

$$Y_m = \sum_{l=0}^{L-1} H_{l,m} \chi_{m-l} + Z_m, \quad (2.4)$$

where  $Z_m$  is the complex Additive White Gaussian Noise (AWGN) with Power Spectral Density (PSD)  $N_0$ .

In this thesis, we limit ourselves to flat fading channels that can be modeled using a filter with a single tap. Thus, the received signal  $Y_m$  can be expressed as:

$$Y_m = H_m \chi_m + Z_m, \quad (2.5)$$

where we drop the suffix  $l$  from  $H_{l,m}$ .

We now discuss a model for the channel filter taps. Assume that there is a large number of *independent* reflected and scattered paths, with random amplitudes in the delay window corresponding to a single tap. Further, assume that the phase of each path is uniformly distributed between 0 to  $2\pi$ , and the phases of the individual paths are independent. The contribution of each path in the tap gain  $H_m$  can be modeled as a circularly symmetric complex random variable. Since  $H_m$  is a sum of large number of such independent circularly symmetric random variables, it can be modeled as a zero mean Gaussian random variable. The uniform phase implies that  $H_m$  is in fact circularly symmetric. Let  $\sigma^2$  denote the variance of  $H_m$ . Since  $H_m$  is modeled as a Gaussian random variable, its magnitude  $|H_m|$  is a Rayleigh random variable with probability density function expressed as:

$$f_H(h) = \frac{h}{\sigma^2} \exp\left(\frac{-h^2}{2\sigma^2}\right), \quad h \geq 0, \quad (2.6)$$

and the squared magnitude  $X_m \triangleq |H_m|^2$  is an exponentially distributed random variable with probability density function expressed as:

$$f_X(x) = \frac{1}{\sigma^2} \exp\left(\frac{-x}{\sigma^2}\right), \quad x \geq 0. \quad (2.7)$$

This model is called Rayleigh fading model.

### Block Fading Model

In this thesis, we assume that the channel tap coefficient remains constant for a *block* of symbols and changes only over block (for simplicity termed as ‘slot’ in this thesis)

boundaries. Such a model is called a *block fading* channel model [27]. We refer to  $X_n$ <sup>1</sup> as the *channel state* in slot  $n$ . Note that under the Rayleigh model,  $X_n$  is an exponentially distributed random variable. The channel may change across slots in an independent or correlated fashion, i.e.,  $X_n$  may be independent from slot to slot or as suggested in [35], follow a Markov model. However, in this thesis, we make the following two assumptions which are considered reasonable in practice.

1. We assume that the channel state  $X_n$ , instead of being a continuous random variable, is a discrete random variable and takes values from a finite set  $\mathbb{X}$ . This assumption is usually justified in practice and has been used in recent work such as [27, 36].
2. We assume that the channel state varies in an i.i.d. manner across slots<sup>2</sup>.

We now review some results about fundamental performance limits of communication over the fading channel under the modeling framework considered here. As will be outlined in subsequent sections later in this chapter, some of the key insights obtained from these limits set the foundation for the design of MAC layer scheduling algorithms.

## 2.2 Capacity of Fading Channel

For a wireless channel, capacity analysis can be performed both in the presence as well as absence of Channel State Information (CSI) at the transmitter. Throughout this thesis, we assume that the transmitter possesses full CSI since interesting possibilities emerge under this assumption. In a Time Division Duplex (TDD) system, due to channel reciprocity, it may be possible for the transmitter to obtain the CSI through channel estimation based on the signal received on the opposite link. In a Frequency Division Duplex (FDD) system, the receiver has to estimate the CSI and feed this information back to the transmitter. In practice, e.g., in IEEE 802.16 [5], the channel related information can be conveyed using ranging request (RNG-REQ) messages. In this thesis, we do not take into account the specific feedback mechanisms; rather, we assume that the transmitter possesses full CSI. Different notions of capacity of fading channels have been defined in the literature. See [19] for a thorough review. The classical notion of Shannon capacity defines the maximum

---

<sup>1</sup>We use  $n$  to denote slot index.

<sup>2</sup>Though some of our results may also hold for correlated variation - this aspect is discussed later.

information rate that can be achieved over the channel with zero probability of error [37]. This notion involves a coding theorem and its converse, i.e., that there exists a code that achieves the capacity (information can be reliably transmitted using this code at a rate less than or equal to the capacity) and that reliable communication is not possible if information is transmitted at a rate higher than the capacity.

For a fast fading channel, Shannon capacity refers to the achievable rates averaged over a large number of coherence time intervals. This capacity is termed as *ergodic* capacity, *throughput* capacity or *expected* capacity since it measures the rates achievable in the long run averaged over the channel variations. Here, the coherence time is assumed to be small, yet sufficiently large such that ‘reasonably’ long codes can be transmitted<sup>3</sup>.

For a slow fading channel, the notion of capacity corresponds to the rates achievable using codeword lengths that are independent of the channel variation. *Outage* is said to occur when the receiver is not able to decode the transmitted message. *Zero-outage* capacity or *delay-limited* capacity [38] refers to the case where the transmitter uses the CSI in order to invert the channel. It, therefore, maintains a constant received power or constant data rate at the receiver and hence zero outage regardless of the channel state. Achieving zero-outage capacity can require excessive amount of power. Moreover, when the channel state is poor, zero-outage capacity can be zero. In order to reduce the power consumption, one can consider an alternate scheme where transmission is suspended when the channel state is poor, while maintaining a higher data rate at better channel states. This leads to the notion of *capacity with outage*, defined as the product of the maximum achievable rate in all non-outage states and the probability of non-outage.

The notion of throughput capacity or expected capacity is relevant for most part of the thesis. Accordingly, we first derive an expression for the throughput capacity of a single user point-to-point link under an average power constraint and then extend the notion to multiuser scenario.

### 2.2.1 Point-to-Point Capacity with Full Transmitter CSI

Consider a single user wireless channel depicted in Figure 2.1. We assume that full CSI is available at the transmitter. Let  $P_n$  denote the transmission power in slot  $n$ . Let

---

<sup>3</sup>Typically, the coherence time corresponds to several thousand symbol transmission times.

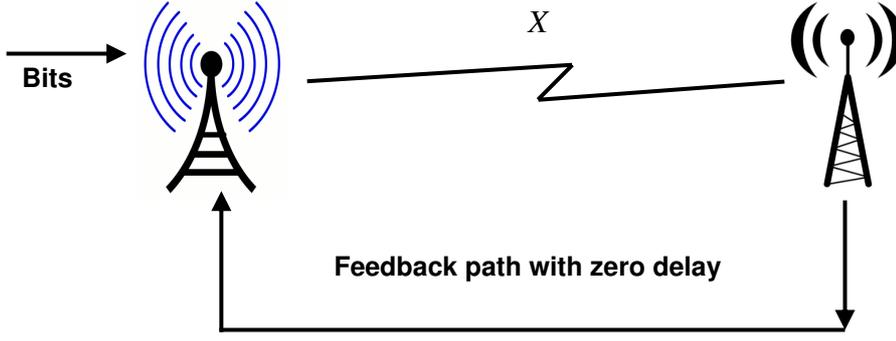


Figure 2.1: Point-to-point transmission model

$H_1 = h_1, \dots, H_M = h_M$  be a given realization of channel gains. We assume that the transmitter has an average power constraint of  $\bar{P}$ . This restriction on the average power expenditure makes the problem of achieving capacity to be a power allocation problem. It can be framed as the following optimization problem: determine a power allocation policy that maximizes the information transmission rate (and hence achieves capacity), while keeping the average power expenditure below the prescribed limit. The problem can be stated as [20, 32]:

$$\max_{P_1, \dots, P_M} \frac{1}{M} \sum_{n=1}^M \log \left( 1 + \frac{P_n |h_n|^2}{N_0} \right), \quad (2.8)$$

subject to,

$$\frac{1}{M} \sum_{n=1}^M P_n = \bar{P}. \quad (2.9)$$

Let  $x^+$  denote  $\max(0, x)$ . A solution to the optimization problem stated in (2.8) and (2.9) is a policy referred to as the *waterfilling* power allocation policy [20, 32], i.e., the optimal power in  $n^{\text{th}}$  slot is given by:

$$P_n^* = \left( \frac{1}{\lambda} - \frac{N_0}{|h_n|^2} \right)^+, \quad (2.10)$$

where  $\lambda$  satisfies:

$$\frac{1}{M} \sum_{n=1}^M \left( \frac{1}{\lambda} - \frac{N_0}{|h_n|^2} \right)^+ = \bar{P}. \quad (2.11)$$

As  $M \rightarrow \infty$ , by ergodicity,

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \left( \frac{1}{\lambda} - \frac{N_0}{|h_n|^2} \right)^+ = \mathbf{E} \left[ \left( \frac{1}{\lambda} - \frac{N_0}{|H|^2} \right)^+ \right] = \bar{P}, \quad (2.12)$$

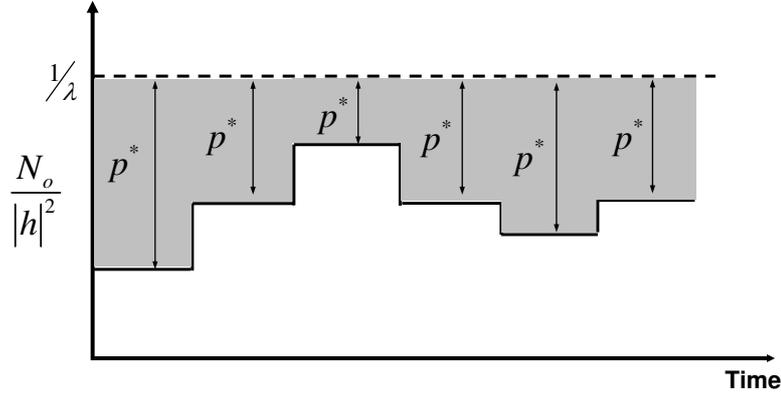


Figure 2.2: Waterfilling power allocation

where the expectation is taken with respect to the stationary distribution of the channel gains. Since we assume that the channel gain process is i.i.d., it is stationary. Hence, for a given realization of the channel gain  $H = h$ , the power allocation policy can be expressed as:

$$P^*(h) = \left( \frac{1}{\lambda} - \frac{N_0}{|h|^2} \right)^+. \quad (2.13)$$

Observe that a power allocation policy is a rule that determines the transmission power depending on the channel gain  $H = h$  or channel state  $X = x$ . Figure 2.2 provides a pictorial description of the waterfilling power allocation policy. It can be observed that the transmitter allocates more power when the channel is good and less power when the channel is poor. This insight has been used later while designing scheduling schemes at higher layer in Section 2.3.4. Note that the waterfilling power allocation is in contrast with the traditional power control policy which attempts to invert the channel.

Once the optimal power allocation is known, it is easy to calculate the channel capacity. The channel capacity with full CSI at the transmitter can be expressed as [20, 32]:

$$C = \mathbf{E} \left[ \log \left( 1 + \frac{P^*(H)|H|^2}{N_0} \right) \right]. \quad (2.14)$$

Though a point-to-point or single user transmission system offers significant insight for transmission over a fading channel, it, nevertheless, represents a restricted scenario. In the next section, we consider a more realistic multiuser scenario where we review generalization of the single user waterfilling power allocation.

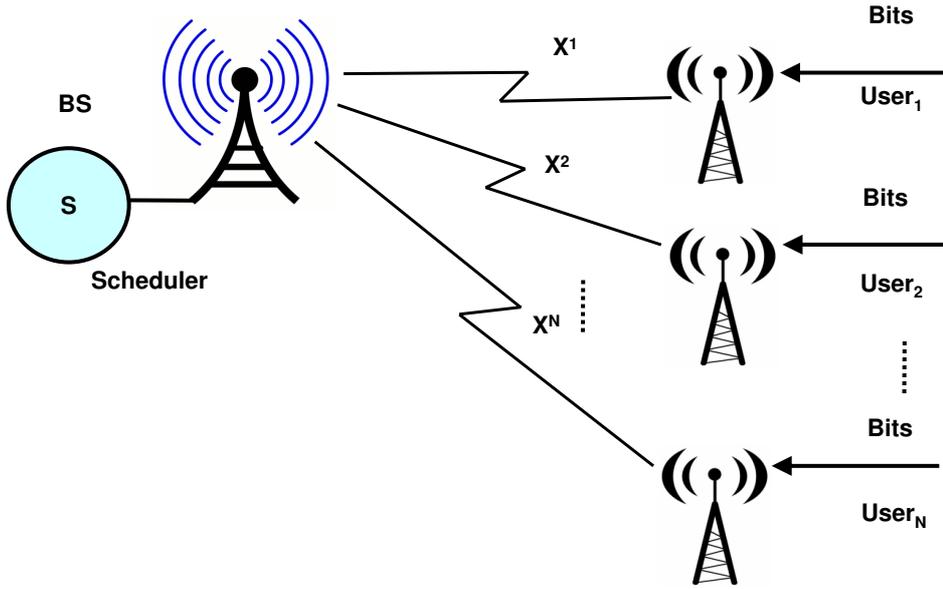


Figure 2.3: Uplink transmission model, infinite backlog of bits at transmitters

## 2.2.2 Multiuser Capacity with Full Transmitter CSI on the Uplink

In this section, our objective is to determine the ergodic capacity for a multiuser (uplink) fading channel as depicted in Figure 2.3 where  $N$  users communicate with a base station. With the block fading model, the signal  $Y_n$  received by the base station in slot  $n$  can be described in terms of the transmitted signals  $\chi_n^i$ ,  $i = 1, \dots, N$  as:

$$Y_n = \sum_{i=1}^N H_n^i \chi_n^i + Z_n, \quad (2.15)$$

where  $H_n^i$  is the channel gain for user  $i$  in slot  $n$ . Let  $H_1^i = h_1^i, \dots, H_M^i = h_M^i$ ,  $i = 1, \dots, N$ , be a given realization of the channel gains. We first consider the notion of ergodic sum capacity, i.e., maximum of the sum of information rates of all users. Under the assumption that each user  $i$  has an average power constraint of  $\bar{P}^i$  ( $\bar{\mathbf{P}} = [\bar{P}^1, \dots, \bar{P}^N]^T$  being the average power constraint vector), the problem is to determine an optimal multiuser power allocation policy that maximizes the sum of information transfer rates of all users subject to maintaining their average power expenditures below prescribed limits. This problem can be stated as:

$$\max_{P_n^i, i=1, \dots, N, n=1, \dots, M} \frac{1}{M} \sum_{n=1}^M W \log \left( 1 + \frac{\sum_{i=1}^N P_n^i |h_n^i|^2}{W N_0} \right), \quad (2.16)$$

subject to the per user power constraint:

$$\frac{1}{M} \sum_{n=1}^M P_n^i = \bar{P}^i, \quad i = 1, \dots, N. \quad (2.17)$$

We first determine the power allocation policy for the *symmetric* scenario where all users have identical channel statistics and power constraints ( $\bar{P}^i = \bar{P}, \forall i$ ). For simplicity, instead of individual power constraints as in (2.17), consider the total power constraint:

$$\frac{1}{M} \sum_{n=1}^M \sum_{i=1}^N P_n^i = N\bar{P}. \quad (2.18)$$

The sum rate in  $n^{\text{th}}$  slot is:

$$W \log \left( 1 + \frac{\sum_{i=1}^N P_n^i |h_n^i|^2}{WN_0} \right). \quad (2.19)$$

This quantity is maximized by allocating the entire power, for slot  $n$ , to the user with the best channel gain [21]. Thus, the solution of the optimization problem in (2.16) and (2.18) is that only one user with the best channel gain is allowed to transmit in a slot. The best user then performs waterfilling power allocation in a manner analogous to that in the point-to-point case (Section 2.2.1). The power allocation policy is expressed as [21]:

$$P_n^{i*} = \begin{cases} \left( \frac{1}{\lambda} - \frac{WN_0}{\max_i |h_n^i|^2} \right)^+ & \text{if } |h_n^i| = \max_i |h_n^i|, \\ 0 & \text{otherwise,} \end{cases} \quad (2.20)$$

where  $\lambda$  is chosen to satisfy the sum power constraint (2.18). Taking  $M \rightarrow \infty$  and by ergodicity of the fading process, we obtain the capacity-achieving power allocation policy that allocates power  $P^{i,*}(\mathbf{h})$  to the user  $i$  as a function of the joint channel gain vector  $\mathbf{h} = (h^1, \dots, h^N)$  where:

$$P^{i,*}(\mathbf{h}) = \begin{cases} \left( \frac{1}{\lambda} - \frac{WN_0}{\max_i |h^i|^2} \right)^+ & \text{if } |h^i|^2 = \max_i |h^i|^2, \\ 0 & \text{otherwise,} \end{cases} \quad (2.21)$$

where  $\lambda$  is chosen to satisfy the power constraint:

$$\sum_{i=1}^N \mathbf{E} [P^{i,*}(\mathbf{H})] = N\bar{P}. \quad (2.22)$$

The resulting sum capacity is:

$$C_{sum} = \mathbf{E} \left[ W \log \left( 1 + \frac{P^{i,*} |H^{i*}|^2}{WN_0} \right) \right], \quad (2.23)$$

where  $i^*(\mathbf{h})$  is an index of user with the best channel when the joint channel gain vector is  $\mathbf{h}$ . Note that this result is derived by imposing a total power constraint (2.18). However, because of symmetry and independence between the user channel gain process, the power consumption of all users is same under the optimal power allocation policy. Hence, the per user power constraints in (2.17) are automatically satisfied.

The above scheduling policy where the user with the best channel is scheduled in a slot is called *opportunistic* scheduling [39]. It takes advantage of *multiuser diversity* in order to improve the sum rate (throughput), i.e., in a system with large number of users having independent and diverse channel gains, there exists a user having a good channel gain with high probability [20]. Moreover, this probability increases with the number of users. The implications of opportunistic scheduling have been investigated in further detail in Section 2.3.2.

Before considering the more general case of asymmetric fading and power constraints, we consider a power allocation policy for a fixed channel gain  $\mathbf{h}$ . Under this condition, a power allocation policy induces a multi-access capacity region  $\mathbb{C}_g(\mathbf{h}, \mathcal{P}(\mathbf{h}))$  which denotes the set of achievable rates under that policy and can be expressed as [33]:

$$\mathbb{C}_g(\mathbf{h}, \mathbf{P}) = \left\{ \mathbf{R} : \sum_{i \in S} R^i \leq W \log \left( 1 + \frac{\sum_{i \in S} h^i P^i}{W N_0} \right) \forall S \in \{1, \dots, N\} \right\}. \quad (2.24)$$

A power allocation policy  $\mathcal{P}$  is *feasible* if it satisfies the power constraints of all users, i.e.,  $\mathbf{E}[\mathcal{P}(\mathbf{H})] = \bar{\mathbf{P}}$ . Let  $\mathbb{F}$  be the set of all feasible power allocation policies. The *throughput capacity region* is defined as the union of the set of rates achievable under all power control policies  $\mathcal{P} \in \mathbb{F}$ , i.e.,

$$\mathbb{C}(\bar{\mathbf{P}}) = \bigcup_{\mathcal{P} \in \mathbb{F}} \mathbf{E}[\mathbb{C}_g(\mathbf{H}, \mathcal{P}(\mathbf{H}))]. \quad (2.25)$$

In a general case of asymmetric channels and power constraints, sum rate maximization may not serve as an appropriate objective. Instead, one may be interested in *weighted* rate maximization. Let  $\boldsymbol{\gamma} = [\gamma^1, \dots, \gamma^N]^T$  be a vector of weights assigned to the users. The weighted rate maximization problem can be expressed as:

$$\max \boldsymbol{\gamma} \cdot \mathbf{R}, \quad (2.26)$$

subject to the constraint that the rate vector lies in the capacity region:

$$\mathbf{R} \in \mathbb{C}(\bar{\mathbf{P}}). \quad (2.27)$$

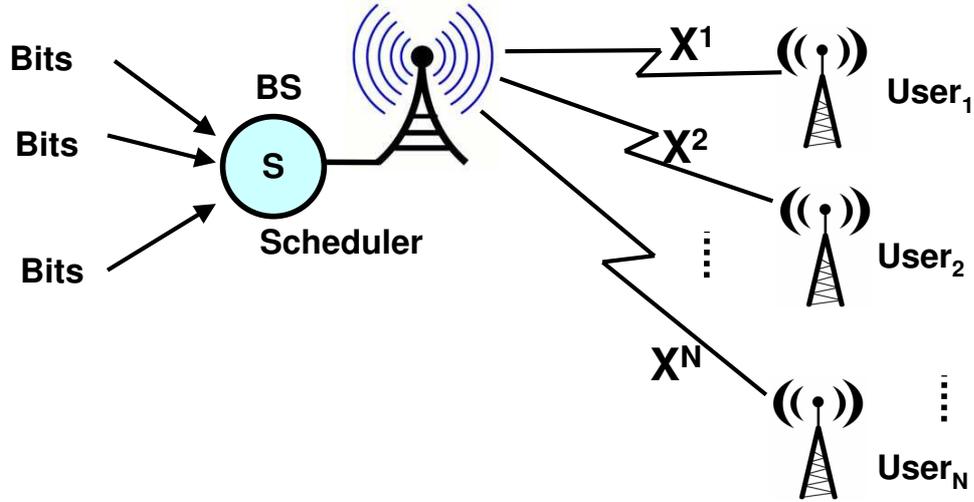


Figure 2.4: Downlink transmission model, infinite backlog of bits at base station for each user

Using a Lagrangian formulation [40], it can be shown that the optimal power allocation policy can be computed by solving, for each channel gain vector  $\mathbf{H} = \mathbf{h}$ , the following optimization problem [33]:

$$\max_{\mathbf{R}, \mathbf{P}} \gamma \cdot \mathbf{R} - \lambda \cdot \mathbf{P}, \quad (2.28)$$

subject to:

$$\mathbf{R} \in \mathbb{C}_g(\mathbf{H}, \mathbf{P}). \quad (2.29)$$

Once an optimal power allocation policy is determined, it induces a multi-access capacity region which is a set of rates. One, therefore, needs a *rate allocation* policy that maximizes the quantity in (2.26) depending on the joint channel gain vector. The optimal solution to (2.28) thus provides a power allocation  $\mathcal{P}(\mathbf{h})$  and a rate allocation  $\mathcal{R}(\mathbf{h})$  at channel gain vector  $\mathbf{H} = \mathbf{h}$ . If the choice of  $\boldsymbol{\lambda} = [\lambda^1, \dots, \lambda^N]^T$  ensures that the power constraint is met then  $\mathbf{R}^* = \mathbf{E}[\mathcal{R}(\mathbf{H})]$  is an optimal solution to (2.28). In [33], the authors show that the optimal solution to (2.28) is a greedy *successive decoding* scheme where the users are decoded in an order that is dependent on the interference experienced by them.

### 2.2.3 Multiuser Capacity with Full Transmitter CSI on the Downlink

In this section, we consider the downlink channel as depicted in Figure 2.4 where a base station transmits to  $N$  users. As in the previous section, we consider a block fading model. If  $\chi_n$  is the signal transmitted by the base station in slot  $n$ , then the signal  $Y_n^i$  received by the user  $i$  in slot  $n$  can be expressed as:

$$Y_n^i = H_n^i \chi_n + Z_n^i, \quad (2.30)$$

where  $Z_n^i$  is the AWGN at user  $i$ .  $\{H_n^i\}$  is an ergodic fading process for user  $i$ . We assume that the base station has the CSI for all  $N$  users in each slot. The policy that maximizes the sum capacity is again the one that transmits to the best user in each time slot. The power allocated to such a user depends on the average power constraint. Under this policy, the channel can be viewed as a point-to-point channel with channel gain distributed as:

$$\max_{i=1,\dots,N} |H^i|^2. \quad (2.31)$$

The optimal power allocation is the waterfilling power allocation:

$$P^*(\mathbf{h}) = \left( \frac{1}{\lambda} - \frac{N_0 W}{\max_{i=1,\dots,N} |H^i|^2} \right)^+, \quad (2.32)$$

where  $\mathbf{h} = [h^1, \dots, h^N]^T$  is the joint channel gain vector and  $\lambda$  is chosen to satisfy the average power constraint. The sum capacity of the downlink can be expressed as:

$$C_{sum}^b = \mathbf{E} \left[ \log \left( 1 + \frac{P^*(\mathbf{h})(\max_{i=1,\dots,N} |H^i|^2)}{N_0 W} \right) \right]. \quad (2.33)$$

It should be noted that the capacity region on the downlink  $\mathbb{C}_b(\mathbf{h}, P)$  for a channel gain vector  $\mathbf{h} = [h^1, \dots, h^N]^T$  and transmit power  $P$  is very different from that of the uplink case [41, 42].

### 2.2.4 Towards a Framework for Cross Layer Scheduling

In the above sections, we have primarily reviewed information theoretic limits on the capacity of fast fading channel when the transmitter has full CSI and there is a constraint on average power. Some of the key insights obtained from these results that have a bearing on the scheduling at higher layers, can be summarized as follows:

- For a single user fading channel, the channel capacity under the constraint on average power can be maximized by the ‘water-filling’ power allocation over the fading states. This suggests that we should transmit more information in good channel states and less in bad channel states in order to maximize the long term average throughput.
- In a multiuser case, the sum capacity can be maximized by Time Division Multiple Access (TDMA) type mechanism where only one user, that has the best channel state, is scheduled in a slot. The power allocation is again ‘waterfilling’. This suggests that users should transmit at *opportunistic time*.

In wireless communications, apart from maximizing throughput under average power constraint, energy efficiency is also an equally important concern. Indeed, as it turns out that since the power required to transmit ‘reliably’ at a particular rate is a strictly convex function of the rate for a given fading state; transmission at lower rates can result in energy savings. This provides us the following insight:

- For energy efficiency, user should transmit data in *opportunistic chunks*.

Note that the above information theoretic results are derived without considering random packet arrivals and queueing at higher layers of the protocol stack. However, in a typical packet based communication, packets (hence bits) arrive randomly and may be subjected to buffering at MAC or network layer. Hence, to maximize throughput (or to minimize energy), scheduling users opportunistically (or scheduling data in opportunistic chunks), has to contend with MAC or network layer issues such as fairness, packet delay and queue stability.

We must remark here that in practice, it can be assumed that the channel coherence times are reasonably long so that ‘near’ capacity rates can be achieved by employing practically implementable codewords and yet the coherence times are smaller than the ‘packet delay time-scale’ of interest at the MAC/network layer. This allows us to formulate the scheduler at MAC/network layer as a controller which exploits fading state information to maximize (or minimize) a given utility function (such as throughput, delay, energy) subject to some constraints such as fairness, average packet delay, stability. Before we review these formulations, at this point, it may be appropriate to describe the system

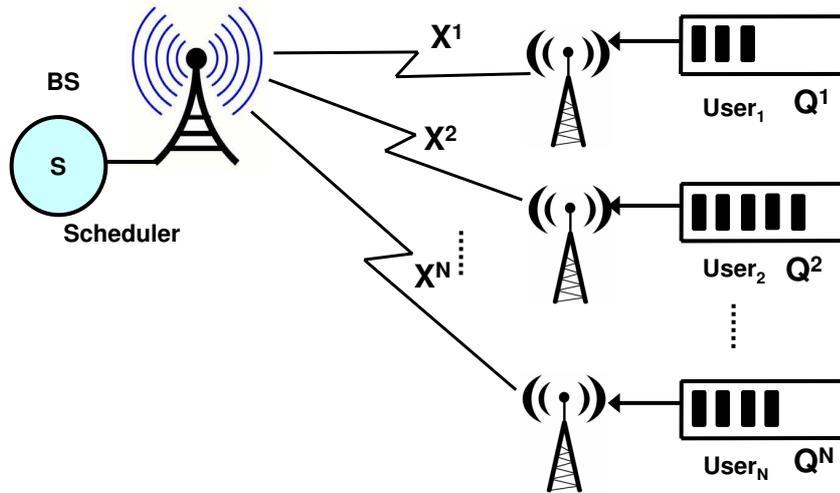


Figure 2.5: Uplink transmission model, finite buffer at each user

model in more detail for our study. Depending on the system model, the scheduler can be either centralized or distributed. Accordingly, we consider both centralized as well as distributed scheduling. We first consider centralized scheduling. In the subsequent section, we review distributed scheduling.

## 2.3 Multiuser Diversity with Centralized Scheduling

### 2.3.1 System Model

We consider a multiuser wireless system where  $N$  users communicate with a base station. This may correspond to a single cell IEEE 802.16 or any other cellular system. On the uplink, as depicted in Figure 2.5, users communicate with the base station using TDMA, i.e., time is divided into slots of equal duration and only one user can transmit in a slot<sup>4</sup>. The base station is the centralized entity that makes the scheduling decision and the user scheduled by the base station transmits in a slot. We assume that packets arrive randomly into the user MAC buffer. Packets are queued in the buffer until they are transmitted. Assumptions regarding the fading process are same as those in Section 2.2.2.

<sup>4</sup>Note that the assumption of TDMA does not restrict the applicability of our formulation in subsequent chapters of the thesis. For most formulation, (with some exception as exemplified later) the discussion is applicable to any orthogonal multichannel system.

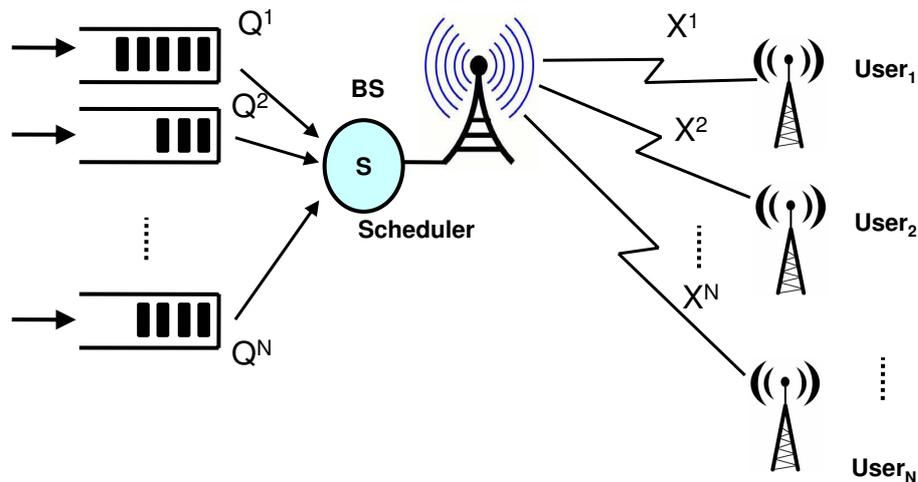


Figure 2.6: Downlink transmission model, finite buffer for each user at base station

On the downlink, as depicted in Figure 2.6, we assume that the base station multiplexes the transmissions corresponding to the  $N$  users using Time Division Multiplexing (TDM). The base station maintains a virtual queue for the user data. Assumptions regarding the arrival process are same as those for the uplink case. Assumptions regarding the fading process are same as those in Section 2.2.3.

Let  $Q_n^i \in \mathbb{Q}$  denote the queue length (in bits) corresponding to user  $i$  in slot  $n$ . Let  $A_n^i$  denote the number of bits arrived into user  $i$  buffer in slot  $n$ . Let  $U_n^i$  be the number of bits transmitted from user  $i$  buffer in slot  $n$ . Since the slot duration is normalized to 1,  $U_n^i$  also denotes the transmission rate of user  $i$  in slot  $n$ . Since the scheduler can at most schedule all the bits in a buffer in any slot,  $U_n^i \leq Q_n^i$ . The queue evolution equation for user  $i$  can be written as:

$$Q_{n+1}^i = \max(0, Q_n^i + A_{n+1}^i - I_n^i U_n^i). \quad (2.34)$$

where  $I_n^i$  is an indicator variable that is set to 1 if the user  $i$  is scheduled in slot  $n$ , otherwise it is set to 0.

Different formulations make various assumptions on the arrival process  $\{A_n^i\}$  and control action process  $\{U_n^i\}$ . For our investigation, these assumptions are stated in later chapters. The average queue length of a user  $i$  over a long period of time can be expressed

as:

$$\bar{Q}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M Q_n^i. \quad (2.35)$$

Average delay  $\bar{D}^i$  can be treated to be equivalent to the average queue length  $\bar{Q}^i$  because of the Little's law [10]:

$$\bar{Q}^i = \bar{a}^i \bar{D}^i, \quad (2.36)$$

where  $\bar{a}^i$  is the average arrival rate. Hence, one can think of minimizing the average queue length instead of the average delay.

Similarly, the sum throughput over a long period of time can be expressed as:

$$\bar{T} = \liminf_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \sum_{i=1}^N I_n^i U_n^i. \quad (2.37)$$

Let  $P(X_n^i, I_n^i U_n^i)$  denote the power transmitted by user  $i$  in slot  $n$ . Note that, for a given  $x$ , the transmission power  $P(x, u)$  is an increasing and strictly convex function of  $u$ . The average power consumed by a user  $i$  over a long period of time can be expressed as:

$$\bar{P}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M P(X_n^i, I_n^i U_n^i). \quad (2.38)$$

In this thesis, we have primarily focused on the above three performance measures. A number of scheduling algorithms have been proposed in the literature that focus on similar measures or variations of these. Broadly, various scheduling algorithms studied in the literature can be classified into three types:

1. *Maximize sum throughput subject to fairness constraint: Throughput-Fairness trade-off.*
2. *Maximize sum throughput subject to delay and queue stability constraint: Throughput-Delay tradeoff.*
3. *Minimize average power subject to delay constraint: Power-Delay tradeoff.*

In the following sections, we review representative literature for each of the above types and discuss some of the limitations of the existing formulations.

### 2.3.2 Throughput-Fairness Tradeoff

Exploiting multiuser diversity in an opportunistic manner by scheduling the user with the best channel gain as discussed in the previous section might introduce unfairness at the higher layers. Users who are closer to the base station might experience perennially better channel conditions and thereby obtain a higher share of the system resources at the expense of users who are farther away from the base station. On the other hand, scheduling users with poor channel gains results in a reduction in the overall achievable sum throughput. Thus, there exists a fairness-sum throughput tradeoff. One of the earliest systems to exploit this tradeoff in order to improve the sum throughput is the Code Division Multiple Access/High Data Rate (CDMA/HDR) system [43].

Different scheduling schemes admit different notions of fairness and achieve varying sum throughput. We first provide details regarding the intervals over which fairness is provided. We, then, review various notions of fairness considered in the literature. Subsequently, we review scheduling algorithms based on these considerations.

#### Fairness Interval

Different scheduling algorithms provide fairness over different time intervals. A scheduling algorithm is *long term* fair if it provides a *fair share* (meaning of fair share depends on the notion of fairness considered later) of a certain quantity such as fraction of time slots or throughput to all users over a long period of time. As outlined earlier, the average throughput achieved by a user  $i$  over a long period of time can be expressed as:

$$\bar{T}^i = \liminf_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M I_n^i U_n^i. \quad (2.39)$$

The fraction of slots allocated to a user in the long run can be expressed as:

$$\bar{I}^i = \liminf_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M I_n^i. \quad (2.40)$$

Long term fair algorithms allocate the quantities such as  $\bar{T}^i$  and  $\bar{I}^i \forall i$  in a fair manner over a long period of time.

On the other hand, a scheduling algorithm is *short term* fair if it provides a fair allocation of a certain quantity such as fraction of time slots or throughput to all users

in an interval of  $M$  slots. The average throughput by a user  $i$  over a window of  $M$  slots can be expressed as:

$$T^i(M) = \frac{1}{M} \sum_{n=1}^M I_n^i U_n^i. \quad (2.41)$$

The fraction of slots allocated to a user  $i$  in a window of  $M$  slots can be expressed as:

$$I^i(M) = \frac{1}{M} \sum_{n=1}^M \sum_{i=1}^N I_n^i. \quad (2.42)$$

Short term fair algorithms allocate the quantities such as  $T^i(M)$  and  $I^i(M) \forall i$  in a fair manner over a window of  $M$  slots.

### Notions of Fairness

In this section, we discuss some notions of fairness that have been proposed in the literature. A more general review of the various other notions of fairness not considered here and the related fair scheduling algorithms can be found elsewhere, e.g., Chapter 8 of [44].

All the notions of fairness can be used to provide fairness either over the long run or over a short interval of time. Let  $\phi = [\phi^1, \dots, \phi^N]^T$  be a weight vector associated with the users indicating their relative priorities.

- **Minimum Allocation:** Under this notion of fairness, the scheduling scheme attempts to provide a certain minimum throughput or fraction of time slots to each user. Let  $\bar{\Psi} = [\bar{\Psi}^1, \dots, \bar{\Psi}^N]^T$  be a vector indicating certain minimum throughput that must be achieved by the users. Let  $\bar{\epsilon} = [\bar{\epsilon}^1, \dots, \bar{\epsilon}^N]^T$  be a vector indicating minimum fraction of time slots that must be allocated to a user. Then the scheme is said to be fair if  $\bar{\mathbf{T}} \geq \bar{\Psi}$  (minimum throughput allocation) or  $\bar{\mathbf{I}} \geq \bar{\epsilon}$  (minimum time slot allocation).
- **Fair Relative Throughput/Time Slot Allocation:** The system attempts to provide equal weighted throughput/fraction of time slots to all users under this notion of fairness. The scheme is said to be fair if  $\frac{\bar{T}^i}{\phi^i} = \frac{\bar{T}^j}{\phi^j}, \forall i, j$  (fair relative throughput allocation) or  $\frac{\bar{I}^i}{\phi^i} = \frac{\bar{I}^j}{\phi^j}, \forall i, j$  (fair relative time slot allocation).
- **Proportional Fair Allocation:** The fraction of slots allocated to a user is proportional to the average channel gain of that user. Better the channel gain perceived by a user

on an average, higher is the fraction of slots allocated to such a user. Note that this is an intuitive definition of proportional fair allocation for the fading channel. For a more general definition of proportional fairness and proportional fair scheduling, the reader is referred to Chapter 8 of [44] and the references therein.

Note that each notion of fairness defined above can have a probabilistic extension, where the system is allowed to be unfair with a certain probability.

### Fair Scheduling Algorithms

The proportional fair scheduler [45, 46] allocates time slots to the users according to the proportional fairness criterion. The algorithm is fair over the long run. Let  $T_n^i$  be the average throughput of a user  $i$  in an exponentially averaged window of length  $t_c$ . The scheduling algorithm schedules the user  $k$  in a slot  $n$  where:

$$k = \arg \max_i \frac{U_n^i}{T_n^i}. \quad (2.43)$$

The average throughput  $T_n^i$  is updated using exponential averaging:

$$T_{n+1}^i = \begin{cases} (1 - \frac{1}{t_c})T_n^i + (\frac{1}{t_c})U_n^i, & i = k, \\ T_n^i & i \neq k. \end{cases} \quad (2.44)$$

Users having the same channel statistics tend to have the same average throughput and consequently the scheduling policy reduces to the opportunistic policy, i.e., in each slot schedule the user with the highest rate. On the other hand, if the channel statistics of the users are not identical, then the users compete for resources based on their rates normalized by their respective throughput. Note that the algorithm schedules a user when its channel gain is high relative to its own average channel gain over the time scale  $t_c$ , i.e., data is transmitted to a user when the channel is near its own peak. The proportional fair scheduler has the following property: For large  $t_c$ , i.e., for  $t_c \rightarrow \infty$ , the algorithm maximizes  $\sum_{i=1}^N \log \bar{T}^i$ .

Long term sum throughput maximization subject to providing minimum throughput or fraction of slots to users has been variously considered in [39, 47, 48]. In [39], the objective is to determine a scheduling policy that maximizes the sum throughput while providing minimum fraction of time slots to the users. This optimization problem can be

expressed as [39]:

$$\max \bar{T} \tag{2.45}$$

subject to:

$$\bar{T}^i \geq \bar{\epsilon}^i, \quad i = 1, \dots, N. \tag{2.46}$$

In [39], the authors propose a scheduling policy based on stochastic approximation as a solution to this problem and prove the optimality of the policy.

In [49], the authors consider sum throughput maximization subject to providing minimum throughput in the long run. Moreover, they consider the probabilistic extension to the above notion of fairness. Furthermore, they propose algorithms based on stochastic approximation for providing the fairness guarantees.

The opportunistic scheduling problem with short term fairness constraints (under the minimum time slot allocation criterion) can be expressed as the following optimization problem: in any window of  $M$  slots,

$$\max \sum_{i=1}^N \bar{T}^i(M), \tag{2.47}$$

subject to:

$$\bar{T}^i(M) \geq M\bar{\epsilon}^i. \tag{2.48}$$

In [50], the authors formulate the above problem and propose a heuristic policy that provides a high sum throughput.

### 2.3.3 Throughput-Delay Tradeoff

In the preceding section, we have considered scheduling algorithms that attempt to maximize sum throughput subject to fairness constraints. In this section, we first study scheduling algorithms that consider queue stability as a notion of QoS. While some of these algorithms are throughput optimal, they do not necessarily ensure small average queue lengths and hence small delays. Subsequently, we consider scheduling algorithms that address this issue while achieving high sum throughput.

### Throughput Optimal Scheduling

We first review feasible rate and power allocation with an objective of stabilizing the queues of the users. We define the overflow function as follows:

$$f^i(\xi) = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M I_{Q_n^i > \xi}, \quad (2.49)$$

where  $I_{Q_n^i > \xi}$  is an indicator variable that is set to 1 if  $Q_n^i > \xi$ , else it is set to 0. We say that the system is *stable* if  $f^i(\xi) \rightarrow 0$  as  $\xi \rightarrow \infty$  for all  $i = 1, \dots, N$ . Let  $\bar{\mathbf{a}} = [\bar{a}^1, \dots, \bar{a}^N]^T$  denote the arrival vector,  $\bar{a}^i$  being the average arrival rate for user  $i$ . In this section, in addition to the average power constraints, we also consider the peak power constraints, i.e., a user  $i$  can transmit at a maximum power  $\hat{P}^i$  in any slot. Let  $\hat{\mathbf{P}} = [\hat{P}^1, \dots, \hat{P}^N]^T$  denote the peak power constraint vector.

Note that, since the objective is to keep the queues stable, the power and rate allocation policies have to be cognizant of the queue lengths of the users in each slot. We, therefore, extend the definitions of power and rate allocation policies. A power allocation policy  $\mathcal{P}$  is a mapping from the joint channel gain and queue length vector  $(\mathbf{h}, \mathbf{q})$  to a power allocation vector  $\mathbf{P}$ . A rate allocation policy  $\mathcal{R}$  is a mapping from the joint channel gain and queue length vector  $(\mathbf{h}, \mathbf{q})$  to a rate allocation vector  $\mathbf{R}$ . As noted previously in Section 2.2.2, a feasible rate allocation policy allocates rates within the multi-access capacity region  $\mathbb{C}_g(\mathbf{h}, \mathbf{P})$ . The *stability region* of the multi-access system is the set of all arrival vectors  $\bar{\mathbf{a}}$  for which there exists some feasible power allocation policy and rate allocation policy under which the system is stable. The stability region of a multi-access system can be shown to be given by [51]:

$$\mathbb{C}_s(\bar{\mathbf{P}}, \hat{\mathbf{P}}) = \bigcup_{\mathcal{P} \in \mathbb{F}} \mathbf{E}[\mathbb{C}_g(\mathbf{H}, \mathcal{P}(\mathbf{H}))]. \quad (2.50)$$

Note that the power control policy  $\mathcal{P}(\mathbf{H})$  depends only on the channel gain vector  $\mathbf{H}$ . More importantly, this stability region of the multi-access system is same as the throughput capacity region under power control defined in Section 2.2.2.

If the joint arrival process  $\{\mathbf{A}_n\}$  and joint channel gain process  $\{\mathbf{H}_n\}$  are ergodic Markov chains, then the system can be stabilized by a power and rate allocation policy if  $\bar{\mathbf{a}} \in \mathbb{C}_s(\bar{\mathbf{P}}, \hat{\mathbf{P}})$ . In practice, one does not have a knowledge of the arrival vector  $\bar{\mathbf{a}}$  and this can only be estimated over time. Power and rate allocation policies that do not assume

knowledge of the arrival vector  $\bar{\mathbf{a}}$  and stabilize the system as long as  $\bar{\mathbf{a}} \in \mathbb{C}_s(\bar{\mathbf{P}}, \hat{\mathbf{P}})$  are referred to as *throughput optimal* policies. Throughput optimal scheduling policies have been explored in [33, 52]. Longest Connected Queue (LCQ) [53], Exponential (EXP) [54], Longest Weighted Queue Highest Possible Rate (LWQHPR) [55] and Modified Longest Weighted Delay First (M-LWDF) [16] are some other throughput optimal scheduling policies. We now review some of these scheduling rules that are throughput-optimal under a power allocation policy  $\mathcal{P}$ .

- LWQHPR: Let  $\boldsymbol{\alpha} = [\alpha^1, \dots, \alpha^N]^T$  be a vector of weights. The throughput optimal rate allocation policy is obtained by maximizing  $\sum_{i=1}^N \alpha^i Q_n^i U_n^i$  over  $\mathbb{C}_g(\mathbf{h}, \mathcal{P}(\mathbf{h}))$ . The solution  $\mathbf{r}^*$  is obtained by successively decoding the users in an increasing order of their weights  $\alpha^i Q_n^i$ , i.e., shorter queues are decoded before longer queues. This implies that longer queues are given preference over shorter queues.
- M-LWDF: Let  $\bar{\Upsilon}^i$  and  $\bar{D}^i$  be the delay requirement and achieved delay for user  $i$  respectively. The M-LWDF scheduler attempts to satisfy the delay constraints of the form,

$$Pr(\bar{D}^i > \bar{\Upsilon}^i) \leq \bar{\rho}^i, \quad (2.51)$$

where  $\bar{\rho}^i$  is an upper bound on the probability with which  $\bar{D}^i$  is allowed to exceed  $\bar{\Upsilon}^i$ . The M-LWDF scheme achieves this by scheduling a user  $i$  in a slot  $n$  where [16]:

$$\frac{-\log(\bar{\rho}^i) \times Q_n^i \times U_n^i}{\bar{\Upsilon}^i} = \max_j \frac{-\log(\bar{\rho}^j) \times Q_n^j \times U_n^j}{\bar{\Upsilon}^j}. \quad (2.52)$$

Note that higher the queue length and better the channel gain (and hence higher the rate) of a user in a slot, higher is the probability of scheduling the user in the slot.

- EXP: Let  $\boldsymbol{\gamma} = [\gamma^1, \dots, \gamma^N]^T$ ,  $\mathbf{b} = [b^1, \dots, b^N]^T$  be an arbitrary set of positive constants. Let  $\beta$  and  $\eta \in (0, 1)$  be fixed. The Exponential (EXP) rule schedules a user  $i$  in a slot  $n$  where [54]:

$$i = \arg \max_j \gamma^j U_n^j \exp\left(\frac{b^j Q_n^j}{\beta + [\hat{Q}_n]^\eta}\right), \quad (2.53)$$

where  $\hat{Q}_n \triangleq \frac{1}{N} \sum_{i=1}^N b^i Q_n^i$ . Thus, a user with better channel gain and hence higher rate and higher queue length has a higher probability of being scheduled.

### Delay Optimal Scheduling

While Throughput optimal scheduling policies maintain the stability of the queueing system, they do not necessarily guarantee small queue lengths and consequently lower delays. Delay optimal scheduling deals with optimal rate and power control such that the average queue length and hence average delay are minimized for arrival rates within the stability region under average and peak power constraints. Due to the nature of the constraints, there is no loss of optimality in choosing the rate and power control policies separately [56]. Hence, to simplify the problem, one can choose any stationary power control policy that satisfies the peak and average power constraints. The delay optimal policy, therefore, deals with optimal rate allocation for minimizing delays under a given power allocation policy. The objective is to maximize a weighted combination of the rates expressed in (2.26), while at the same time minimizing the achieved delay  $\bar{Q}^i$ ,  $i = 1, \dots, N$ . Note that this problem is a multi-objective optimization problem [57, 58].

We now study a scheme that is throughput optimal and delay optimal under certain assumptions on the arrival and channel gain processes for both multi-access (uplink) and broadcast (downlink) channels. Before outlining these assumptions, we define a symmetric channel gain process. The channel gain process is called *symmetric* or *exchangeable* if for all  $n$  and  $\mathbf{h} = [h^1, \dots, h^N]^T$  in the channel gain space  $\mathbb{H}^N$ ,

$$Pr(H_n^1 = h^1, \dots, H_n^N = h^N) = Pr(H_n^1 = h^{\pi(1)}, \dots, H_n^N = h^{\pi(N)}), \quad (2.54)$$

for any permutation  $\pi \in \Pi$ , where  $\Pi$  is the set of all permutations on the set  $\{1, \dots, N\}$ . A power control policy  $\mathcal{P}$  that is a function of the channel gain vector only is symmetric if for all  $\mathbf{h} \in \mathbb{H}$ ,

$$\mathcal{P}^i(h^1, \dots, h^N) = \mathcal{P}^{\pi^{-1}(i)}(h^{\pi(1)}, \dots, h^{\pi(N)}). \quad (2.55)$$

Intuitively, under a symmetric power control policy, the power allocated to a given user is determined by the channel gain perceived by that user relative to the channel gains perceived by the other users and not on the identity of that user. In [55], the authors consider symmetric channel gain and power control. Moreover, they assume Poisson arrivals and exponentially distributed packet lengths. Under these assumptions, they prove that the Longest Queue Highest Possible Rate (LQHPR) policy, besides being through-

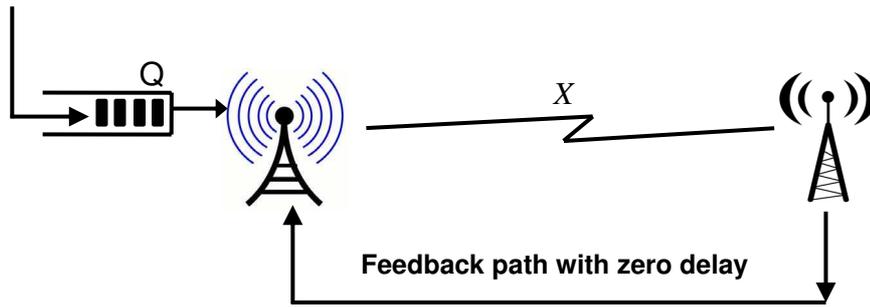


Figure 2.7: Point-to-point transmission model with finite buffer

put optimal, also minimizes delay. At the physical layer, LQHPR policy corresponds to adaptive successive decoding, whereby the user with the shortest queue is decoded first and the user with the longest queue is decoded last.

*Remark 2.1.* For a multiuser queuing system with scheduler on a TDM channel, there is an extensive literature that we have reviewed in Section 2.3.2, and Section 2.3.3. However, the specific optimization problem of maximizing the sum throughput subject to constraints on the individual user delays has not been explicitly addressed so far. We show in Chapter 5 that this problem has the structure of a Constrained Markov Decision Process (CMDP) [59]. However, the primary difficulty in computing optimal policy (as exemplified later in Chapter 5) lies in large state space size that increases exponentially with the number of users. Moreover, computation of such a policy requires knowledge of the system model, i.e., knowledge of the probability distributions of the channel state and the arrival process for each user. The exact system model is not known in practice. We believe that state space explosion and unknown system model are the primary reasons for inadequate attention towards optimal delay constrained multiuser scheduler despite abundant literature in wireless scheduling with various other performance objectives.

### 2.3.4 Power-Delay Tradeoff

Apart from providing fairness and minimizing queuing delay, minimizing energy expenditure is another important objective. In this section, we first present an energy efficient scheduling problem that exploits the power-delay tradeoff. We begin with single user scenario and then extend this formulation for multiuser scenario.

### Single User Scenario

The power required to transmit reliably at a particular rate is a convex function of the rate. Thus, transmission at lower rates can result in power savings, i.e., the scheduler should transmit data in opportunistic chunks. Moreover, data should be transmitted at an opportunistic time, i.e., when the channel condition is good. Both these power saving considerations result in higher queuing delays at the transmitter. The higher layer applications, however, have a certain QoS requirement in terms of the average delay. Thus, the objective is to determine a power/rate allocation policy that minimizes the average power expenditure subject to a constraint on the average delay experienced at the MAC layer. Note that this problem can be considered to be a dual of the problem considered in Section 2.2.1.

The system model considered in this section is depicted in Figure 2.7. The objective is to minimize the average power expenditure subject to the average queue length (equivalently the average delay from (2.36)) being below a certain threshold<sup>5</sup>, say,  $\bar{\delta}$ , i.e.,

$$\text{minimize } \bar{P} \text{ subject to } \bar{Q} \leq \bar{\delta}. \quad (2.56)$$

The scheduling policy can be considered as a control policy which decides the number of packets to be transmitted to minimize the average power expenditure subject to a given queue length constraint. This problem has the structure of a CMDP and was first addressed in the pioneering work of [27, 60]. Subsequently, other work [36, 29, 61, 30, 28, 62] has also considered this problem under various assumptions on packet arrival and channel gain processes. In [27, 60], the tradeoff between the average delay and the average power has been analyzed. The power-delay tradeoff has also been quantified in the region of asymptotically large delays. In [60], structural results for optimal policy have been derived. Structural results have also been discussed in [29] for a policy which minimizes the average delay subject to a constraint on the average power. It is proved in [29] and [60] and that there exists an optimal stationary policy which increases as the buffer occupancy increases, and decreases as the channel state goes from good to bad. What this means in physical terms is that for a fixed channel gain, the greater the queue length, the more you transmit, and for a fixed queue length, the better the channel, the more you transmit.

<sup>5</sup>Since we consider a single user we drop the superscript  $i$  in the notation.

While the existence of a stationary optimal policy for the average cost problem has been considered in [29] when the packet arrival process is i.i.d., the problem becomes much more difficult when the arrival process is Markovian. For this case, in [27], only the unconstrained average cost dynamic programming formulation has been given. Subsequently, the model of [27] and [29] has been extended in [28] where the authors consider a more general state space version of the average cost CMDP. In this model, both the arrival and the channel state process are considered to be Markovian. In [30], a discrete state space version of the same problem has been considered for correlated arrivals and correlated fading.

The fundamental problem with Dynamic Programming (DP) algorithms is that of the so called *curse of dimensionality* [31]. For moderate to large state space, techniques that numerically determine the optimal policy quickly become computationally infeasible. The structural results of the optimal policy can be exploited in order to develop efficient heuristics that are computationally less expensive and hence can be implemented in a practical system. In [36], the authors propose a simple heuristic that exploits the structural properties of the optimal policy. This policy is of ‘thresholding’ type. They also suggest a mechanism to derive the optimum thresholds for queue length and channel gain.

Interestingly, this problem has also been considered for the AWGN channel in [18, 61]. In [61], the authors show that the optimal scheduler is a convex combination of a small class of deterministic schedulers.

While all the above approaches have provided significant insights into the problem, none of them explicitly deals with the *computation* of optimal packet scheduling policy. Consequently, the practical implementation of the policy remains an important open issue. This is primarily due to the following two reasons:

1. Since the state space is large, the standard dynamic programming algorithms are hard to implement. The preferred technique for CMDP has been linear programming [59]. Combined with the more recent approach based on function approximation [63], this holds great promise. The structural results of the policy may help in the choice of basis functions in function approximation based computation. But none of the related work discussed above seems to have explored this issue in developing

an implementable optimal packet scheduling algorithm.

2. Secondly and more importantly, computation of the optimal policy using the above mentioned techniques assumes a knowledge of the underlying model. This means that knowledge of the probability distribution of both the arrival process and the channel state is necessary for the computation of optimal policy. This is usually not the case in practice. In [36], the authors have proposed a suboptimal algorithm, but even here, the computation of the appropriate thresholds seems to assume the knowledge of the arrival process and channel state distribution.

In the next section, we consider multiuser scheduling where a centralized scheduler has the responsibility of determining the user to be scheduled in a slot in addition to determining the number of bits/packets to be transmitted and the corresponding transmission power.

### Multiuser Scenario

While a lot of work has been done for exploiting the power delay tradeoff for the single user fading channel, extensions of these approaches for the multiuser fading channel are rather limited. In [60], the author explores the two user problem. The objective is to minimize a weighted combination of power expenditure of the users as well as their queue length costs. The author also provides a ‘near optimal’ algorithm for this problem. However, the algorithm is not scalable for large number of users.

Recently, in [64], the author has considered the problem of minimizing power expenditure on the downlink subject to rate constraints. The objective is to exploit the power delay tradeoff on the downlink. The author proves a bound on the achievable delay for a certain energy expenditure and proposes an algorithm called Tradeoff Optimal Control Algorithm (TOCA) that comes to within a logarithmic factor of achieving this bound.

*Remark 2.2.* Note, however, that there are two major issues with the approach of [64]. Firstly, on the downlink, average power minimization is not a major concern since the base station typically transmits at a fixed maximum power sufficient to reach the farthest user. Secondly, TOCA is exponential in the number of users and is not practically implementable even for moderate number of users.

*Remark 2.3.* On the uplink, each user would like to minimize the power it expends while obtaining a certain QoS from the base station. Thus, the problem that needs to be addressed is to minimize the average power expenditure of *each* user subject to *individual* delay constraints. This problem has not been addressed in the literature so far. Moreover, as pointed out in the single user scenario, a knowledge of the probability distributions of the channel state and arrival process of the users is not available in practice.

## 2.4 Multiuser Diversity with Distributed Scheduling

In the preceding sections, we have studied centralized schemes where a centralized scheduler residing at a base station schedules the user transmissions. In this section, we focus on *distributed* scheduling or channel access schemes where there is no centralized entity for making the scheduling decisions. The literature on distributed scheduling is vast. Recently, distributed scheduling has found applications in ad hoc networks [65]. The general problem of providing end to end QoS in such networks involves other issues such as routing which is not the focus of this thesis. Our discussion in this section is restricted to representative channel aware distributed scheduling schemes in the literature.

We first review some aspects of the ALOHA protocol [66]. Subsequently, we review distributed channel access mechanisms that attempt to exploit multiuser diversity. Opportunistic scheduling assumes CSI corresponding to all users. This introduces an additional challenge in a distributed scenario where multiple transmitters access the channel in a decentralized fashion. This is because implementing opportunistic channel access requires that the CSI corresponding to each transmitter-receiver pair be known at the transmitter.

Transmission strategies like ALOHA and its variants are distributed channel access schemes in which the users randomly access the channel. These have been widely studied in the literature [10]. Game theoretic models [67] have been applied to model the ALOHA protocol [68, 69, 70, 71]. The slotted ALOHA protocol has been modeled both as a non-cooperative game as well as a cooperative game. Various objectives like delay minimization [68], throughput maximization [68, 69, 70] have been considered. Power control coupled with retransmission control has been variously studied in [72, 73, 74]. The reader is referred to excellent reviews [73, 75] as well as a book on game theory ap-

plications in wireless communications [76] for further information on applications of game theory for modeling the random access problem.

Recently in [77], the authors consider the  $N$  user uplink scenario where the users transmit to a common base station in a distributed fashion. Each user chooses its access control policy and transmission power based on its channel gain and buffer occupancy. The objective is to maximize the long term throughput subject to average power and average delay constraints. Given the power and delay constraint of a user, its throughput depends on the actions and states of all other users. It is assumed that a user has information about its own channel gain and buffer occupancy and does not know this information about other users. The authors cast the problem within the CMDP framework and make use of the Linear Programming approach for determining the optimal policy. Moreover, the authors also provide the equilibrium analysis of the  $N$  player stochastic game. As in the case of centralized scheduling, the authors assume a knowledge of the probability distributions of the channel gain and arrival processes of all users.

Approaches for exploiting multiuser diversity in a distributed fashion have also been considered in [78, 79, 80, 81, 82]. In [78, 82], the authors attempt to exploit multiuser diversity in a distributed fashion with only local channel information, i.e., each user is aware of its own channel condition only. The authors propose a *channel aware ALOHA* protocol and provide a throughput analysis of the proposed protocol under the infinitely backlogged model. In [80], the authors propose opportunistic splitting algorithms for resolving collisions over a sequence of mini-slots and determine the user with the best channel condition. The authors provide an analysis of the throughput of the system and prove a bound on the number of mini-slots required to resolve the collisions. In [79], the authors consider symmetric as well as asymmetric fading. The authors propose a *binary* scheduling algorithm where users access the channel when the corresponding channel condition is above a certain threshold and prove that it maximizes sum throughput under symmetric fading. Moreover, for asymmetric fading, they prove that binary scheduling maximizes the sum of log of average throughput of the users and is fair in the long run. Furthermore, they also consider channels with memory and provide simple extensions of the binary scheduling algorithm. In [83], the authors consider the tradeoff between joint *channel probing*, i.e., the process of informing the transmitters regarding the channel

conditions perceived by each other, and distributed scheduling.

## 2.5 Discussion and Open Problems

In this chapter, we have seen how fading can be exploited as an opportunity for improving performance at the MAC or higher layers of the protocol stack. In recent years, there has been an explosion of research in cross layer scheduling algorithms with various objectives under different models. These scheduling algorithms exploit the channel gain information and consider the scheduler as a controller that optimizes a given utility (such as sum throughput, power) subject to a given constraint (such as delay, fairness) under various assumptions on the arrival and channel gain processes.

It turns out that it is possible to formulate these problems within the stochastic control framework. As reviewed in this chapter, these problems have the structure of a CMDP and the optimal policy can be computed with various assumptions on the probability distributions of the arrival and channel gain processes. For example, in energy efficient scheduling, a number of papers have explored the structural results of the optimal policy. In principle, the optimal policy can be numerically computed using the LP approach or alternately using the iterative algorithms such as value iteration. However, these approaches suffer from the problem of curse of dimensionality, i.e., for large state spaces, these approaches are computationally infeasible. We will demonstrate how this limitation can be addressed using a novel approach proposed in Chapter 3.

Due to the above mentioned computational issues even for the single user scenario, the problem of energy efficient scheduling for the multiuser scenario has not received adequate attention. We have been successful in exploiting the framework of Chapter 3 for designing a novel scheduling algorithm for multiuser scenario, that we believe is Pareto optimal. Despite a plethora of literature, the problem of average sum throughput maximization on the downlink subject to individual user average delay constraints has also not been addressed in the literature. As pointed out in Section 2.3.3, while this problem can be formulated as a CMDP, large state space size is the primary reason for inadequate attention towards this problem. Moreover, in the absence of the system model, the problem becomes even more difficult to tackle.

Finally, while a large part of this thesis is devoted to centralized scheduling, we have also considered distributed scheduling. We consider the following interesting problem in the distributed scheduling scenario. Consider a random access system where pairs of users wish to communicate. All users are in the transmission range of each other. The receiver imposes an average delay constraint on the transmitter. Each time the transmitter transmits, some amount of energy is expended. The transmitter, therefore, has the objective of accessing the channel at as low a rate as possible in order to save energy. On the other hand, accessing the channel at too low a rate would lead to the delay constraint being violated. Thus, there exists a channel access rate-delay tradeoff. The specific problem of determining the minimum channel access rate or steady state transmission probability sufficient to satisfy the QoS (in this case, delay requirements) over a fading channel has not been studied so far. Towards the end of the thesis, we demonstrate how this problem can be addressed using stochastic approximation framework.

In the next chapter, we begin our investigations by studying the problem of energy efficient scheduling in a point-to-point system. The objective is to minimize power expenditure in the long run while providing QoS by maintaining the average queue length below a prescribed limit.



## Chapter 3

# Energy Efficient Scheduling for a Point-to-Point Link

In this chapter, we investigate energy efficient scheduling for a point-to-point link, i.e., over a single user fading channel. As pointed out in Chapter 2, power can be saved at an expense of increase in the delay suffered at the higher layers. However, QoS requirements at the higher layers (such as MAC or network layer) may mandate the delay to be maintained below a certain prescribed bound. The objective, therefore, is to determine a power/rate allocation policy that minimizes the average power expenditure subject to a constraint on the average delay. As pointed out in Chapter 2, this problem has been formulated as an optimization problem within the MDP framework and there are numerous papers that prove structural properties of the optimal policy.

The problem of computation of optimal scheduling policy has not been satisfactorily addressed in the sense that most of these schemes assume a knowledge of the probability distributions of the arrival and channel state processes for modeling the transition probability mechanism of the underlying Markov chain. This model knowledge is not easily available in practice, i.e., the exact model is not known. To address this issue, we propose a model unaware online algorithm that computes the optimal packet scheduling policy. The proposed algorithm is an online version of the well known Relative Value Iteration Algorithm (RVIA) for the average cost problem formulated within the CMDP framework. It is based on reformulating the value iteration equation by introducing a virtual state called *post-decision state*. The resultant value iteration equation has a nice structure that

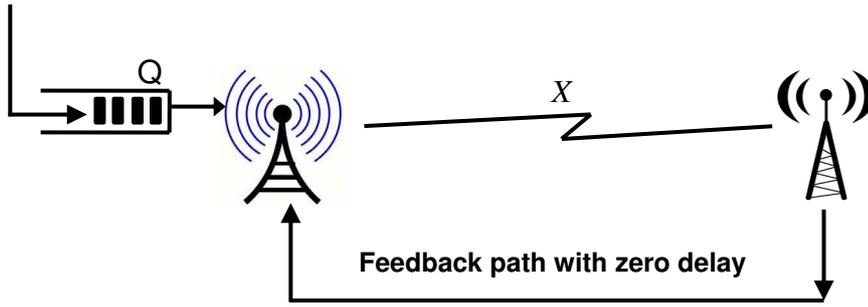


Figure 3.1: Point-to-point transmission model with finite buffer

lends itself naturally to its online implementation based on stochastic approximation. Note that like all other work [27, 36, 29, 30, 28, 62], we assume that the transmitter is aware of the channel state information at the beginning of each time slot. But unlike others, an explicit knowledge of the probability distribution of the channel state as well as the arrival process is not required for the proposed implementation. We also prove that the proposed algorithm converges to the optimal policy. Moreover, we present simulation results to demonstrate the efficacy of the algorithm in practical situations.

The rest of the chapter is organized as follows. We present the system model in Section 3.1. In Section 3.2, we cast the problem within the CMDP framework. The Lagrangian approach, a natural way of handling constraints, is presented in Section 3.3. We propose the optimal online algorithm in Section 3.4. A convergence analysis of the proposed algorithm is performed in Section 3.5, where we prove that it determines the optimal packet scheduling policy. Simulation results demonstrating the practical utility of the algorithm have been presented in Section 3.6. We conclude in Section 3.7.

### 3.1 System Model

The system model considered in this section is similar to the single user model reviewed in Section 2.3.4. However, for the sake of completeness, we revisit this model with assumptions that are specific to this formulation. The system model is illustrated in Figure 3.1. We assume that time is divided into slots of equal duration which is normalized to unity. Packets arrive at the transmitter buffer and get queued until they are transmitted. The packet arrival process  $\{A_n\}$  is assumed to be an i.i.d. sequence. For simplicity, we

make an additional assumption that every packet has a constant size, say  $\ell$  bits.

Recall from the previous chapter that  $Q_n$  denotes the queue length at the beginning of slot  $n$  and  $U_n$  denotes the number of packets to be transmitted in slot  $n$ . Since the transmitter can at most transmit all the packets that are present in the buffer at any time slot, we have  $U_n \leq Q_n$ . The queue evolution equation can be expressed as:

$$Q_{n+1} = Q_n + A_{n+1} - U_n. \quad (3.1)$$

In practice, the transmitter buffer may have a finite size, say  $B$  and therefore  $Q_n \in \mathbb{Q} \triangleq \{0, \dots, B\}$ . However, as compared to the packet arrival rate (and the average queue length/delay constraint), we assume  $B$  to be large enough so as to neglect the buffer overflow and hence packet drops. Alternately, we can formulate a problem where one attempts to minimize a ‘buffer cost’ that takes into account both the queueing delay and the buffer overflow. The formulation in this chapter can be easily extended to solve this alternate problem that minimizes the average power subject to a constraint on the buffer cost. However, for the sake of simplicity, we do not consider the effect of finite buffer size in the queue dynamics mentioned above.

We assume a correlated block fading channel model where the channel state process  $\{X_n\}$ ,  $X_n \in \mathbb{X}$  can be represented using a Markov chain. Full CSI is assumed to be known at the transmitter. We assume that the power  $P(x, u)$  required for transmitting  $u$  packets (each of  $\ell$  bits) at a channel state  $x$  is an increasing and strictly convex function of  $u$ . This assumption can be justified for most digital communication systems as discussed below:

- *Error free communication:* Following the discussion in [27], the power required for error-free communication in Shannon’s sense [37] at a rate  $U_n = u$  packets/sec when  $X_n = x$  can be expressed as:

$$P(x, u) = \frac{WN_0}{x} (2^{\frac{u\ell}{W}} - 1). \quad (3.2)$$

Note that for a given  $x$ , the transmission power  $P(x, u)$  is an increasing and strictly convex function of  $u$ .

- *Maintaining a target BER at the receiver:* In this case, the transmission power is determined based on the specific modulation scheme and target BER. It can be

shown that in this case too, the transmission power  $P(x, u)$  is an increasing and strictly convex function of the transmission rate [9].

We would like to point out that our formulation is not limited by the specific coding or modulation schemes implemented at the physical layer. We only exploit the convexity of transmission power in the rate.

The state of the above system  $S_n$  at time  $n$  can be described by the 2-tuple,  $S_n = (Q_n, X_n)$ , comprising of the queue length and the channel state. Note that the state space of the system  $\mathbb{S} = \mathbb{Q} \times \mathbb{X}$  is discrete and finite.  $\{U_n\}$  denotes the control process taking values from the finite action space  $\mathbb{U} = \{0, \dots, B\}$ . The control action determines the number of packets  $U_n$  to be transmitted in slot  $n$ . The control or scheduling policy is a sequence of functions  $\{\mu_n\}$  where  $\mu_n$  specifies  $U_n$  (more generally, the conditional law thereof) given the past history of the system state and past controls up to time  $n$ . The conditional law of the state  $S_{n+1}$  of the underlying dynamical system given the history depends only upon the state  $S_n$  and the control  $\mu_n$ , thus making it a Markov decision process. Let  $p : \mathbb{S} \times \mathbb{U} \times \mathbb{S} \rightarrow [0, 1]$  denote the state transition probability of this Markov Decision Process.

We note that we can also handle Markov arrivals by augmenting the state space further to include the arrival process. In this case, the state  $S_n$  in a slot  $n$  can be represented by the 3-tuple  $(Q_n, X_n, A_n)$ . The analysis will be similar. We avoid this generality in the interest of notational simplicity.

## 3.2 Formulation as a CMDP

As discussed in Chapter 2, the average power expenditure over a long period of time can be expressed as:

$$\bar{P} = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M P(X_n, U_n). \quad (3.3)$$

The average queue length over a long period of time can be expressed as:

$$\bar{Q} = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M Q_n. \quad (3.4)$$

The problem of average power minimization subject to average delay or equivalently queue length constraint (by Little's law [10]) can be expressed as the following optimization

problem:

$$\text{Minimize } \bar{P} \text{ subject to, } \bar{Q} \leq \bar{\delta}. \quad (3.5)$$

We now cast this problem within the CMDP framework. Let  $c_p(S_n, U_n)$  denote the cost in terms of power required in transmitting  $U_n$  packets when the state is  $S_n$ . Thus  $c_p(S_n, U_n) = P(X_n, U_n)$ . Since the packets get queued, they suffer a cost of buffering. Let  $c_q(S_n, U_n) \triangleq Q_n$ . The time averaged power and queue length can be expressed as:

$$\begin{aligned} \bar{P} &= \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbf{E} \sum_{n=1}^N c_p(S_n, U_n), \quad \text{and} \\ \bar{Q} &= \limsup_{N \rightarrow \infty} \frac{1}{N} \mathbf{E} \sum_{n=1}^N c_q(S_n, U_n), \end{aligned} \quad (3.6)$$

respectively. Our objective is to design a scheduler that minimizes  $\bar{P}$  subject to a constraint  $\bar{\delta}$  on  $\bar{Q}$ . It is a CMDP with average cost and finite state and action spaces. It is well known that a stationary randomized optimal policy exists [59]. Hence we concentrate only on stationary randomized policies characterized by  $\mu(\cdot|s) : s \in \mathbb{S} \rightarrow$  probability measures on  $\mathbb{U}$ . That is,  $\mu(\cdot|s)$  for each state  $s$  specifies the distribution with which the control in that state is applied. We assume irreducibility of the chain under such policies. Then  $\{S_n\}$  is an ergodic Markov chain and thus has a unique stationary distribution  $\rho^\mu$ . Let  $\mathbf{E}^\mu$  denote the expectation with respect to  $\rho^\mu$ . Under a randomized policy  $\mu$ , the costs in (3.6) can be expressed as:

$$\bar{P}^\mu \triangleq \mathbf{E}^\mu \left[ c_p(S_n, \mu(S_n)) \right] = \sum_{u,s} \rho^\mu(s) \mu(u|s) c_p(s, \mu(s)), \quad (3.7)$$

and,

$$\bar{Q}^\mu \triangleq \mathbf{E}^\mu \left[ c_q(S_n, \mu(S_n)) \right] = \sum_{u,s} \rho^\mu(s) \mu(u|s) c_q(s, \mu(s)), \quad (3.8)$$

respectively. Then the scheduler objective can be stated as:

$$\text{Minimize } \bar{P}^\mu \text{ subject to } \bar{Q}^\mu \leq \bar{\delta}. \quad (3.9)$$

### 3.3 Lagrangian Approach

The constrained problem in (3.9) can be converted into an unconstrained one using standard Lagrangian approach [59, 84]. In this section, we discuss the Lagrangian formulation and the corresponding dynamic programming equation.

Let  $\lambda \geq 0$  be a real number. Define  $c : \mathbb{R}^+ \times \mathbb{S} \times \mathbb{U} \rightarrow \mathbb{R}$  as follows,

$$c(\lambda, s, u) = c_p(s, u) + \lambda(c_q(s, u) - \bar{\delta}). \quad (3.10)$$

Note that the function  $c(\cdot, \cdot, u)$  is a strictly convex function of  $u$  (as the power required to transmit  $u$  packets is a strictly convex function of  $u$ ). The unconstrained problem is to determine an optimal stationary policy  $\mu^*(\cdot)$  that minimizes:

$$L(\mu, \lambda) = \mathbf{E}^\mu \left[ c(\lambda, S_n, \mu(S_n)) \right], \quad (3.11)$$

for a particular value of  $\lambda$  called the Lagrange Multiplier (LM).  $L(\cdot, \cdot)$  is called the Lagrangian. The following dynamic programming equation provides the necessary condition for optimality of the policy.

$$V(s) = \min_u \left[ c(\lambda, s, u) - \beta + \sum_{s'} p(s, u, s') V(s') \right], \quad s' \in \mathbb{S}, \quad (3.12)$$

where  $\beta \in \mathbb{R}$  is uniquely characterized as the corresponding optimal cost per stage. If we impose  $V(s^0) = 0$  for a fixed  $s^0 \in \mathbb{S}$ ,  $V$  is unique [22]. Furthermore, an optimal  $\mu^*$  must satisfy:

$$\text{support}(\mu^*(\cdot|s)) \subseteq \arg \min_{s'} \left[ c(\lambda, s, u) - \beta + \sum_{s'} p(s, u, s') V(s') \right] \quad \forall s \in \mathbb{S}. \quad (3.13)$$

By using standard arguments (see, e.g., [84]), it follows that the constrained problem has a stationary, though possibly randomized, optimal policy which is also optimal for the unconstrained problem considered in (3.11) for a particular choice of  $\lambda = \lambda^*$  (say). In fact, we know from [84] that the optimal stationary policy can be taken to be *deterministic* for all but at most one  $s$ , i.e., there exists a unique  $u^*(s)$  such that  $\mu^*(u^*(s)|s) = 1$  and  $u^*$  is the solution to the following equation:

$$u^*(s) = \arg \min_u \left[ c(\lambda^*, s, u) - \beta + \sum_{s'} p(s, u, s') V(s') \right] \quad \forall s \in \mathbb{S}. \quad (3.14)$$

Furthermore, for the single (if any) state  $s$  for which this fails,  $\mu(\cdot|s)$  is supported on exactly two points. The optimal average cost  $\beta$  gives the minimum power consumed  $\bar{P}^*$  subject to the specified delay (or equivalent queue length) constraint. Moreover, the following *saddle point condition* holds:

$$L(\mu^*, \lambda) \leq L(\mu^*, \lambda^*) \leq L(\mu, \lambda^*). \quad (3.15)$$

For a fixed  $\lambda$ , the Relative Value Iteration Algorithm (RVIA) is a known algorithm for solving the dynamic programming equation for the unconstrained problem in an iterative fashion. The average cost RVIA for determining the value function such that (3.12) is satisfied can be written as:

$$V_{n+1}(s) = \min_{u \in U(s)} [c(\lambda, s, u) + \sum_{s'} p(s, u, s') V_n(s')] - V_n(s^0), \quad (3.16)$$

where  $s, s', s^0 \in \mathbb{S}$  and  $s^0$  is any fixed state.  $V_n(\cdot)$  is an estimate of the value function after  $n$  iterations for a fixed LM  $\lambda$ .

RVIA (3.16) requires the knowledge of transition probabilities  $p(s, u, s')$  which in turn requires the knowledge of channel state and packet arrival distributions, which are unknown. In the rest of the chapter, we address this limitation by proposing a new approach based on post-decision state. We begin by first introducing the concept of a post-decision state.

## 3.4 Online Algorithm

In this section, we lay ground for developing an online algorithm by introducing the concept of a post decision state. We then reformulate the RVIA based on this state to obtain an online algorithm. The saddle point of the Lagrangian is reached by introducing coupled LM iterations on a slower timescale or update rate. Finally, we present algorithm details and discuss implementation issues.

### 3.4.1 Post-Decision State Framework

We define the post-decision state<sup>1</sup> to be the virtual state of the system immediately *after* taking a decision but *before* the action of the noise. Let  $s = (q, x)$  be the state  $S$  of the system in some time slot and the transmitter transmits  $U = u$  packets, then the post-decision state denoted by  $\tilde{S}$ ,  $\tilde{S} \in \mathbb{S}$  is  $(q-u, x)$ . If  $A = a$  arrivals occur in the post-decision state and the channel state changes to  $X = x'$ , then the system reaches the next actual system state, which can also be called the *pre-decision* state,  $(q', x') = (q - u + a, x')$ .

<sup>1</sup>Similar ideas have been around in the literature before, see, e.g., [85, 86].

Let  $\tilde{V} : S \rightarrow \mathbb{R}$  be the value function based on the post-decision state given by:

$$\tilde{V}(\tilde{s}) = \mathbf{E}^s[V(S)].$$

where  $\mathbf{E}^s$  is the expectation taken over all the pre-decision states that can be reached from the post-decision state  $\tilde{s}$ . Let  $\zeta$  be the law for the arrivals and  $\kappa(\cdot|\cdot)$  the transition probability function for the channel state process. Then  $\tilde{V}$  satisfies the post-decision dynamic programming equation, for  $\tilde{s} = (q, x)$ :

$$\tilde{V}(\tilde{s}) = \sum_{a, x'} \zeta(a) \kappa(x'|x) \left( \min_{u \leq q+a} [c(\lambda, (q+a, x'), u) + \tilde{V}((q+a-u, x'))] \right) - \beta. \quad (3.17)$$

From (3.16) and (3.17), we get the ‘one component at a time’ RVIA based on post-decision state as follows. Fix  $\tilde{s}^0$ . If the post-decision state at time  $n$  is  $\tilde{s} = (q, x)$ , then do:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \sum_{a, x'} \zeta(a) \kappa(x'|x) \left( \min_{u \leq q+a} [c(\lambda, (q+a, x'), u) + \tilde{V}_n((q+a-u, x'))] \right) - \tilde{V}_n(\tilde{s}^0); \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}'') \quad \forall \tilde{s}'' \neq \tilde{s}. \end{aligned} \quad (3.18)$$

The important thing to note here is that we update only the  $\tilde{s}$ -th component, not the rest. This is to lay ground for the online scheme we propose below, which is perforce ‘one at a time’, because one learns only about the current state being observed, and can, therefore, update only the corresponding component.

### 3.4.2 Reformulation of RVIA

In this section, we propose an online algorithm to evaluate  $\tilde{V}$ . We note that the RVIA (3.16) is not amenable to online implementation because of the occurrence of *min* operator outside the averaging operation w.r.t. an unknown conditional law. On the other hand, (3.18) has a useful structure in the sense that the *expectation* operation has been moved outside of the *min* operator. The *expectation* can thus be dropped by performing averaging in time in order to determine the optimal value function.

Let  $\{f_n\}$  be a positive update sequence that has the following properties:

$$\sum_n f_n = \infty; \quad \sum_n (f_n)^2 < \infty. \quad (3.19)$$

Then, following the theory of stochastic approximation [25], we can remove the expectation from (3.18), and perform averaging via the following update equation. Recall that  $S_n = (Q_n, X_n)$ ,  $n \geq 0$ , is our state process. If  $\tilde{s} = (q, x)$ , the post-decision state at time  $n$ , then do:

$$\begin{aligned}\tilde{V}_{n+1}(\tilde{s}) &= (1 - f_n)\tilde{V}_n(\tilde{s}) + f_n \left\{ \min_u [c(\lambda, (q + A_{n+1}, X_{n+1}), u) \right. \\ &\quad \left. + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}^0) \right\}, \\ &= \tilde{V}_n(\tilde{s}) + f_n \left\{ \min_u [c(\lambda, (q + A_{n+1}, X_{n+1}), u) + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] \right. \\ &\quad \left. - \tilde{V}_n(\tilde{s}) - \tilde{V}_n(\tilde{s}^0) \right\}, \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}) \quad \forall \tilde{s}'' \neq \tilde{s}.\end{aligned}\tag{3.20}$$

The algorithm (3.20) is a primal RVIA scheme that attempts to solve the dynamic programming equation for a fixed value of the LM  $\lambda$ .

### 3.4.3 LM Update

Let  $\{e_n\}$  be a positive update sequence that has the same properties as  $f_n$  expressed in (3.19). To reach the saddle point of the Lagrangian in (3.15), we introduce the following LM iterations:

$$\lambda_{n+1} = \Lambda[\lambda_n + e_n (Q_n - \bar{\delta})],\tag{3.21}$$

where we use the projection operator  $\Lambda$  to project the LM onto interval  $[0, \Gamma]$  for large enough  $\Gamma > 0$ , to ensure boundedness of the LM. We impose the following additional requirements on the update sequences  $f_n$  and  $e_n$ :

$$\sum_n (f_n^2 + e_n^2) < \infty, \quad \lim_{n \rightarrow \infty} \frac{e_n}{f_n} \rightarrow 0.\tag{3.22}$$

### 3.4.4 Complete Primal Dual Algorithm

The complete primal-dual RVI algorithm can be expressed as: for  $S_n = \tilde{s} = (q, x)$ :

$$\begin{aligned}\tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f_n \left\{ \min_u [c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) \right. \\ &\quad \left. + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}) - \tilde{V}_n(\tilde{s}^0) \right\},\end{aligned}\tag{3.23}$$

$$\tilde{V}_{n+1}(\tilde{s}'') = \tilde{V}_n(\tilde{s}) \quad \forall \tilde{s}'' \neq \tilde{s},\tag{3.24}$$

$$\lambda_{n+1} = \Lambda[\lambda_n + e_n (Q_n - \bar{\delta})].\tag{3.25}$$

That these iterates, indeed, converge to the optimal values is proved in Section 3.5. In (3.23, 3.24, 3.25), iterating simultaneously on the primal variable as well as the dual variable on different *timescales* ensures that the update rates of the primal and dual variables are different. The dual variable is updated on a slower timescale than the primal variable. This means that as viewed from the slower LM timescale, the primal variable appears to be equilibrated or converged to the optimal value corresponding to the current value of LM, while as viewed from the faster value function timescale, the LM values appear to be almost constant. This can be interpreted as iterating the LM after every  $k_n = \frac{f_n}{e_n} \gg 1$  iterations of the value function. Note that separation of timescales introduces a ‘leader-follower’ behavior among the two components (fast and slow) of the algorithm which prevents the possible interference of one in the convergence of the other if they were run concurrently on the same timescale. We prove in Section 3.5 that this scheme converges ‘almost surely’ (a.s.).

### 3.4.5 Implementation Details

Based on online primal-dual RVI computations (3.23, 3.24, 3.25), the transmitter implements the scheduling scheme as explained in Algorithm 1. We assume that the transmitter is aware of the value of channel state  $X_n$  in each time slot  $n$ . In practice, this may be achieved by receiver first estimating the channel state and then informing this to the transmitter through a feedback mechanism. In each time slot, the transmitter determines the number of packet arrivals, channel state and current queue length. The number of packets to be transmitted is then determined as explained in Algorithm 1. The value functions and the LM are appropriately updated. The algorithm thus continues in each slot  $n$ .

*Remark 3.1.* While the theoretical convergence is proved in Section 3.5, our simulation results demonstrate that convergence of the algorithm occurs in reasonable number of iterations (time slots) for practical purposes. In long file transfer applications, the duration of transfer is of the order of seconds, while the slot duration in wireless systems is of the order of milli-seconds. Hence, non-optimality may be present only for certain part of data transfer.

*Remark 3.2.* In practical scenarios, we may not wait for the actual convergence to take

- 1: Initialize the value function matrix  $\tilde{V}(\tilde{s}) = \tilde{V}(q, x) \leftarrow 0 \quad \forall q \in \mathbb{Q}, x \in \mathbb{X}$ , the LM  $\lambda_0 \leftarrow 0$ , the slot counter  $n \leftarrow 1$ , queue length  $q \leftarrow 0$ , channel states  $x \leftarrow 0, x' \leftarrow 0$ . Let reference state  $\tilde{s}^0 = (0, x^1)$ , where  $x^1 \in \mathbb{X}$ .
- 2: **while** TRUE **do**
- 3:   Determine number of arrivals  $A_{n+1} = a$  and channel state  $X_{n+1} = x'$  in the current slot.
- 4:   Transmit  $u$  packets, such that  $u$  minimizes the r.h.s in (3.23), thereby, power  $P(x', u)$  required to transmit  $u$  packets of size  $\ell$  bits is also determined.
- 5:   Update the component  $\tilde{s} = (q, x)$  of the value function matrix using (3.23). Rest of the components of the matrix remain unchanged.
- 6:   Update the LM  $\lambda$  using (3.25) ( $Q_n = q$ ).
- 7:    $n \leftarrow n + 1, q \leftarrow q + a - u, x \leftarrow x'$ .
- 8: **end while**

**Algorithm 1:** The Online Algorithm

place, but would like to be within a prescribed neighborhood of the optimal solution with high probability. Results of [87] give a bound on the number of iterations required for the iterate to be within a given distance from the convergence point thereafter with a prescribed high probability.

## 3.5 Convergence Analysis

In this section, we prove that the online algorithm indeed determines the optimal policy, i.e., for the algorithm (3.23, 3.24, 3.25) both the LM and value function iterates converge to their optimal values. The following theorem states this result.

**Theorem 3.1.** *For the algorithm (3.23, 3.24, 3.25), the iterates  $(\tilde{V}_n, \lambda_n) \rightarrow (\tilde{V}, \lambda^*)$ .*

We prove this in several steps. First, note that the purpose of subtracting  $\tilde{V}_n(\tilde{s}^0)$  from the r.h.s. in (3.23) is to keep the iterates stable. It turns out that  $\tilde{V}_n(\tilde{s}^0)$  converges to the optimal average cost per stage  $\beta$ . More generally, we can replace  $\tilde{V}_n(\tilde{s}^0)$  with a generic offset term  $\varkappa(\tilde{V}_n)$  if we make the following assumption on the function  $\varkappa : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}$  [88].

*Assumption 3.1.*  $\varkappa(\cdot)$  is Lipschitz and for  $\boldsymbol{\eta}$  equal to the constant vector of all 1's in  $\mathbb{R}^{|\mathcal{S}|}$ ,

$\varkappa(\boldsymbol{\eta}) = 1$  and  $\varkappa(\mathbf{x} + c\boldsymbol{\eta}) = \varkappa(\mathbf{x}) + c$  for  $c \in \mathbb{R}$ . We further assume that  $\varkappa(a\mathbf{x}) = a\varkappa(\mathbf{x})$  for  $a > 0$ .

With Assumption 3.1, a generalized form of the primal-dual algorithm (3.23, 3.24, 3.25) can be written as follows. If the post-decision state at time  $n$  is  $S_n = \tilde{s} = (q, x)$ , then do:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f_n \left\{ \min_u [c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) \right. \\ &\quad \left. + \tilde{V}_n((q + A_{n+1} - u, X_{n+1}))] - \tilde{V}_n(\tilde{s}) - \varkappa(\tilde{V}_n(\tilde{s})) \right\}, \end{aligned} \quad (3.26)$$

$$\tilde{V}_{n+1}(\tilde{s}'') = \tilde{V}_n(\tilde{s}) \quad \forall \tilde{s}'' \neq \tilde{s}, \quad (3.27)$$

$$\lambda_{n+1} = \Lambda[\lambda_n + e_n (Q_n - \bar{\delta})]. \quad (3.28)$$

We now proceed to show that the algorithm (3.26, 3.27, 3.28) tracks an associated ordinary differential equation (o.d.e.) as described later. Recall that  $\zeta$  is the law for the arrivals and  $\kappa(\cdot|\cdot)$  the transition probability function for the channel state process. Let  $T_\lambda : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$  be the map defined by:

$$\begin{aligned} (T_\lambda \tilde{V})(\tilde{s}) &= \sum_{a, x'} \kappa(x'|x)\zeta(a) \left\{ \min_{u \leq q+a} [c(\lambda, (q+a, x'), u) + \tilde{V}_n(q+a-u, x')] \right\}, \\ \tilde{s} &= (q, x) \in \mathcal{S}. \end{aligned} \quad (3.29)$$

Define  $T'_\lambda : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$  by  $T'_\lambda(\tilde{V}) = T_\lambda(\tilde{V}) - \varkappa(\tilde{V})\boldsymbol{\eta}$ . The RVIA can be written as:

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{s}) &= \tilde{V}_n(\tilde{s}) + f_n [T_{\lambda_n}(\tilde{V}_n(\tilde{s})) - \varkappa(\tilde{V}_n(\tilde{s})) - \tilde{V}_n(\tilde{s}) + M_{n+1}(\tilde{s})], \\ \tilde{V}_{n+1}(\tilde{s}'') &= \tilde{V}_n(\tilde{s}) \quad \forall \tilde{s}'' \neq \tilde{s}, \\ \lambda_{n+1} &= \Lambda[\lambda_n + e_n (Q_n - \bar{\delta})], \end{aligned} \quad (3.30)$$

where, for  $\tilde{s} = (q, x)$ ,

$$M_{n+1}(\tilde{s}) = \min_{u \leq q+A_{n+1}} [c(\lambda_n, (q + A_{n+1}, X_{n+1}), u) + \tilde{V}_n(q + A_{n+1} - u, X_{n+1})] - T_{\lambda_n}(\tilde{V}_n(\tilde{s})).$$

Let  $\mathbb{F}_n$  denote the  $\sigma$ -algebra,  $\sigma(S_m, A_m, U_m, m \leq n), n \geq 0$ . It can be verified that  $\mathbf{E}[M_{n+1}|\mathbb{F}_n] = 0$ . Consider

$$\dot{\tilde{V}}(t) = T'_\lambda(\tilde{V}(t)) - \tilde{V}(t). \quad (3.31)$$

It can be argued as in [88] that as  $t \rightarrow \infty$ ,  $\tilde{V}(t)$  converges to the unique fixed point of  $T'_\lambda(\cdot)$ , i.e.,  $\tilde{V}^\lambda$  such that

$$T'_\lambda(\tilde{V}) = \tilde{V}^\lambda. \quad (3.32)$$

$\tilde{V}^\lambda$  is the globally asymptotically stable equilibrium for the above o.d.e.

**Lemma 3.1.** *The post-decision value function iterates  $\{\tilde{V}_n\}$  remain bounded a.s.*

*Proof.* Consider  $T^0 : \mathbb{R}^{|S|} \rightarrow \mathbb{R}^{|S|}$  defined by:

$$(T^0 \tilde{V})(\tilde{s}) = \sum_{a, x'} \kappa(x'|x) \zeta(a) \min_{u \leq q+a} [\tilde{V}(q+a-u, x')] - \varkappa(\tilde{V}(\tilde{s})) \boldsymbol{\eta}. \quad (3.33)$$

Then  $T^0$  is also a contraction w.r.t  $\|\cdot\|_w$ ,  $\lim_{c \rightarrow \infty} \frac{T'_\lambda(c\tilde{V})}{c} = T^0(\tilde{V})$ , and the o.d.e.:

$$\dot{\tilde{V}}(t) = T^0(\tilde{V}) - \tilde{V}, \quad (3.34)$$

has origin as the globally asymptotically stable equilibrium (again, by arguments of [89]). This is the scaled limit of the o.d.e. (3.34) in the sense of [90]. Note that it is independent of  $\lambda$ . The claim follows from Theorem 2.1 of [90].  $\square$

**Lemma 3.2.** *In algorithm (3.26, 3.27, 3.28),  $\tilde{V}_n - \tilde{V}^{\lambda_n} \rightarrow 0$  a.s., where  $\tilde{V}^{\lambda_n}$  is the value function based on post-decision state for  $\lambda = \lambda_n$ .*

*Proof.* The algorithm (3.30) is the standard stochastic approximation algorithm with martingale difference noise  $M_{n+1}$ . From (3.22) and (3.30) it can be seen that the LM is varied on a much slower timescale than the post-decision relative value function estimate  $\tilde{V}_n$ . Therefore, the post decision value function iterations *see* the LM to be almost constant. To be precise, the  $\lambda$  iterations in (3.30) can be written as:  $\lambda_{n+1} = \lambda_n + \nu(n)$ , where  $\nu(n) = \mathcal{O}(e_n) = o(f_n)^2$ . Hence the limiting o.d.e.'s associated with (3.30) for analyzing the  $\tilde{V}_n$  iterates are,  $\dot{\tilde{V}}(t) = T'_\lambda(\tilde{V}(t)) - \tilde{V}(t)$ ;  $\dot{\lambda}(t) = 0$ . Since  $\dot{\lambda}(t) = 0$ , for analyzing the  $\tilde{V}_n$  iterates, it suffices to consider the o.d.e.:

$$\dot{\tilde{V}}(t) = T'_\lambda(\tilde{V}(t)) - \tilde{V}(t), \quad (3.35)$$

for any prescribed value of the LM  $\lambda$ . The rest follows by standard arguments as in [91].  $\square$

The  $\{\lambda_n\}$  iterations are bounded since they are constrained to remain in the interval  $[0, \Gamma]$ . We now prove that the coupled iterates converge to their optimal values  $(\tilde{V}^{\lambda^*}, \lambda^*)$ . Let  $\varkappa(\lambda) \triangleq \min_\mu L(\mu, \lambda)$ . We reproduce the following results from [92].

---

<sup>2</sup> $\mathcal{O}$  and  $o$  stand for the ‘‘Big-oh’’ and ‘‘Small-oh’’ notation respectively. Intuitively, this means that  $\nu(n)$  goes faster to 0 than  $f_n$ .

**Lemma 3.3.**  *$G$  is piecewise linear and concave. In particular, it is continuously differentiable except at finitely many points where both right and left derivatives exist.*

Define  $h(\lambda) \triangleq \sum_{s,u} \rho^\mu(s) c(\lambda, s, u)$ , where  $\mu$  is the optimal stationary policy when LM  $\lambda$  is used (note that this introduces an additional  $\lambda$ -dependence not explicitly shown). Consider

$$\dot{\lambda}(t) = h(\lambda(t)) - \bar{\delta}, \quad (3.36)$$

constrained to remain in  $(0, \infty)$ .

**Lemma 3.4.** *(3.36) is same as the gradient ascent:*

$$\dot{\lambda}(t) = \nabla G(\lambda(t)), \quad (3.37)$$

interpreted in the Caratheodory sense, i.e., as the integral equation,

$$\lambda(t) = \lambda(0) + \int_0^t \nabla G(\lambda(s)) ds, \quad t \geq 0. \quad (3.38)$$

*Proof.* This follows using the ‘generalized envelope theorem’ as in [92]. □

**Corollary 3.1.** *The iterates  $\lambda_n$  a.s. converge to the set of maxima of  $G$ .*

This follows by standard arguments as in [92].

**Corollary 3.2.**  *$\{\mu_n\}$  converge to the set of optimal policies corresponding to  $\lambda \in \arg \max(G)$ , a.s.*

Note that any  $\lambda \in \arg \max(G)$  is a valid Lagrange multiplier. This completes the proof of the theorem.

## 3.6 Simulation Results

In this section, we describe several simulation experiments to validate the analytical results. While the convergence results proved in the previous section are asymptotic, we demonstrate that for all practical purposes, the quantities like LM, power and delay converge within reasonable number of iterations. Consequently, for long file transfers lasting several seconds, we demonstrate that the algorithm essentially operates *optimally* for a

vast majority of the file transfer duration. Moreover, our simulation results also demonstrate the convexity of the power-delay curve as has been proved analytically in [27].

We implement Algorithm 1 in a simulation environment using MATLAB where we simulate the single user scenario depicted in Figure 2.7. We simulate a time slotted system with slot duration of 1 msec. Although the algorithm does not depend on any distribution for the channel gain  $H$ , for the purposes of modeling, we simulate an i.i.d. Rayleigh channel across slots. For a Rayleigh model, channel state  $X$  is an exponentially distributed random variable with probability density function given by  $f_X(x) = \frac{1}{\alpha}e^{-\frac{x}{\alpha}}$ , where  $\alpha$  is the mean of  $X$ . We discretize the channel into eight equal probability bins, with the boundaries specified by  $\{(-\infty, -8.47 \text{ dB}), [-8.47 \text{ dB}, -5.41 \text{ dB}), [-5.41 \text{ dB}, -3.28 \text{ dB}), [-3.28 \text{ dB}, -1.59 \text{ dB}), [-1.59 \text{ dB}, -0.08 \text{ dB}), [-0.08 \text{ dB}, 1.42 \text{ dB}), [1.42 \text{ dB}, 3.18 \text{ dB}), [3.18 \text{ dB}, \infty )\}$ . We choose the channel state space to be  $\mathbb{X} = \{x^1 = -13 \text{ dB}, x^2 = -8.47 \text{ dB}, x^3 = -5.41 \text{ dB}, x^4 = -3.28 \text{ dB}, x^5 = -1.59 \text{ dB}, x^6 = -0.08 \text{ dB}, x^7 = 1.42 \text{ dB}, x^8 = 3.18 \text{ dB}\}$ . This discretization of the state space of  $X$  has been justified in [36]. For the sake of simplicity, we assume that the power required for transmitting  $u$  packets each of length  $\ell$  bits when the channel state is  $x$  is given by (3.2), where we assume that  $W = 5$  MHz and the product  $WN_0$  to be normalized to 1. We simulate i.i.d. arrivals with a Poisson distribution with mean  $\varepsilon$ . This implies that the probability of generating  $j$  packets is given by  $p(j) = e^{-\varepsilon} \frac{\varepsilon^j}{j!}$ . We assume that packets are of equal size  $\ell = 5000$  bits. We also assume that the transmitter can transmit 1 to 8 packets in a slot. In practice, it can correspond to transmission using the following modulation schemes - Binary Phase Shift Keying (BPSK), Quadrature Phase shift Keying (QPSK), 8-Quadrature Amplitude Modulation (QAM), 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM respectively. This is because these modulation schemes have a spectral efficiency of 1 – 8 bits/sec/Hz, and with  $W = 5$  MHz, slot duration of 1 msec and packet size  $\ell = 5000$  bits, the transmitter can potentially transmit 1 – 8 packets/slot respectively. In all the simulation scenarios, we simulate the algorithm for 100,000 time slots. For the LM and value function update, we choose  $e_n = \frac{10}{n}$  and  $f_n = \frac{1}{n^{0.7}}$ . The parameters common to all the scenarios have been summarized in Table 3.1.

*Scenario 3.1. Convergence of LM for various delay constraints, arrival rates and channel states:* This scenario demonstrates the convergence of the LM  $\lambda$ . In each slot, arrivals

Simulation Parameter	Value
Packet size $\ell$	5000 bits
Slot duration	1 msec
Number of rates	8
Rates	1-8 packets/msec
Simulation time	100000 slots
$f_n$	$\frac{1}{n^{0.7}}$
$e_n$	$\frac{10}{n}$

Table 3.1: Summary of parameters common for all scenarios

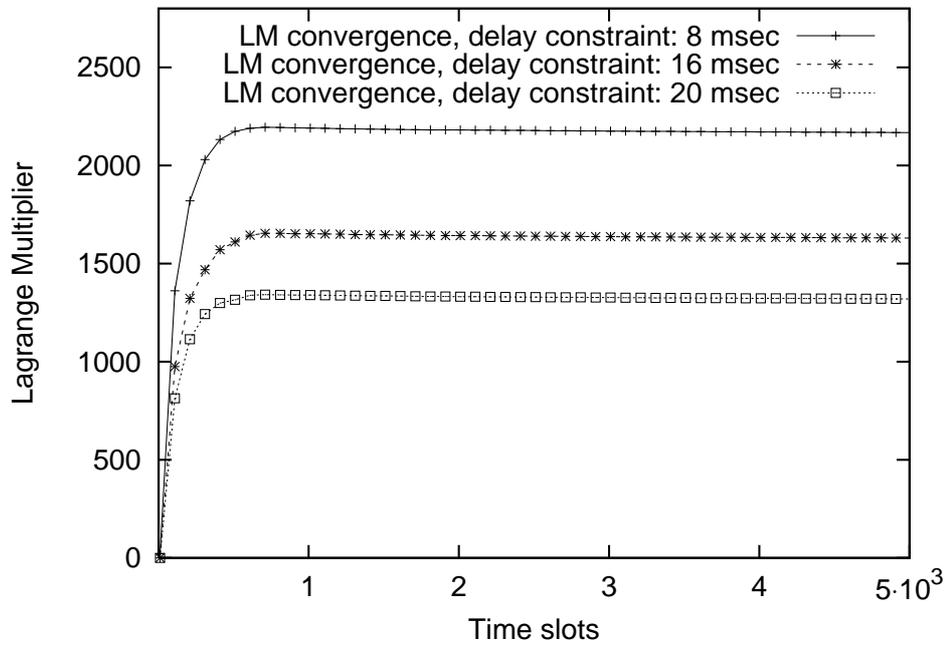


Figure 3.2: Convergence of Lagrange multiplier for various average delay constraints

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698 ( $-3.28$ dB)
Mean arrival rate $\varepsilon$	2 packets/msec, i.e., 10 Mbits/sec
Delay constraint	8, 16, 20 msec

Table 3.2: Summary of parameters for Scenario 3.1, (Figure 3.2)

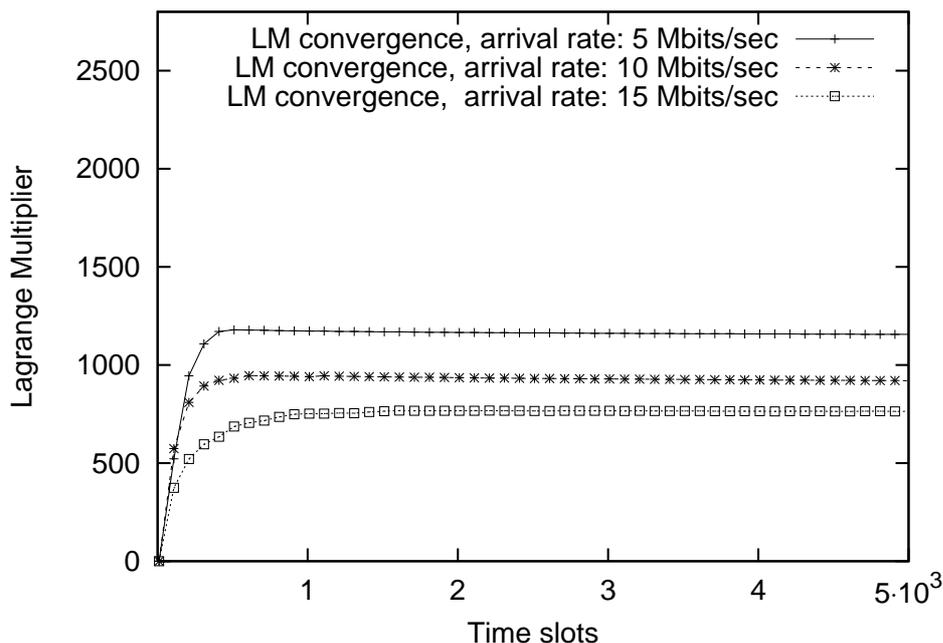


Figure 3.3: Convergence of Lagrange multiplier for various average arrival rates

are generated with Poisson distribution with mean  $\varepsilon = 2$  packets/msec, i.e., arrival rate is 10 Mbits/sec. We choose  $\alpha = 0.4698$  ( $-3.28$  dB). In each slot, we generate  $X$  using the exponential distribution with mean  $\alpha$ . We determine the channel state based on the partition that contains  $X = x$  as explained above. We then use Algorithm 1 to determine the number of packets  $u$  that must be transmitted, the transmission power  $P(x, u)$  and update the LM  $\lambda$  and the value function matrix  $\tilde{V}$ . We plot the variations in  $\lambda$  for delay constraints of 8 msec, 16 msec and 20 msec (summary of parameters provided in Table 3.2) in Figure 3.2. From Figure 3.2, it can be observed that the LM converges in about 2000 slots. Each time slot may, typically, correspond to 1 msec. Thus convergence can be achieved in about 2 seconds.

We repeat the scenario for average arrival rates of 5 Mbits/sec, 10 Mbits/sec and 15 Mbits/sec with the delay constraint at 24 msec and the average channel state  $\alpha$  to be

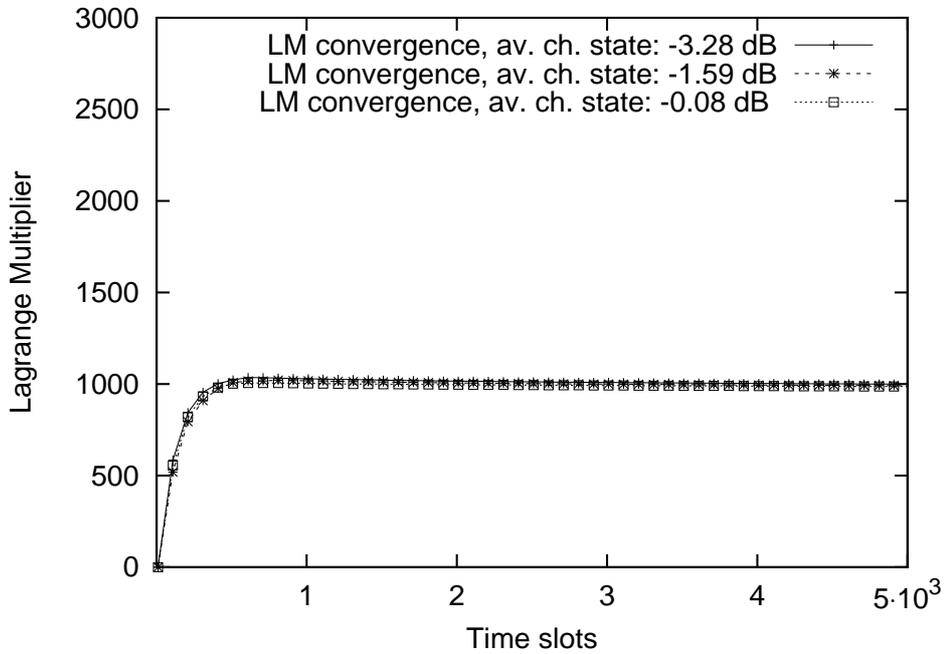


Figure 3.4: Convergence of Lagrange multiplier for various average channel states

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698 (-3.28 dB)
Mean arrival rate $\varepsilon$	1/2/3 packets/msec, i.e., 5/10/15 Mbits/sec
Delay constraint	24 msec

Table 3.3: Summary of parameters for Scenario 3.1, (Figure 3.3)

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698/0.6934/0.9817 ( $-3.28/ -1.59/ -0.08$ dB)
Mean arrival rate $\varepsilon$	2 packets/msec, i.e., 10 Mbits/sec
Delay constraint	24 msec

Table 3.4: Summary of parameters for Scenario 3.1, (Figure 3.4)

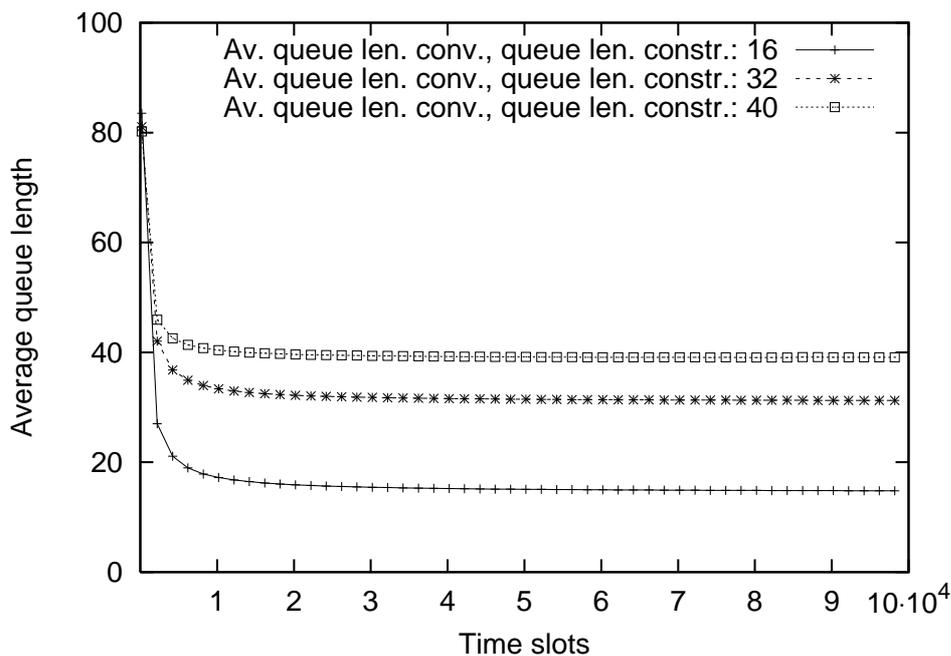


Figure 3.5: Convergence of average delay for various average delay constraints

0.4698 ( $-3.28$  dB). Table 3.3 summarizes the parameters. The convergence of LM  $\lambda$  for various arrival rates has been plotted in Figure 3.3. We further repeat the scenario with average channel state  $\alpha$  as 0.4698, 0.6934 and 0.9817, i.e.,  $-3.28$  dB,  $-1.59$  dB, and  $-0.08$  dB. We keep the delay constraint at 24 msec and average arrival rate  $\varepsilon$  at 2 packets/msec, i.e., 10 Mbits/sec. Table 3.4 summarizes the values of the parameters. The convergence of LM  $\lambda$  for various average channel states has been plotted in Figure 3.4. Figures 3.3 and 3.4 illustrate that the LM converges in approximately 2000 slots for all the arrival rate and channel state variations.

*Scenario 3.2. Convergence of delay and power for various delay constraints:* In this scenario, we demonstrate the convergence of the average delay and the average power for various delay constraints. We determine the running averages of the queue length  $Q_n^{av}$

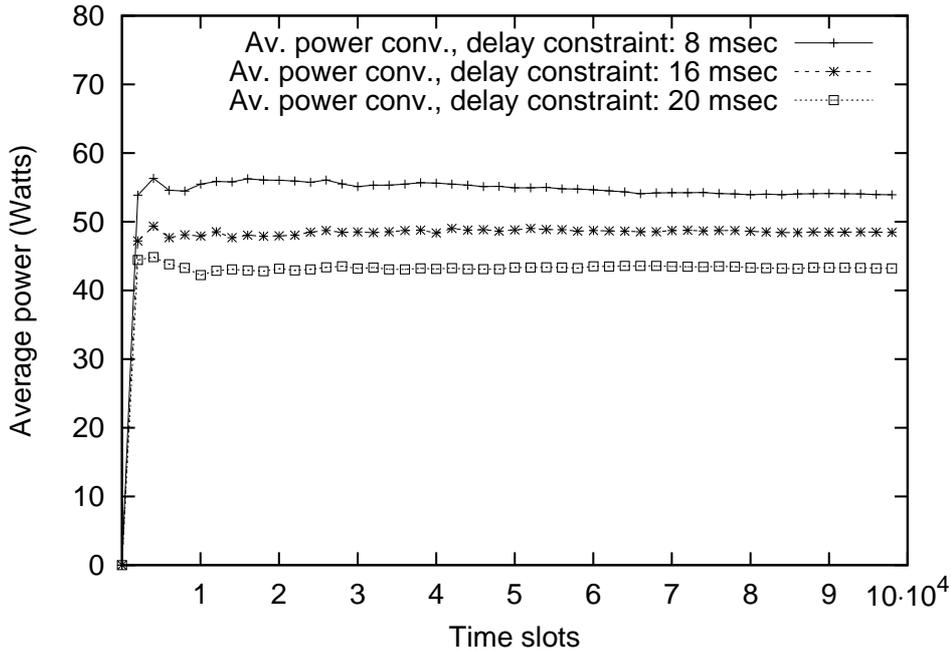


Figure 3.6: Convergence of average power for various average delay constraints

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698 ( $-3.28$ dB)
Mean arrival rate $\varepsilon$	2 packets/msec, i.e., 10 Mbits/sec
Delay constraint	8/16/20 msec

Table 3.5: Summary of parameters for Scenario 3.2, (Figures 3.5 and 3.6)

and the power expended  $P_n^{av}$  in slot  $n$  as follows

$$\begin{aligned}
 Q_n^{av} &= \frac{n-1}{n} Q_{n-1}^{av} + \frac{1}{n} Q_n, \\
 P_n^{av} &= \frac{n-1}{n} P_{n-1}^{av} + \frac{1}{n} P_n,
 \end{aligned} \tag{3.39}$$

with  $Q_0^{av} = 0$ ,  $P_0^{av} = 0$ . The variations in  $Q_n^{av}$  and  $P_n^{av}$  with simulation time for the delay constraints of 8 msec, 16 msec and 20 msec are plotted in Figures 3.5 and 3.6 respectively. We keep the arrival rate  $\varepsilon$  at 2 packets/msec, i.e., 10 Mbits/sec and the average channel state  $\alpha$  at 0.4698, i.e.,  $-3.28$  dB. We summarize the parameters in Table 3.5. Figures 3.5 and 3.6, illustrate that the average delay and average power converge reasonably fast.

*Scenario 3.3.* This scenario demonstrates that the algorithm satisfies various average delay constraints. We simulate the algorithm with delay constraints as 8, 12, 16, 20, 24, 28,

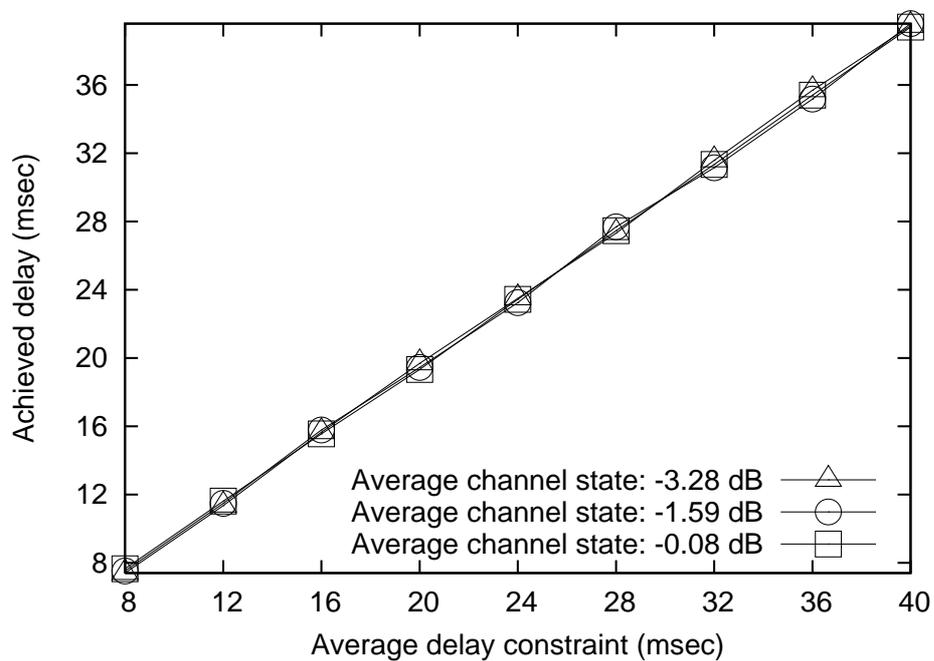


Figure 3.7: Achieved system delay for various average delay constraints

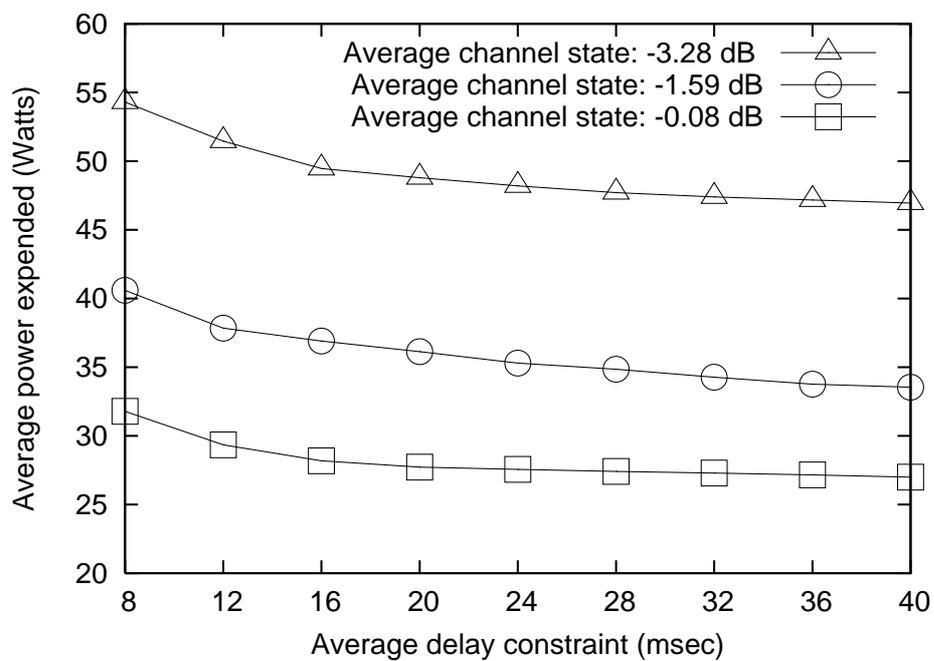


Figure 3.8: Power-delay curve for various average channel states

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698/0.6934/0.9817 ( $-3.28/ -1.59/ -0.08$ dB)
Mean arrival rate $\varepsilon$	2 packets/msec, i.e., 10 Mbits/sec
Delay constraint	8 – 40 msec in steps of 4 msec

Table 3.6: Summary of parameters for Scenario 3.3, (Figures 3.7 and 3.8)

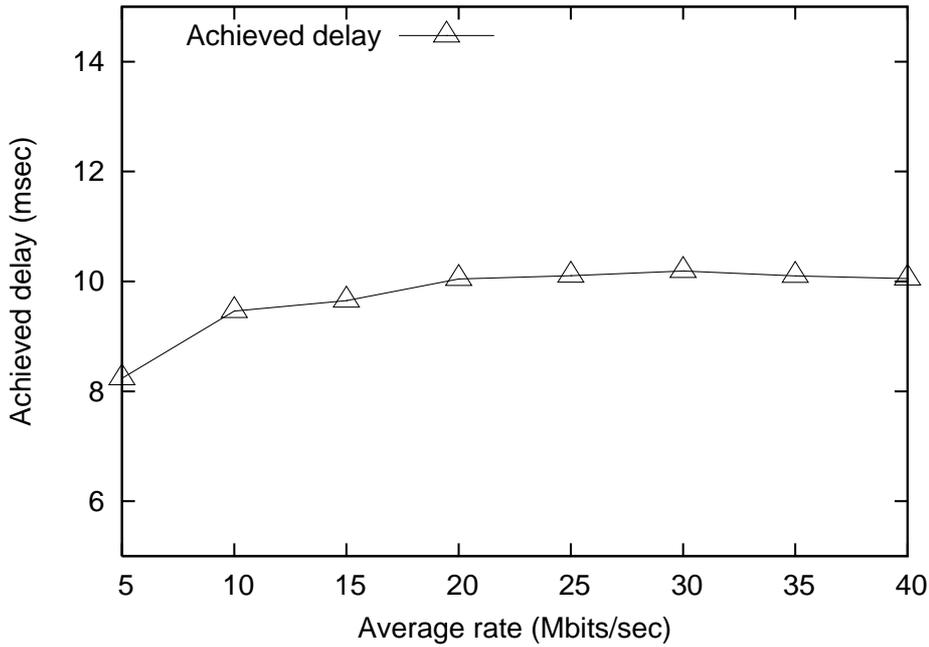


Figure 3.9: Achieved system delay for various average arrival rates

32, 36, 40 msec. We fix  $\varepsilon$  at 2 packets/msec, i.e., 10 Mbits/sec. We then repeat the scenario for values of  $\alpha = 0.4698, 0.6934, 0.9817$ , i.e.,  $-3.28$  dB,  $-1.59$  dB,  $-0.08$  dB. Table 3.6 summarizes the parameters. From Figure 3.7, it can be observed that, in all the cases, the average delay constraints are met. As the constraint on the delay increases, the average power required for transmission decreases as can be observed from Figure 3.8. The plot also demonstrates the convex characteristics of the power-delay curve that has been proved analytically in [27].

*Scenario 3.4.* This scenario demonstrates the range of arrival rates for which the algorithm satisfies a delay constraint of 10 msec. We perform the simulations for the arrival rates 5, 10, 15, 20, 25, 30, 35, 40 Mbits/sec by varying the mean  $\varepsilon$  of the Poisson distribution as 1, 2, 3, 4, 5, 6, 7, 8, packets/msec. We choose  $\alpha = 0.4698$  ( $-3.28$  dB). Table 3.7 summarizes

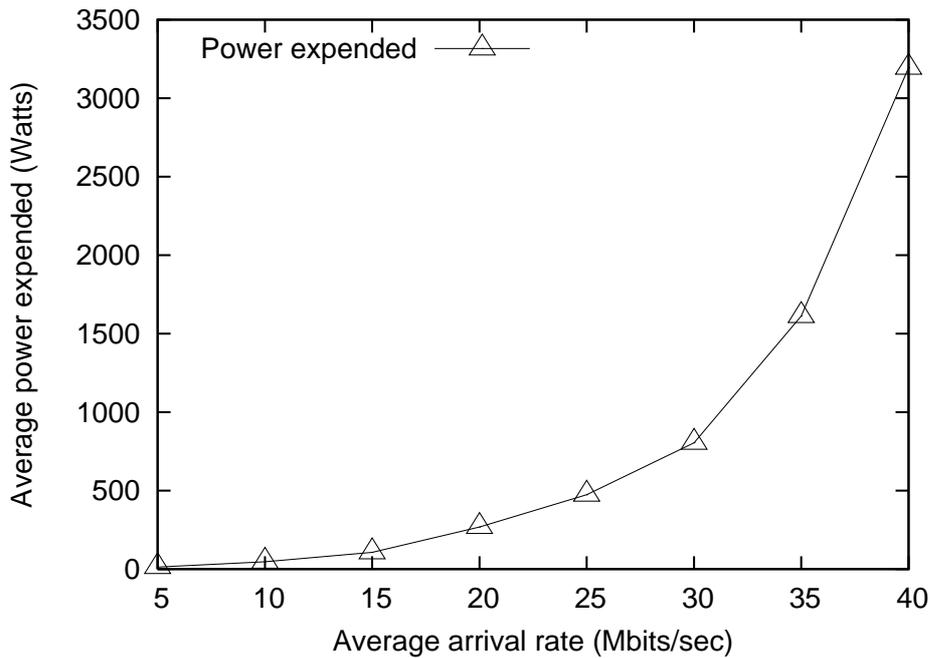


Figure 3.10: Power-arrival rate curve

Simulation Parameter	Value
Mean channel state $\alpha$	0.4698 (-3.28 dB)
Mean arrival rate $\varepsilon$	1-8 packets/msec, i.e., 5-40 Mbits/sec
Delay constraint	10 msec

Table 3.7: Summary of parameters for Scenario 3.4, (Figures 3.9 and 3.10)

the parameters. From Figure 3.9, it can be observed that the average delay constraint is met till the arrival rate becomes 8 packets/msec or 40 Mbits/sec. Beyond this, the arrival rate becomes more than the departure rate and thus the delay constraint cannot be satisfied as it violates the stability condition of the queue assumed in the formulation. As the arrival rate nears the capacity, the average power required for transmission is extremely high as can be observed from Figure 3.10.

## 3.7 Conclusions

In this chapter, we have considered the problem of minimizing the average power expenditure subject to maintaining average delay less than or equal to a prescribed limit. This problem has been studied previously, however, an important issue of computing the

optimal packet scheduling policy under an unknown system model was not addressed earlier. Towards this end, we have proposed an online learning algorithm for computing the optimal packet scheduling policy. Our approach is based on online implementation of the RVIA based on the novel concept of a post decision state. We have proved that the proposed algorithm indeed converges to the optimal policy. Our simulation results have illustrated the performance of the algorithm under various scenarios and have demonstrated that it is quite useful in practice.

The problem of packet scheduling for multiuser fading channel under various assumptions has been explored in [39, 45, 54, 93]. However, extension of the problem considered in this chapter (i.e., energy efficient delay constrained scheduling) for multiuser setting has not received much attention, notable exceptions being [64, 55]. We study the applicability of the framework developed in this chapter for solving the multiuser problem in the next chapter.

# Chapter 4

## Energy Efficient Scheduling for Multiuser Uplink

In this chapter, we consider an extension of the framework developed in Chapter 3, for multiuser scheduling. Specifically, we consider a single cell multiuser wireless uplink system with Time Division Multiple Access (TDMA). For such a system, we consider the problem of determining the user to be scheduled in each time slot (along with its transmission rate) so that the average transmission power expended by *each user* is minimized subject to a constraint on the average queuing delay experienced by *each user*. We believe that this problem formulation represents the user requirement more accurately than a related problem considered in literature [93] where the objective is to minimize the *sum power* on the uplink subject to individual user QoS (delay) constraint.

The primary difficulty in determining an optimal scheduling policy is the large state space as will be elaborated later in Section 4.2. Moreover, the state space size increases exponentially with the number of users. Furthermore, as argued in Chapter 3, exact model information is difficult to possess and the performance of the schemes developed under an assumed system model is limited by the accuracy of the model.

To address these issues, we develop a novel online algorithm. In the proposed approach, each user's queue evolution behaves as if it were controlled by a single user optimal policy. Depending on each user's channel state and queue size, the algorithm allocates a certain rate to each user in a slot using a modified version of the single user algorithm outlined in Chapter 3. The algorithm then schedules the user with the highest rate in a

slot. We argue that this algorithm satisfies the QoS constraints and exhibits a stabilizing behavior.

Multiuser scheduling algorithms proposed in the literature such as EXP [54], LQHPR [94], M-LWDF [16] (reviewed in Chapter 2) require the queue length information for determining the scheduling decision. In the downlink scenario, this information is readily available to the scheduler residing at the base station. However, in the uplink scenario, this information needs to be communicated by the users to the scheduler. Communicating the queue length information poses a significant overhead. In our approach, each user determines the rate at which it would transmit if it were scheduled in a slot. All the users inform these rates to the base station. In a practical system, we may have few possible rates, say eight. This means that we may need 3 bits of information to be conveyed. Thus the overhead imposed by the proposed scheme is not significant.

The rest of the chapter is organized as follows. In Section 4.1, we present the system model. We formulate the problem as an optimization problem in Section 4.2. In Section 4.3, we consider a hypothetical single user scenario. In Section 4.4, we propose an online algorithm that is based on a variation of the single user scenario described in Section 4.3. We also discuss the implementation issues. In Section 4.5, we prove certain properties of the algorithm. We present the simulation setup and results in Section 4.6. Finally, we conclude in Section 4.7.

## 4.1 System Model

The system model considered in this chapter has been introduced in Section 2.3.1 of Chapter 2. Here, we briefly recapitulate the important assumptions. We consider uplink transmissions (as in Figure 4.1) in a TDMA system with  $N$  users, i.e., time is divided into slots, each of unit duration and only one user is allowed to transmit in a slot. The base station is a centralized entity that schedules the users in each slot. We assume a fading wireless channel with i.i.d. block fading.  $\{X_n^i\}$  represents the i.i.d. channel state process for user  $i$ . We assume that the distribution of  $X_n^i$  is unknown.

We assume that the packets are of equal length:  $\ell$  bits. Each user has a buffer with finite size, say,  $B$  packets. Packets arrive into the user buffer and are queued until they

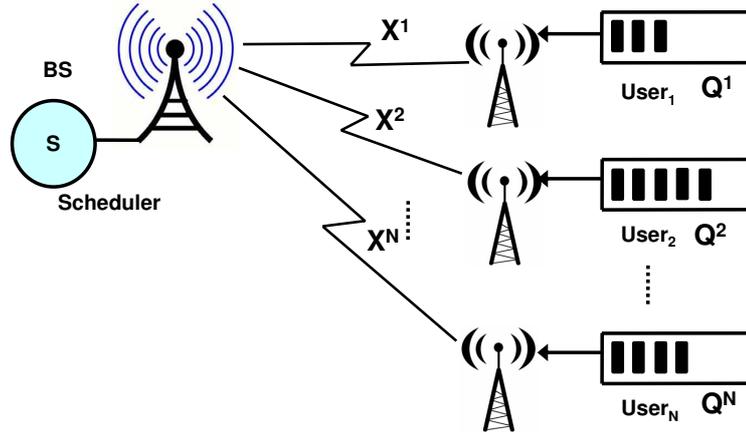


Figure 4.1: System Model

are transmitted.  $A_n^i \in \mathbb{A} \triangleq \{0, \dots, A\}$  denotes the number of packets arriving into the user  $i$  buffer in slot  $n$ , and the distribution of  $A_n^i$  is unknown. The packet arrival process for user  $i$ ,  $\{A_n^i\}$  is assumed to be i.i.d. across slots.  $Q_n^i \in \mathbb{Q} \triangleq \{0, \dots, B\}$  denotes the queue length or buffer occupancy of user  $i$  in slot  $n$ .  $U_n^i \in \mathbb{U} \triangleq \{0, \dots, B\}$  denotes the number of packets transmitted by user  $i$  in slot  $n$ . Let  $I_n^i$  be an indicator variable that is set to 1 if user  $i$  is scheduled in slot  $n$  and is set to 0 otherwise. Let  $\mathbf{I}_n$  be the vector  $[I_n^1, \dots, I_n^N]^T$ . Note that, since only one user can transmit in a slot, only one element of  $\mathbf{I}_n$  is equal to 1 and the rest are 0. Let  $\mathbb{I}$  be the set of all possible  $N$  dimensional vectors with one element equal to 1 and the rest being 0. Let  $R_n^i \in \mathbb{U}$  denote the number of packets that user  $i$  transmits in a slot, if it is scheduled. Then  $U_n^i$  can be represented as  $U_n^i = I_n^i R_n^i$ . Moreover, since a user can at most transmit all the packets in its buffer in a slot,  $R_n^i \leq Q_n^i$ . Since we assume that the slot length is normalized to unity,  $U_n^i$  also represents the rate at which user  $i$  transmits in slot  $n$ . Let  $\mathbf{U}_n$  be the vector  $[U_n^1, \dots, U_n^N]^T$ ,  $\mathbf{U}_n \in \mathbb{U}^N$ .

Following the discussion in Section 3.1, the queue evolution equation for user  $i$  can be expressed as:

$$Q_{n+1}^i = Q_n^i - U_n^i + A_{n+1}^i. \quad (4.1)$$

Recall that  $P(x, u)$  denotes the power required in transmitting  $u$  packets, each of length  $\ell$  bits when the channel state is  $x$ . From Chapter 3, the power required for error-free or reliable communication is given by (3.2). Let  $\hat{P}$  denote the peak power constraint. Let  $\hat{R}_n^i$

be the maximum number of packets which user  $i$  can transmit in slot  $n$  when the channel condition is  $X_n^i$  while satisfying the peak power constraint (i.e.,  $P(\hat{R}_n^i, X_n^i) \leq \hat{P}$ ). Then the set of feasible actions for user  $i$  in slot  $n$ ,  $\mathbb{F}_n^i \triangleq \{0, \dots, \min(\hat{R}_n^i, Q_n^i)\}$ .

We assume that the users specify their QoS in terms of the average delay requirement. These delay requirements of the users are known a priori to the scheduler. As discussed in Chapter 2, by Little's law [10], the average queue length  $\bar{Q}$  can be treated to be synonymous with the average delay  $\bar{D}$ .

## 4.2 Problem Formulation

In this section, we formulate the multiuser scheduling problem as a constrained optimization problem.

### 4.2.1 Formulation as a Constrained Optimization Problem

The objective is to schedule a user in a slot and also to determine its transmission rate (i.e., number of packets to be transmitted) such that the average power expenditure of each user is minimized and the delay constraint is satisfied. The average power consumed by user  $i$  over a long period of time can be expressed as:

$$\bar{P}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=1}^M P(X_n^i, I_n^i R_n^i). \quad (4.2)$$

The average queue length of user  $i$  over a long period of time can be expressed as:

$$\bar{Q}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=1}^M Q_n^i. \quad (4.3)$$

Each user  $i$  desires that its average queue length be maintained below a certain value, say,  $\bar{\delta}^i$ . Our objective is to design a scheduling algorithm that minimizes  $\bar{P}^i$  for each user  $i$  subject to a constraint on  $\bar{Q}^i$ . Thus the scheduler objectives can be stated as:

$$\text{Minimize } \bar{P}^i \text{ subject to } \bar{Q}^i \leq \bar{\delta}^i, \quad i = 1, \dots, N. \quad (4.4)$$

*Remark 4.1.* Note that there are actually  $N$  problems in (4.4). However, these problems are not independent. This is because in a TDMA system, only one user can be scheduled in a slot. Consequently, the scheduling decision in a slot impacts the buffer occupancy of all the users in future slots.

### 4.2.2 Notion of an Optimal Solution

The problem in (4.4) is a multi-objective optimization problem with  $N$  objectives and  $N$  constraints. There can be multiple average power vectors that can be considered as optimal. We seek *Pareto optimal* solutions [57]. Let  $[\bar{P}_\psi^1, \dots, \bar{P}_\psi^N]^T$  denote the average power expended under the scheduling policy  $\psi$ . The scheduling policy  $\psi$  is Pareto optimal if and only if there exists no policy  $\zeta$  with the corresponding power vector  $[\bar{P}_\zeta^1, \dots, \bar{P}_\zeta^N]^T$  having the following properties:

$$\forall i \in \{1, \dots, N\} P_\zeta^i \leq P_\psi^i \quad \wedge \quad \exists i \in \{1, \dots, N\} | P_\zeta^i < P_\psi^i. \quad (4.5)$$

A Pareto optimal solution is generally not unique and the set of Pareto optimal solutions is called the set of non-dominated solutions. The weighted sum approach is a common approach for solving a multi-objective optimization problem [57]. In this approach, one aggregates the  $N$  objective functions into a single objective function. The resultant problem has a single objective function with  $N$  constraints and can be expressed as:

$$\begin{aligned} \text{Minimize } \bar{P} &= \gamma^1 \bar{P}^1 + \dots + \gamma^N \bar{P}^N, \\ &\text{subject to,} \\ \bar{Q}^i &\leq \bar{\delta}^i, \quad i = 1, \dots, N. \end{aligned} \quad (4.6)$$

where  $\boldsymbol{\gamma} \triangleq [\gamma^1, \dots, \gamma^N]^T$  is the weight vector. It is generally assumed that  $\gamma^i \in [0, 1]$ ,  $\forall i$ ,  $\sum_{i=1}^N \gamma^i = 1$  implying that  $\bar{P}$  is a convex combination of the individual powers. In general, the non-dominated set (i.e., the set of all Pareto optimal policies) may be a non-convex set. By varying the weight vector in the weighted sum approach, we can determine the Pareto optimal policies within a convex subset of the non-dominated set. However, choosing the weight vector in order to obtain a particular solution is not straightforward.

### 4.2.3 Difficulties in Determining an Optimal Solution

The traditional approaches based on Linear Programming (LP) [59] for determining the optimal policy can not be used to solve (4.4) because of the following reasons:

1. *Large state space:* In our model, the system state space is large even for moderate number of users and the state space size increases exponentially with the number

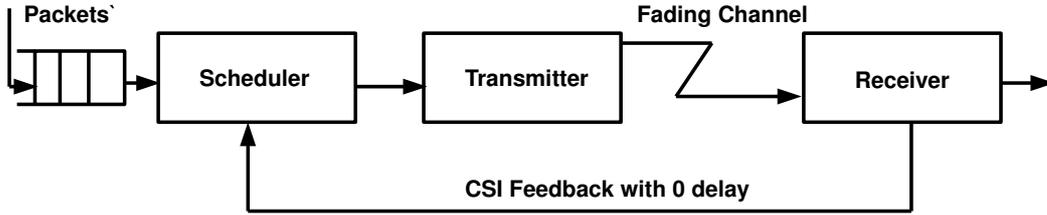


Figure 4.2: Hypothetical single user scenario

of users. We illustrate this with a simple example. Consider a system with 4 users. Assume that the each user reserves a buffer of size 50 packets (assuming equal sized packets). Assume that the channel condition of each user can be represented using 8 states, which is a practical assumption justified in [35]. For this scenario, the system state space contains  $50^4 \times 8^4 = 2.56 \times 10^{10}$  states. The computational complexity for determining the optimal policy (possibly based on the MDP approach) is proportional to the state space size [22, 24] and thus increases exponentially with the number of users.

2. *Unknown system model:* As argued in Chapter 3, we do not impose any system model related restrictions since exact model information is difficult to possess in practice.

The issue of unknown system model can be resolved by using reinforcement learning algorithms [24]. However, with such a large state space, the learning algorithms would take prohibitively large time to converge to the optimal scheduling policy. One, therefore, has to address the issue of the large state space first and then employ the reinforcement learning algorithms appropriately. This provides the motivation for designing the multiuser scheduling policy as an extension of the single user policy that searches over a relatively small state space. We consider a variant of the single user model proposed in Chapter 3. For this model, we first present an optimal on-line rate allocation algorithm. In the subsequent section, we then present the multiuser scheduling policy as an extension of this solution.

### 4.3 Transmission in the Presence of Transmitter Errors

In this section, we consider a hypothetical variant of the single user model described in Chapter 3, wherein we assume that the scheduling algorithm and actual transmitter are two different entities as shown in Figure 4.2. Once the online scheduling algorithm has determined the rate<sup>1</sup>  $R_n \in \mathbb{F}_n$ , we assume that, with a certain unknown random probability  $\theta_n \in [0, 1]$ , the transmitter executes this action, while with probability  $(1 - \theta_n)$ , it is unable to proceed with the transmission. We assume that the probability distribution of  $\theta_n$  is not known. If the transmitter does not transmit the packets as directed by the scheduler, the packets remain in the queue. Though this is an unrealistic scenario, we will demonstrate later that the algorithm for this model can be adapted for a more realistic multiuser setting.

Under the given model, the queue evolution equation can be expressed as:

$$Q_{n+1} = Q_n + A_{n+1} - I_n R_n, \quad (4.7)$$

where, as before,  $I_n$  is an indicator variable that is set to 1 if the transmitter actually transmits the packets and is set to 0 otherwise. The state of the system  $S_n$  at time  $n$  can be described by the two tuple,  $S_n = (Q_n, X_n)$  comprising of the queue length and the channel state. We now formulate the scheduling problem for this scenario. The long term power expenditure can be expressed as:

$$\bar{P}_e = \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=1}^M P(X_n, I_n R_n). \quad (4.8)$$

The average queue length over a long period of time can be expressed as:

$$\bar{Q}_e = \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=1}^M Q_n. \quad (4.9)$$

Hence, the single user scheduler objective can be stated as:

$$\text{Minimize } \bar{P}_e \text{ subject to } \bar{Q}_e \leq \bar{\delta}. \quad (4.10)$$

Note that the problem in (4.10) has the structure of a CMDP with average cost criterion. The objective is to determine an optimal policy  $\mu^*$  such that the power expended under this policy is minimum possible while satisfying the delay constraint.

<sup>1</sup>Since we consider a single user, we drop the superscript  $i$  in the notation.

### 4.3.1 Primal Dual Approach

The constrained problem in (4.10) can be converted into an unconstrained problem using the Lagrangian approach [59]. Let  $\lambda \geq 0$  be a real number termed as the Lagrange Multiplier (LM). Let  $\mathbb{B}$  be the set  $\{0, 1\}$ . Let  $c : \mathbb{R}^+ \times \mathbb{Q} \times \mathbb{X} \times \mathbb{B} \times \mathbb{U} \rightarrow \mathbb{R}$  be defined as the following:

$$c(\lambda, Q_n, X_n, I_n, R_n) \triangleq P(X_n, I_n R_n) + \lambda(Q_n - \bar{\delta}), \quad (4.11)$$

where  $R_n$  is determined using the rate allocation policy  $\mu : \mathbb{Q} \times \mathbb{X} \rightarrow \mathbb{U}$ . The unconstrained problem is to minimize:

$$L(\mu, \lambda) = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M c(\lambda, Q_n, X_n, I_n, \mu(Q_n, X_n)). \quad (4.12)$$

$L(\cdot, \cdot)$  is called the Lagrangian. Our objective is to determine the optimal rate allocation policy  $\mu^*$  and optimal LM  $\lambda^*$  such that the following saddle point optimality condition is satisfied:

$$L(\mu^*, \lambda) \leq L(\mu^*, \lambda^*) \leq L(\mu, \lambda^*). \quad (4.13)$$

### 4.3.2 Online Rate Allocation Algorithm

Let  $\{f_n\}$  and  $\{e_n\}$  be two sequences that have properties described in (3.19) and (3.22). The significance of these properties has already been explained in Chapter 3. Let the user state at the beginning of slot  $n$  be  $(Q_n, X_n) = (q, x)$ . Suppose that  $u$  packets are transmitted in slot  $n$ . The following primal-dual algorithm can be used to compute the rate  $R_{n+1} = r_{n+1}$  at which the transmitter should transmit in slot  $n + 1$ :

$$\begin{aligned} r_{n+1} = \arg \min_{v \in \mathbb{F}_{n+1}} & \left\{ (1 - f_n) \tilde{V}_n(\tilde{q}, \tilde{x}) + f_n \times \left\{ c(\lambda_n, \tilde{q} + A_{n+1}, X_{n+1}, 1, v) \right. \right. \\ & \left. \left. + \tilde{V}_n(\tilde{q} + A_{n+1} - v, X_{n+1}) - \tilde{V}_n(\tilde{q}^0, \tilde{x}^0) \right\} \right\}, \end{aligned} \quad (4.14)$$

$$\begin{aligned} \tilde{V}_{n+1}(\tilde{q}, \tilde{x}) = & (1 - f_n) \tilde{V}_n(\tilde{q}, \tilde{x}) + f_n \times \left\{ c(\lambda_n, \tilde{q} + A_{n+1}, X_{n+1}, I_{n+1}, r_{n+1}) \right. \\ & \left. + \tilde{V}_n(\tilde{q} + A_{n+1} - I_{n+1} r_{n+1}, X_{n+1}) - \tilde{V}_n(\tilde{q}^0, \tilde{x}^0) \right\}, \end{aligned} \quad (4.15)$$

$$\lambda_{n+1} = \Lambda[\lambda_n + e_n (Q_n - \bar{\delta})]. \quad (4.16)$$

These equations are explained below:

1. (4.14), (4.15) and (4.16) constitute the rate allocation algorithm. It consists of two phases: *rate determination phase* and *update phase*. (4.14) constitutes the rate determination phase of the algorithm, i.e., it is used to determine the rate at which a user transmits in a slot *if the transmission is successful*. (4.15) is a primal iteration to determine the optimal value function and thereby the optimal policy, while (4.16) is coupled dual iteration for determining the optimal LM. They constitute the update phase of the algorithm.
2. If in a state  $(Q_n, X_n) = (q, x)$ , the transmitter decides to transmit  $u \leq q$  packets, then then the system reaches the post-decision state  $\tilde{s} = (\tilde{q}, \tilde{x})$  where  $\tilde{q} \triangleq q - u$ ,  $\tilde{x} \triangleq x$ .
3. (4.15) determines the optimal value function  $\tilde{V}(\cdot)$  based on this post-decision state  $(\tilde{q}, \tilde{x})$ .
4. The rate determination phase (4.14) determines the rate assuming that the transmitter would be successful in transmitting in slot  $n + 1$  ( $I_{n+1}$  is assumed to be equal to 1). However in (4.15), updating the value function requires the knowledge of whether the transmission is successful or not. This because the immediate cost function  $c(\cdot, \cdot, \cdot, \cdot, \cdot)$  depends on  $I_{n+1}$ , i.e., whether the transmission is successful or not. Note that the value function is updated differently based on whether the transmission was successful or not based in (4.11). Moreover, successful transmission results in a corresponding queue transition.
5.  $(\tilde{q}^0, \tilde{x}^0)$  is any pre-designated state. On the RHS in (4.15), the value function corresponding to this state is subtracted in order to keep the iterates bounded.
6. The LM iteration in (4.16) ensures that the specified delay constraint is satisfied.

### 4.3.3 Proof of Convergence

**Theorem 4.1.** *For the rate determination algorithm (4.14), (4.15) and (4.16), the iterates  $(V_n, \lambda_n) \rightarrow (V, \lambda^*)$ .*

*Proof.* The proof of convergence is similar to that of Theorem 3.1 of Chapter 3. The term  $I_n$  in each slot serves as extra *noise* term. The algorithm, being a stochastic approximation

algorithm, averages out this extra noise term and determines the optimal policy and LM.  $\square$

## 4.4 An Online Primal Dual Algorithm for the Multiuser Problem

In this section, we propose an efficient approach to solve the problem in (4.4). We first determine the rate  $R_n^i \in \mathbb{F}_n^i$  for user  $i$  in slot  $n$ , *if it were to be scheduled*, based on its state  $S_n^i \triangleq [Q_n^i, X_n^i]$ . Note that  $S_n^i \in \mathbb{S} \triangleq \mathbb{Q} \times \mathbb{X}$ . The rate  $R_n^i$  is determined using a rate allocation policy  $\rho^i$ , i.e., a mapping from the history of states and rate allocations for user  $i$  to its current computed rate. Once the rate  $R_n^i$  for each user  $i$  is determined, the next task is to determine the user to be scheduled in that slot. The user selection policy  $\vartheta$  is a mapping,  $\vartheta : \mathbb{F}_n^1 \times \dots \times \mathbb{F}_n^N \rightarrow \mathbb{I}$ .

### 4.4.1 Rate Allocation Algorithm for a User

The rate allocation algorithm for each user behaves as if it were controlled by a single user policy as explained in Section 4.3. Each user  $i$  determines the rate  $R_{n+1}^i$  at which it would transmit in slot  $n+1$  if it were to be scheduled in slot  $n+1$  and informs this rate to the base station. The base station employs the user selection algorithm to schedule a user. The users who are not scheduled in a slot, update their value functions assuming unsuccessful transmission, while the user who is scheduled updates its value function assuming successful transmission.

### 4.4.2 User Selection Algorithm

The user selection algorithm is simple: select the user with the largest  $R_n^i$ , i.e., select the user with the best rate. The intuition behind this is the following. The rate allocation algorithm of user  $i$  would direct it to transmit at a high rate  $R_n^i$  under two circumstances: either the channel condition for that user is very good, in which case, transmission at high rate saves power, or the delay constraint of that user is not being satisfied. Thus selecting a user with a high rate results in either power savings or the user delay constraint being

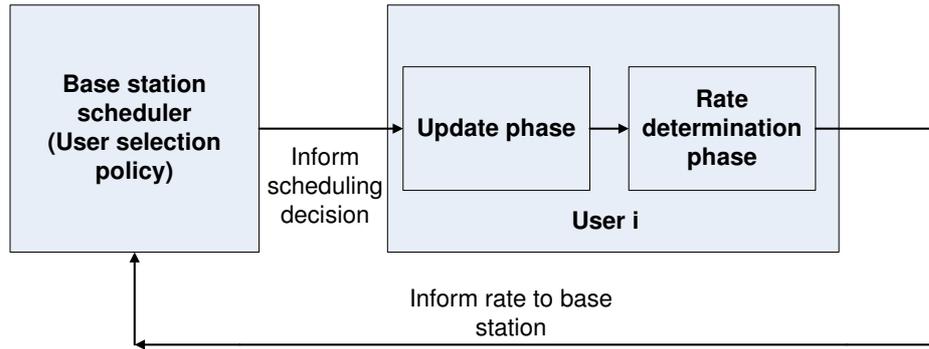


Figure 4.3: Scheduling phases

satisfied.

*Remark 4.2.* In the case of the single user model, the probability with which packets are finally transmitted by the transmitter is independent of the scheduler action, i.e., the transmission rate determined by the online algorithm. In the multiuser scenario, this independence does not hold. This makes the problem a multiagent learning problem [95], [96] where each agent (user) attempts to learn the optimal strategy and the actions taken by an agent (a user) influences the actions taken by the other agents (users). We argue in Section 4.5 that the algorithm has a stabilizing structure.

### 4.4.3 Implementation Details

The rate allocation algorithm is implemented on the user devices while the user selection algorithm is implemented at the base station as illustrated in Figure 4.3. From (4.14), note that the rate determination phase requires  $X_n^i$ , i.e., the knowledge of the channel state on the base station. The communication overhead incurred by the base station in informing the channel state perceived by it depends on the number of states used to represent the channel. We represent the channel using 8 states. Thus the base station needs 3 bits per slot in order to inform a user about the channel state perceived by it. Each user informs the base station about the rates at which it would transmit if it were to be scheduled. We allocate 3 bits for conveying this information, i.e., the system can employ 8 rates. The user selection algorithm then determines the user to be scheduled and all the users are informed about this decision. The rate allocation algorithm at each

```

1: Initialize the value function matrix  $\tilde{V}^i(q, x) \leftarrow 0 \quad \forall q \in \mathbb{Q}, x \in \mathbb{X}$ , LM  $\lambda_0^i \leftarrow 0$ , slot
   counter  $n \leftarrow 1$ , queue length  $Q_0^i \leftarrow 0$ , channel states  $X_0^i \leftarrow 0, X_0^{i'} \leftarrow 0$ .
2: Reference state  $\tilde{s}^{i,0} = (0, x^1)$ , where  $x^1 \in \mathbb{X}$ .
3: while TRUE do
4:   while Base station has not informed the channel state  $X_{n+1}^i = x^{i'}$  do
5:     wait.
6:   end while
7:   Determine the number of arrivals  $A_{n+1}^i = a^i$  in the current slot.
8:   Determine the queue length in the current slot  $Q_n^i = q^i$ .
9:   Use the rate determination phase of the rate allocation algorithm, i.e., (4.14) to
   determine the rate  $r^i$ , for transmission.
10:  Determine the power  $P(x^{i'}, r^i)$  required to transmit  $r^i$  packets.
11:  Inform the base station of the rate  $r^i$ .
12:  while Base station has not scheduled a user do
13:    wait.
14:  end while
15:  if user  $i$  is scheduled in slot  $n$  then
16:     $I_n^i \leftarrow 1$ .
17:  else
18:     $I_n^i \leftarrow 0$ .
19:  end if
20:  Update the component  $(q^i, x^i)$  of the value function matrix  $\tilde{V}^i$  using (4.15). Rest
   of the components of the matrix remain unchanged.
21:  Update the LM  $\lambda^i$  using (4.16) ( $Q_n^i = q^i$ ).
22:   $q^i \leftarrow q^i + a^i - u^i$ .
23:   $x^i \leftarrow x^{i'}$ .
24:   $n \leftarrow n + 1$ .
25: end while

```

**Algorithm 2:** The Rate Allocation Algorithm at the User  $i$  Device

user then enters the update phase where the value function and the LM for each user are appropriately updated using (4.15) and (4.16). The algorithm thus continues in each

slot. The rate allocation algorithm that is executed at each user device is illustrated in Algorithm 2, where steps 7–10 represent the rate determination phase, while steps 15–24 represent the update phase. The user selection algorithm executed at the base station is explained in Algorithm 3.

```

1: while TRUE do
2:   for  $i \in 1, \dots, N$  do
3:     Estimate the channel state  $X_{n+1}^i = x^{i'}$  in the current slot for user  $i$ .
4:     Inform  $x^{i'}$  to user  $i$ .
5:   end for
6:   while Rate of each user is not known do
7:     wait.
8:   end while
9:   Determine the user  $k$  who has the highest rate.
10:  Schedule user  $k$  in the current slot.
11: end while

```

**Algorithm 3:** The User Selection Algorithm at the Base Station

#### 4.4.4 Discussion

Here, we discuss certain aspects of the online algorithm:

1. *Computational complexity:* The computational complexity of the rate allocation algorithm executed at a user device is independent of the number of users in the system. This is because the rate allocation algorithm for any user  $i$  is dependent on the user  $i$  state  $S^i$  only and is independent of the states of the other users. The user selection algorithm has to determine the maximum of  $N$  numbers and hence is linear in  $N$ . Thus the computational complexity of the user selection algorithm grows only linearly with the number of users.
2. *An auctioning interpretation:* The solution can be interpreted as an auction, where the user selection algorithm auctions each slot. The users bid in the form of their transmission rates to the user selection algorithm, which allocates the slot to the user bidding the highest rate. The rate bid by a user is dependent on its channel

state and queue length constraint violation (i.e., the difference between the current queue length and the queue length constraint). If the channel state is quite good and queue constraint violation is large, the user bids a high rate. This is because transmitting at a high rate when the channel state is good saves power, while doing it when the queue length constraint violation is large aids in satisfying the delays. Note that the users do not bid unnecessarily high rates because that might result in higher power consumption. For a user, not winning an auction in a certain slot, implies that other users either have better channel states or higher queue length constraint violation or both. If a user does not win the auction for a certain number of slots successively, its queue length grows thus forcing it to bid a higher rate. Motivated by this interpretation, we refer to the scheduling scheme proposed in this chapter as *Auction Algorithm* (AA).

## 4.5 Analysis of AA

In this section, we investigate certain properties of AA. Specifically, we analyze the convergence behavior of the algorithm. Though the proof is somewhat less rigorous, it offers an intuitive justification for the stabilization of value function and LM. Indeed, we have demonstrated later through simulations that the algorithm does exhibit this behavior.

Recall our algorithm:

$$\begin{aligned} \tilde{V}_{n+1}^i(\tilde{q}^i, \tilde{x}^i) &= (1 - f_n) \tilde{V}_n^i(\tilde{q}^i, \tilde{x}^i) + f_n \left\{ (I_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 1, r_{n+1}^i)] \right. \\ &\quad + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i - r_{n+1}^i, X_{n+1}^i)] + (1 - I_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 0, r_{n+1}^i)] \\ &\quad \left. + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i, X_{n+1}^i)] - \tilde{V}_n^i(\tilde{q}^{i,0}, \tilde{x}^{i,0}) \right\}, \\ \lambda_{n+1}^i &= \Lambda[\lambda_n^i + e_n (Q_n^i - \bar{\delta}^i)], \end{aligned} \quad (4.17)$$

Note that the purpose of subtracting  $\tilde{V}_n^i(\tilde{q}^{i,0}, \tilde{x}^{i,0})$  from the r.h.s. in first equation in (4.17) is to keep the iterates stable. More generally, we can replace  $\tilde{V}_n^i(\tilde{q}^{i,0}, \tilde{x}^{i,0})$  with a generic offset term  $\varkappa(\tilde{V}_n^i)$  if we make the following assumption on the function  $\varkappa : \mathbb{R}^{|S|} \rightarrow \mathbb{R}$  [88].

*Assumption 4.1.*  $\varkappa(\cdot)$  is Lipschitz and for  $\boldsymbol{\eta}$  equal to the constant vector of all 1's in  $\mathbb{R}^{|S|}$ ,  $\varkappa(\boldsymbol{\eta}) = 1$  and  $\varkappa(\mathbf{x} + c\boldsymbol{\eta}) = \varkappa(\mathbf{x}) + c$  for  $c \in \mathbb{R}$ . We further assume that  $\varkappa(a\mathbf{x}) = a\varkappa(\mathbf{x})$  for  $a > 0$ .

With Assumption 4.1, a generalized form of the primal-dual algorithm (4.17) can be written as follows. At time  $n$ , if the state after taking the scheduling decision for user  $i$ ,  $\tilde{s}^i = (\tilde{q}^i, \tilde{x}^i)$ , then do:

$$\begin{aligned}\tilde{V}_{n+1}^i(\tilde{s}^i) &= (1 - f_n)\tilde{V}_n^i(\tilde{q}^i, \tilde{x}^i) + f_n \times \left\{ (I_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 1, r_{n+1}^i)] \right. \\ &\quad + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i - r_{n+1}^i, X_{n+1}^i)] + (1 - I_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 0, r_{n+1}^i)] \\ &\quad \left. + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i, X_{n+1}^i)] - \varkappa(\tilde{V}_n^i(\tilde{s}^{i,0})) \right\}, \\ \lambda_{n+1}^i &= \Lambda[\lambda_n^i + e_n (Q_n^i - \bar{\delta}^i)],\end{aligned}\tag{4.18}$$

where  $\tilde{s}^{i,0} = (\tilde{q}^{i,0}, \tilde{x}^{i,0})$ .

Let  $\theta_n^i$  be the probability with which user  $i$  transmits in slot  $n$  and  $\theta^i$  be the average probability of transmission for user  $i$ . We can express (4.18) equivalently in terms of  $\theta_n^i$  as:

$$\begin{aligned}\tilde{V}_{n+1}^i(\tilde{s}^i) &= (1 - f_n)\tilde{V}_n^i(\tilde{q}^i, \tilde{x}^i) + f_n \times \left\{ (\theta_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 1, r_{n+1}^i)] \right. \\ &\quad + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i - r_{n+1}^i, X_{n+1}^i)] + (1 - \theta_n^i)[c(\lambda_n^i, \tilde{q}^i + A_{n+1}^i, X_{n+1}^i, 0, r_{n+1}^i)] \\ &\quad \left. + \tilde{V}_n^i(\tilde{q}^i + A_{n+1}^i, X_{n+1}^i)] - \varkappa(\tilde{V}_n^i(\tilde{s}^{i,0})) \right\}, \\ \lambda_{n+1}^i &= \Lambda[\lambda_n^i + e_n (Q_n^i - \bar{\delta}^i)],\end{aligned}\tag{4.19}$$

Some analytical properties of the algorithm in (4.19) have been investigated in Appendix D. These relate to the stabilizing behavior of LM and value function iterates. Note that an increase in  $\lambda^i$  for user  $i$  results in an increase in transmission rate for user  $i$  (Lemma D.1). This results in an increase in  $\theta^i$  resulting in a decrease in  $\theta^j$  for user  $j$ . The reduced  $\theta^j$  results in an increase in  $\lambda^j$  by Lemma D.2. Combining these results, we have  $\frac{\partial \lambda^i}{\partial \lambda^j} > 0$  by Lemma D.4. Since the LMs are constrained to remain in the interval  $[0, \Gamma]$ , they are bounded. By employing an argument similar to the cooperative o.d.e. concept, we justify stabilization of the LMs. Following a similar argument, the value function stabilization can also be analyzed. Note that the convergence of LMs to an equilibrium indicates that the delay constraints of all the users are satisfied. A more rigorous proof of the convergence remains an interesting issue; although, as we have justified and later, illustrated through simulations, we believe that the algorithm does converge to a Pareto equilibrium.

*Remark 4.3.* The users always attempt to keep their average queue lengths close to their respective queue length constraint, i.e., constraint is satisfied with equality. This is because the users' objective is to minimize the power expenditure. By informing a higher rate than what is required, a user might achieve an average queue length that is much lower than the queue length constraint. However, as proved in [27] for single user policy, since the power is an increasing convex function of the delay, the average queue length should be as large as possible (in this case, equal to the queue length constraint) in order to save power. Thus, the users transmit at a rate such that the average throughput achieved by them is just sufficient to meet the delay constraint with equality.

*Remark 4.4.* Note that, once the users begin transmitting at a stable transmission power, in order to reduce the power consumption of say, user  $i$ , the base station has to reduce the average rate with which user  $i$  transmits. Since the delay constraint of user  $i$  must be satisfied, this can be done by increasing the fraction of slots allocated to that user. This results in decreasing the fraction of slots allocated to some other user, say, user  $j$ . Now, if user  $j$  has to satisfy its delay constraint, it has to increase the rate at which it transmits, thus increasing its power expenditure. Thus, once the system has stabilized, reduction in the power expenditure of one user is possible only at an expense of increase in the power expenditure of some other user.

*Remark 4.5.* Consider a modification of the user selection algorithm proposed in Section 4.4.2 where with a very small probability  $\chi$ , a user not having the maximum rate is scheduled in a slot. This modification ensures that the algorithm *explores* sufficiently, i.e., all the states are visited sufficiently often and consequently the algorithm estimates the value function for all the states with sufficient accuracy. More precisely, let  $\zeta(s^i, n)$  be the number of times that user  $i$  state  $s^i$  is visited upto time  $n$ . The modified algorithm ensures the following property:

$$\lim_{n \rightarrow \infty} \frac{\zeta(s^i, n)}{n} \neq 0, \quad \forall s^i, i. \quad (4.20)$$

It is difficult to prove that the auction algorithm suggested in this chapter ensures this property. However, the simulation results are quite promising and hence we do not implement this modification to the user selection algorithm.

## 4.6 Experimental Evaluation

We demonstrate the performance of our algorithm under the IEEE 802.16 [5] framework through simulations in a discrete event simulator. Specifically, we intend to demonstrate the following:

1. The algorithm satisfies the delay constraints of all the users.
2. The algorithm is efficient in terms of the power consumed for each of the users. Moreover, power consumed is commensurate with the delay requirement, average arrival rate and average channel state of a user.

We begin by providing some details regarding the IEEE 802.16 system.

### 4.6.1 The IEEE 802.16 System

The IEEE 802.16 standard specifies two modes for sharing the wireless medium: Point-to-Multipoint (PMP) and mesh. In this chapter, we concentrate on the PMP mode where a centralized base station (BS) serves multiple Subscriber Stations (SSs). We consider the uplink (UL) transmissions. IEEE 802.16 MAC specifies four different scheduling services in order to meet the QoS requirement of various applications. These are: Unsolicited Grant Service (UGS) (for real-time applications with strict delay requirement), real-time polling service (rtPS) (for real-time applications with less stringent delay requirement), non-real time polling service (nrtPS) and best effort (BE) (for applications that do not have any delay requirement). However, unlike BE connection, nrtPS connection is reserved a minimum amount of bandwidth. We consider the *residential* scenario as in [97]. It consists of a BS providing Internet access to the subscribers. Although the standard does not specify any QoS class for providing average delays, the nrtPS can be extended to cater to the average delay requirement of the users. The unicast polling service of nrtPS can be extended to inform a user about the channel state perceived by the base station as well as to determine the rate at which a user would transmit if it were to be scheduled. The scheduling algorithm can thus be implemented as a part of nrtPS.

The system can be operated in either TDD or FDD mode. We assume the FDD mode of operation where all SSs have full-duplex capability. We consider a single carrier

system (WirelessMAN-SC)<sup>2</sup> with a frame duration of 1 msec and bandwidth  $W$  of 10 MHz. To keep the scenario simple, we assume that the users transmit at a rate such that data is delivered reliably to the base station. Thus we do not consider retransmissions and Automatic Repeat Request (ARQ). The SSs employ the following modulations: 64-QAM, 16-QAM, QPSK, QPSK with 1/2 rate convolutional code which provide us with 4 rates of transmission.

## 4.6.2 Simulation Results

Internet traffic is modeled as a web traffic source [97, 98]. Variable sized packets are generated at the application layer. Packet sizes are drawn from a truncated Pareto distribution. This distribution is characterized by three parameters: shape factor  $\xi$ , mode  $v$  and cutoff threshold  $c$ . The probability that a packet has a size  $\ell$  can be expressed as:

$$\begin{aligned} f_{TP}(\ell) &= \frac{\xi \cdot v^\xi}{\ell^{\xi+1}}, & v \leq \ell < c \\ f_{TP}(\ell) &= \eta, & \ell \geq c, \end{aligned} \quad (4.21)$$

where  $\eta$  can be calculated to be equal to:

$$\eta = \left(\frac{\xi}{c}\right)^\xi, \quad \xi > 1. \quad (4.22)$$

We choose shape factor  $\xi = 1.2$ , mode  $v = 2000$  bits, cutoff threshold  $c = 10000$  bits, which provides us with an average packet size of 3860 bits. In each time frame, we generate the arrivals for all the users using Poisson distribution. Arrivals are generated in an i.i.d. manner across frames. We divide the packets into fragments at the MAC layer with each fragment being of size  $\ell = 2000$  bits. Fragments of size less than 2000 bits are padded with extra bits. Since all fragments are of equal size, we determine the transmission rate for users in terms of number of fragments. We simulate a Rayleigh fading channel<sup>3</sup> for each user. For a Rayleigh model, channel state  $X^i$  is an exponentially distributed random variable with mean  $\alpha^i$  and probability density function expressed in (2.7). We assume that the power required for transmitting  $u$  fragments of size  $\ell$  bits when the channel state is  $x$  is  $P(x, u)$  given by (3.2). We assume that the product  $WN_0$  is normalized to 1.

<sup>2</sup>This is assumed for simplicity. The algorithm can be easily extended for the Orthogonal Frequency Division Multiplexing (OFDM) system.

<sup>3</sup>Note that our algorithm does not use this knowledge of the channel and arrival process model.

Simulation Parameter	Value
Slot duration	1 msec
Number of rates	4
Bandwidth	10 MHz
Simulation time	100000 slots
$f_n$	$\frac{C}{n^{0.9}}, C > 0$
$e_n$	$\frac{D}{n^{0.7}}, D > 0$
$\xi$	1.2
$v$	2000
$c$	10000 bits
$N$	20
$\Gamma$	10000

Table 4.1: Summary of parameters common for all scenarios

We measure the sum of queuing and transmission delays of the packets and ignore the propagation delays. In all the scenarios described below, a single simulation run consists of running the algorithm for 100000 frames and the results are obtained after averaging over 20 simulation runs. We discretize the channel into eight equal probability bins, with the discretization procedure explained in Chapter 3. For each bin, we associate a channel state and the state space  $\mathbb{X} = \{-13 \text{ dB}, -8.47 \text{ dB}, -5.41 \text{ dB}, -3.28 \text{ dB}, -1.59 \text{ dB}, -0.08 \text{ dB}, 1.42 \text{ dB}, 3.18 \text{ dB}\}$ . We assume  $N = 20$ , i.e., a system with 20 users and thereby 20 UL connections. We assume that the number of users does not change during the course of simulations. Users are divided into two groups (Group 1 and Group 2) of 10 users each. A summary of the parameters common to all the simulation scenarios is provided in Table 4.1.

*Scenario 4.1.* In this scenario, we demonstrate that the AA satisfies the various user specified delay constraints. We consider two cases: symmetric and asymmetric. In each frame, arrivals are generated with Poisson distribution with mean 0.1 packets/msec. This results in an arrival rate of 0.386 Mbits/sec/user. We choose  $\alpha^i = 0.4698(-3.28 \text{ dB}) \forall i$ . In each slot, we generate  $X^i$  using exponential distribution with mean  $\alpha^i$ . We determine

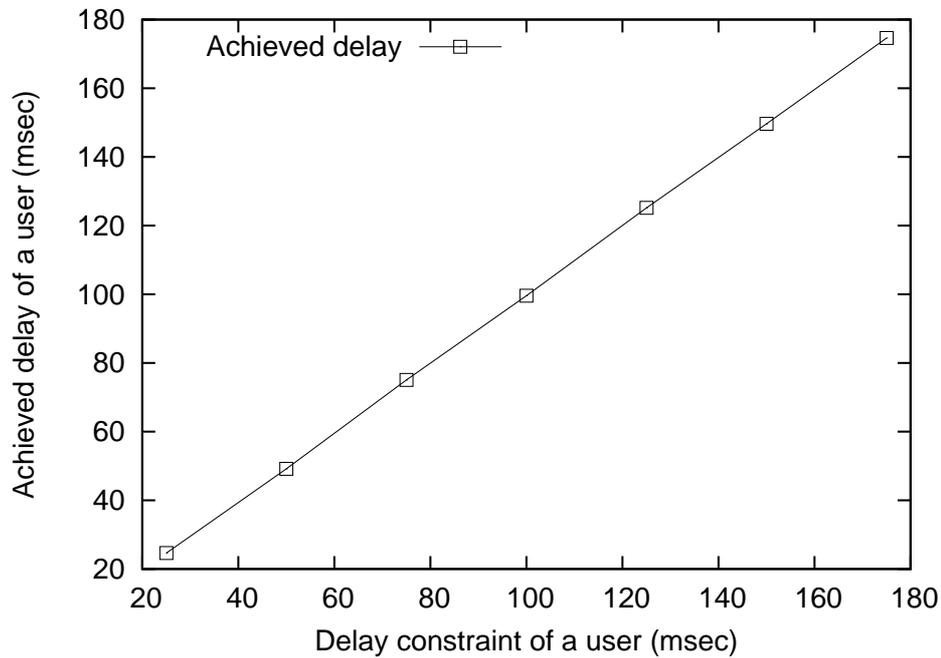


Figure 4.4: Achieved delay of a user with specified average delay constraints - symmetric case

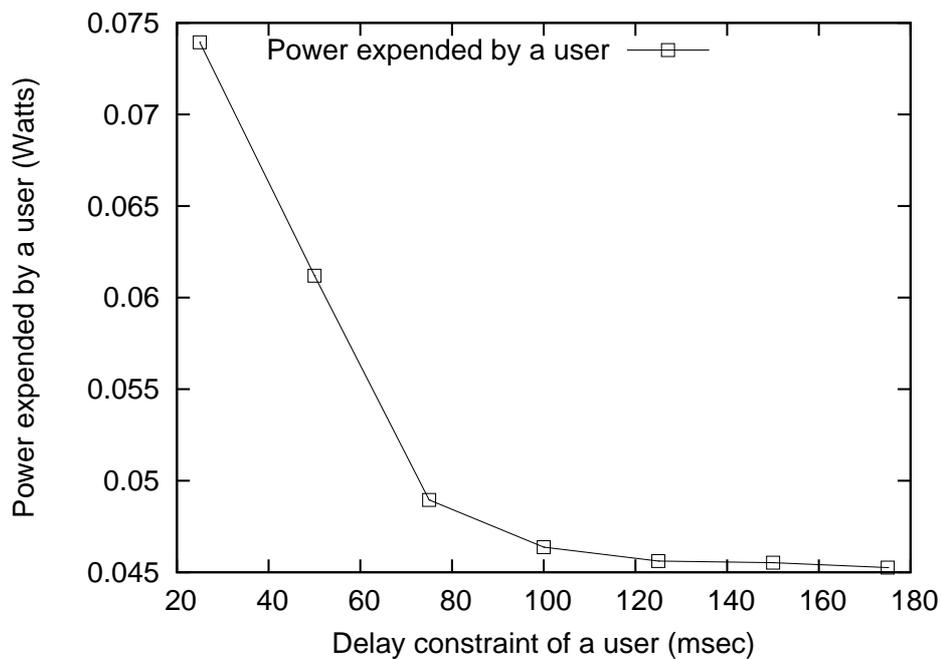


Figure 4.5: Power expended with specified average delay constraints - symmetric case

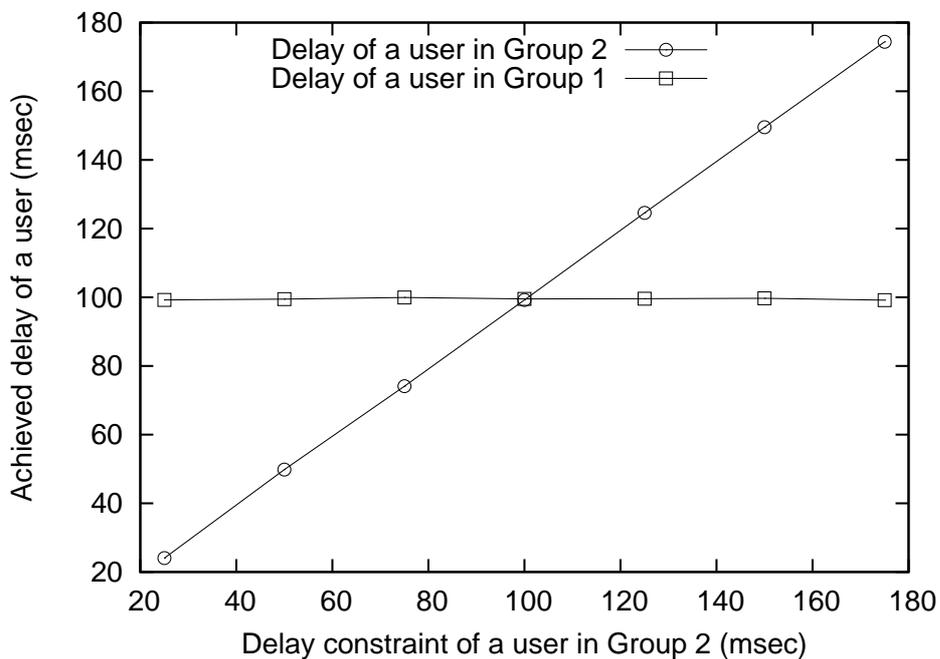


Figure 4.6: Achieved delay of a user with specified average delay constraints - asymmetric case

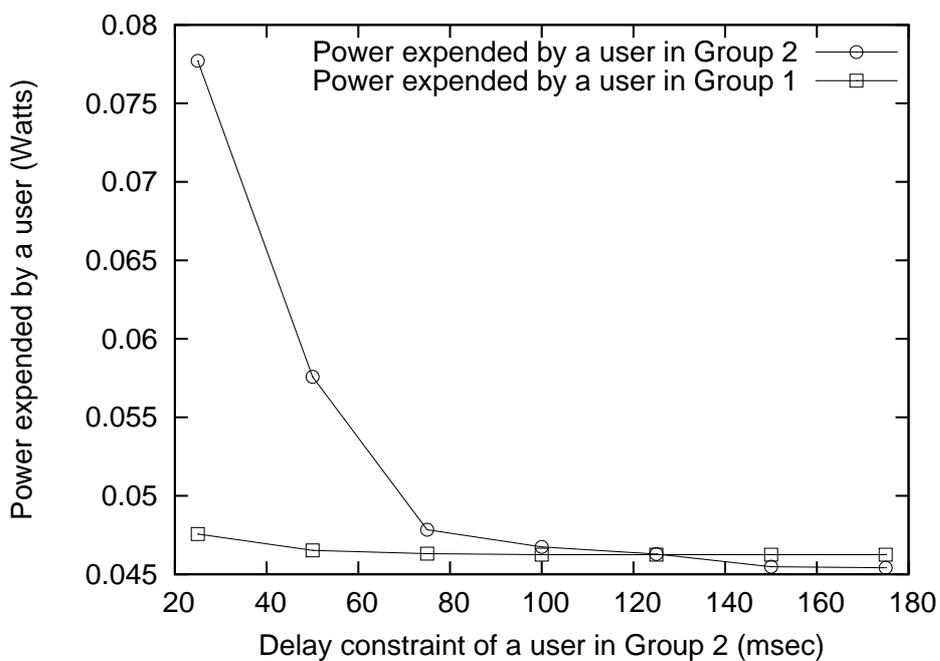


Figure 4.7: Power expended with specified average delay constraints - asymmetric case

Simulation Parameter	Value
Delay constraint (Group 1&2), symmetric case	25 – 175 msec in steps of 25 msec
Delay constraint (Group 1), asymmetric case	100 msec
Delay constraint (Group 2), asymmetric case	25 – 175 msec in steps of 25 msec
Mean arrival rate (Group 1&2), both cases	0.386 Mbps/user
Mean channel state (Group 1&2), both cases	0.4698 (–3.28 dB)

Table 4.2: Summary of parameters for Scenario 4.1

Simulation Parameter	Value
Mean channel state, symmetric case	–13 to 1.42 dB in 7 steps
Mean channel state (Group 1), asymmetric case	–3.28 dB
Mean channel state (Group 2), asymmetric case	–13 to 1.42 dB in 7 steps
Mean arrival rate (Group 1&2), both cases	0.386 Mbps/user
Delay constraint (Group 1&2), both cases	100 msec

Table 4.3: Summary of parameters for Scenario 4.2

the channel state based on the bin that contains  $X^i$  as explained above.

We perform multiple experiments. In the symmetric case, in successive experiments, the delay constraints of all the users are fixed at 25, 50, 75, 100, 125, 150, 175 msec respectively. We measure the average delay experienced and the average power expended by each user in each experiment. In the asymmetric case, the delay constraint of the users in Group 1 is fixed at 100 msec in each experiment, while the delay constraints of the users in Group 2 are fixed at 25, 50, 75, 100, 125, 150, 175 msec in successive experiments. The parameters used in this scenario have been summarized in Table 4.2. Results for this scenario are illustrated in Figures 4.4, 4.5, 4.6 and 4.7. It can be observed from Figures 4.4 and 4.6 that the delay constraints are satisfied in both the cases. Moreover, from Figures 4.5 and 4.7, it can be observed that the power expended is a convex decreasing function of the delay constraint. Larger delay constraints imply that lesser power is required to satisfy the constraint.

*Scenario 4.2.* In this scenario, we demonstrate that the AA satisfies the user specified

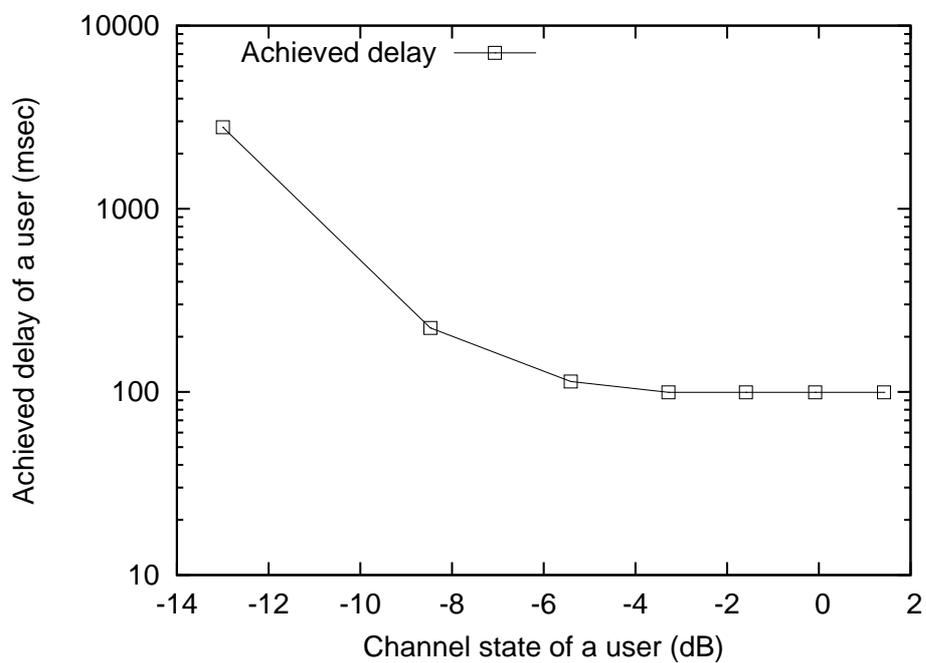


Figure 4.8: Achieved delay of a user with varying average channel states - symmetric case

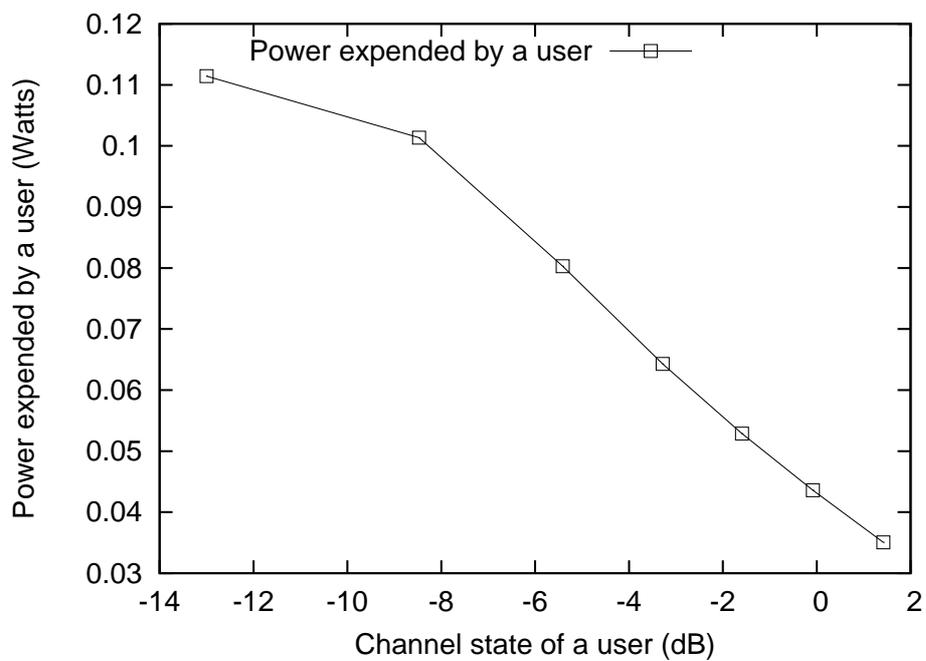


Figure 4.9: Power expended with varying average channel states - symmetric case

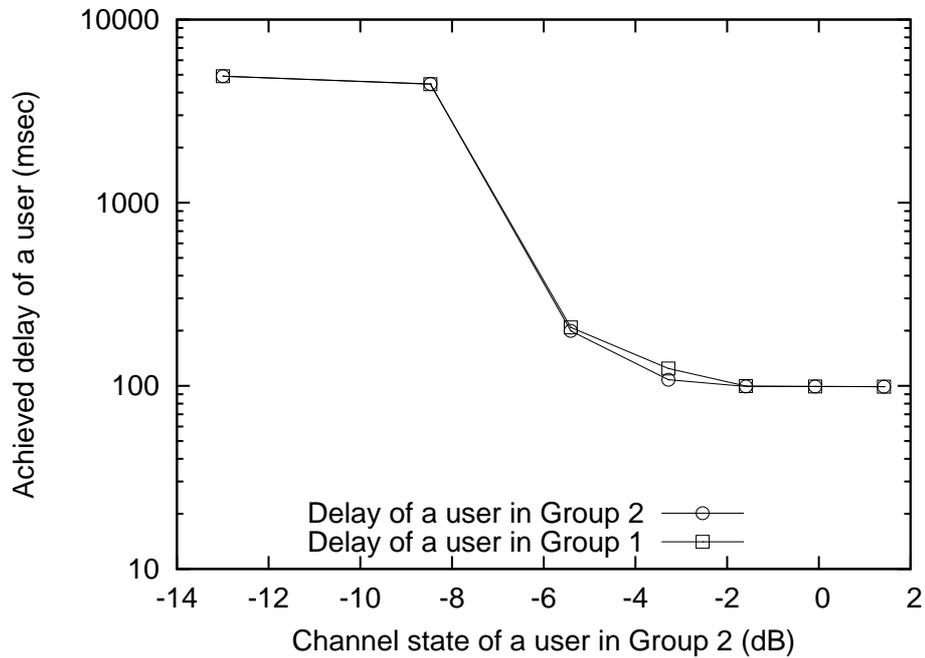


Figure 4.10: Achieved delay of a user with varying average channel states - asymmetric case

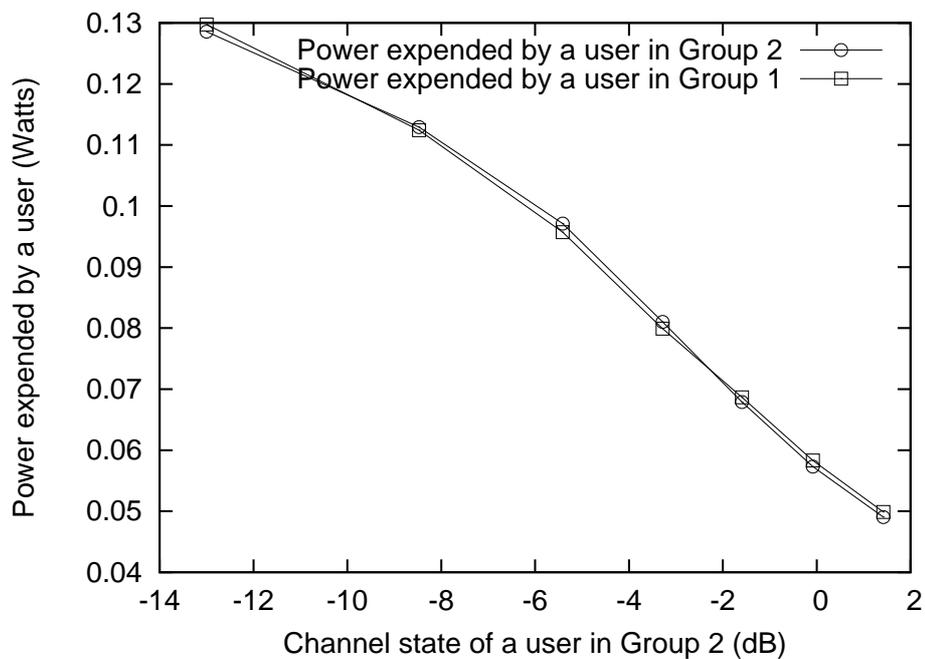


Figure 4.11: Power expended with varying average channel states - asymmetric case

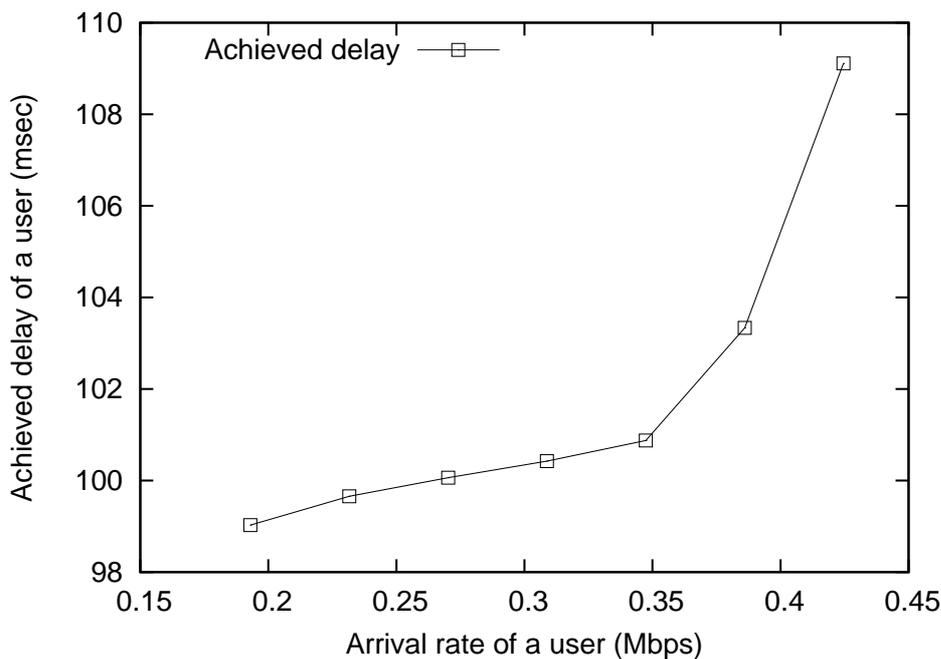


Figure 4.12: Achieved delay of a user with varying average arrival rates - symmetric case

delay constraint for various average channel states. We consider two cases: symmetric and asymmetric. The delay constraint of all users is kept constant at 100 msec. For the symmetric case, we fix  $\alpha^i$  as  $-13$  dB,  $-8.47$  dB,  $-5.41$  dB,  $-3.28$  dB,  $-1.59$  dB,  $-0.08$  dB,  $1.42$  dB,  $\forall i$  in successive experiments. Rest of the parameters are the same as in Scenario 4.1. The parameters for this scenario have been summarized in Table 4.3. We measure the average delay and the average power of each user. These quantities are plotted in Figures 4.8 and 4.9 respectively. In the asymmetric case, we maintain the average channel state for users in Group 1 constant for all the experiments, i.e.,  $\alpha^i = -3.28$  dB,  $i \in 1, \dots, 10$ . For the users in Group 2, i.e.,  $\alpha^i$  for  $i \in 11, \dots, 20$ , the average channel state is fixed at  $\alpha^i = -13$  dB,  $-8.47$  dB,  $-5.41$  dB,  $-3.28$  dB,  $-1.59$  dB,  $-0.08$  dB,  $1.42$  dB, in successive experiments. The average delay suffered by a user in Group 1 and in Group 2 and the power consumed by them are plotted in Figures 4.10 and 4.11 respectively. From Figures 4.8 and 4.10, it can be observed that the scheme is able to satisfy the delay constraints above a certain average channel state<sup>4</sup>. From Figure 4.9 and Figure 4.11, it can be observed that better average channel states result in much lesser power being required for satisfying the delay constraints.

*Scenario 4.3.* In this scenario, we demonstrate the range of arrival rates for which the

<sup>4</sup>This average channel state is dependent on the peak transmission power.

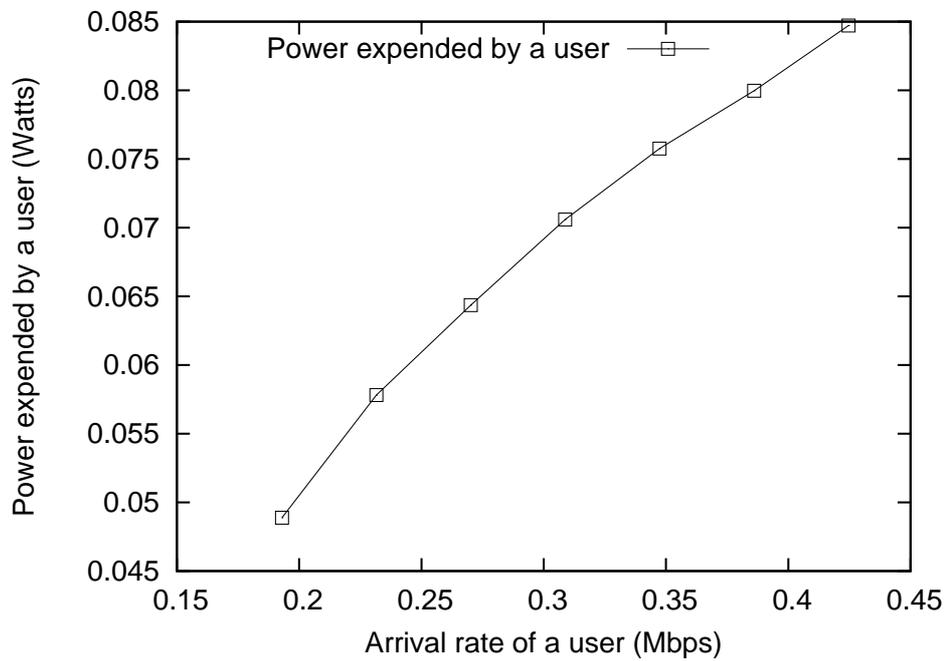


Figure 4.13: Power expended with variation in average arrival rate - symmetric case

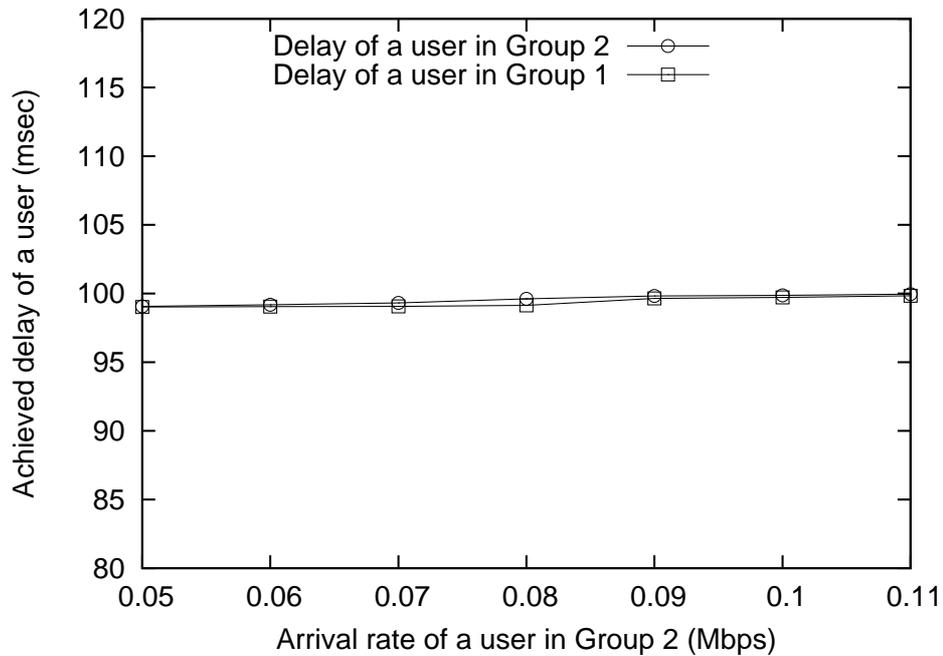


Figure 4.14: Achieved delay of a user with varying average arrival rates - asymmetric case

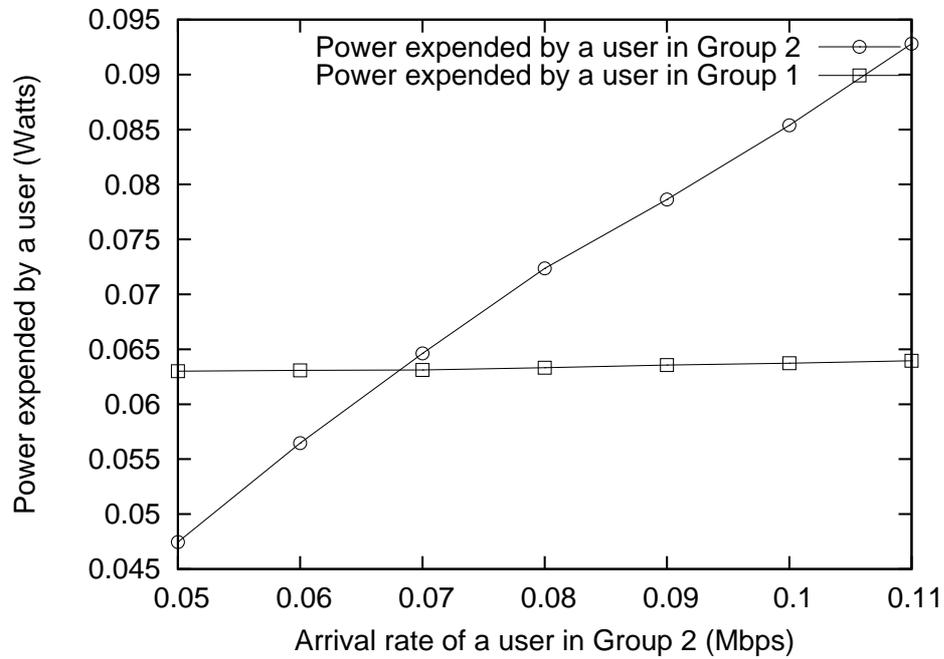


Figure 4.15: Power expended with variation in average arrival rate - asymmetric case

Simulation Parameter	Value
Mean arrival rate (Group 1&2), symmetric case	0.1930 to 0.4246 Mbps/user in 7 steps
Mean arrival rate (Group 1), asymmetric case	0.2702 Mbps/user
Mean arrival rate (Group 2), asymmetric case	0.1930 to 0.4246 Mbps/user in 7 steps
Mean channel state (Group 1&2), both cases	-3.28 dB
Delay constraint (Group 1&2), both cases	100 msec

Table 4.4: Summary of parameters for Scenario 4.3

AA satisfies the user specified delay constraint of 100 msec. We consider two cases - symmetric and asymmetric. In the symmetric case, the arrival rates of all the users are fixed at 0.1930 to 0.4246 Mbits/sec (0.05 to 0.11 packets/msec) in successive experiments. Rest of the parameters are same as in Scenario 4.1. We measure the average delay suffered and the average power expended by each user. These quantities for a user chosen at random are plotted in Figures 4.12 and 4.13 respectively. In the asymmetric case, the arrival rate of the users in Group 1 is fixed at 0.2702 Mbits/sec (0.07 packets/msec) for all the experiments, while the arrival rates of the users in Group 2 are increased from 0.1930 to 0.4246 Mbits/sec (0.05–0.11 packets/msec) in 7 steps in successive experiments. Rest of the parameters are same as in Scenario 4.1. A summary of the parameters is provided in Table 4.4. Average delay suffered by a user from Group 1 and Group 2 (each selected at random) and power consumed by them are plotted in Figures 4.14 and 4.15 respectively. Figures 4.12 and 4.14 demonstrate the range of arrival rates for which the delay constraints are satisfied. From Figures 4.13 and 4.15 it can be seen that power expended is an increasing function of the average arrival rate for the same delay constraint. Higher the arrival rate, higher is the power expended.

## 4.7 Conclusions

In this chapter, we have considered the problem of energy efficient uplink scheduling in a TDMA system over a fading wireless channel. Specifically, we have proposed a novel scheduling algorithm for minimizing the average power of each user subject to individual user constraint. We have provided a novel extension of single user optimal algorithm of Chapter 3 to the multiuser setting. In our approach, the users can be thought of bidding their rates to the base station which then schedules the user bidding the highest rate. We note that it is not in the interest of users to bid unnecessarily higher rates as that might result in higher power consumption. We have argued that the algorithm satisfies the delay constraints of the users and that the algorithm has a stabilizing structure. Another advantage of our approach is that it does not require an explicit knowledge of the probability distribution of channel state and arrival process. The algorithm has low computational complexity and communication overheads. It thus provides a powerful framework for uplink scheduling.

## Chapter 5

# Throughput Efficient Scheduling for Multiuser Downlink

In the previous two chapters, we have focused on energy efficient scheduling. In this chapter, we consider the problem of scheduling users on the downlink of a Time Division Multiplexing (TDM) system with constraints on the average packet delays over a fading wireless channel. On the downlink, typically, the base station transmits at a constant power. Hence, the scheduling problem consists of determining the user to which the base station transmits and transmission rate based on the channel state.

For a multiuser queuing system with scheduler on a TDM channel, there is an extensive literature that we have reviewed in Chapter 2. However, the specific optimization problem of maximizing the sum throughput subject to constraints on the individual user delays has not been explicitly addressed so far. We show in Section 5.2 that this problem has the structure of a CMDP. As in the case of multiuser scheduling discussed in the previous chapter, the primary difficulty in computing an optimal policy lies in large state space size that increases exponentially with number of users. Moreover, as already argued in Chapters 3 and 4, computation of such a policy requires knowledge of the system model, i.e., knowledge of the probability distributions of the channel state and the arrival process for each user, which as pointed out earlier, is not available in practice.

We believe that state space explosion and unknown system model are the primary reasons for inadequate attention towards optimal delay constrained multiuser scheduler. We address this problem by proposing a sub-optimal scheduler that is based on computing

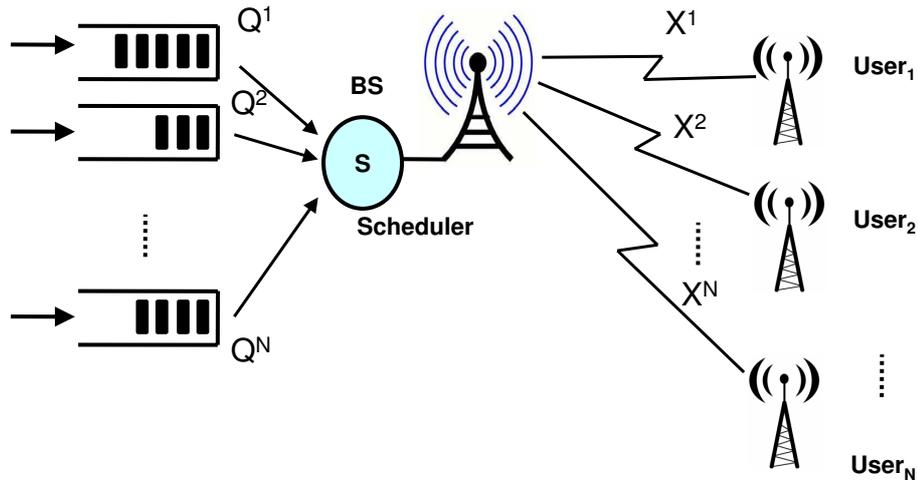


Figure 5.1: Downlink transmission schematic, finite buffer for each user at base station

appropriate indices and scheduling the user with the highest index. The scheme generates indices in each slot in such a fashion that the delay constraints of the users are satisfied while still achieving a very high sum throughput. We demonstrate the applicability of our algorithm to an IEEE 802.16 based system through simulation experiments. Since no other scheme exists for the problem considered in this chapter, for comparison purposes, we adapt the M-LWDF scheduler to our scenario to illustrate that our algorithm achieves a high sum throughput even while satisfying the delay constraints.

The rest of the chapter is organized as follows. We introduce the system model in Section 5.1. In section 5.2, we first formulate the problem as a multistage optimization problem. We then cast it within the CMDP framework and subsequently point out the difficulties in determining the optimal policy within this framework. This motivates the need for a heuristic algorithm. In Section 5.3, we propose an Indexing Scheduler (IS) and prove that the scheme satisfies the delay constraints of the users. Section 5.4 presents the simulation results. We conclude in Section 5.5.

## 5.1 System Model

The system model (depicted in Figure 5.1) is similar to those of Section 2.3.1. However, for the sake of completeness, we recapitulate the important assumptions. We consider downlink transmissions in a TDM system with  $N$  users. Time is divided into slots of

equal duration.  $Q_n^i \in \mathbb{Q} \triangleq \{0, \dots, B\}, \forall i$ .  $\{A_n^i\}$  is the i.i.d. packet arrival process for user  $i$  and the probability distribution of  $A_n^i$  is not known. Packets are of constant length equal to  $\ell$  bits.  $\{X_n^i\}$  is an i.i.d. channel state process. The probability distribution of  $X_n^i$  is not known to the base station. However, the base station has the perfect knowledge of the value of  $X_n^i, \forall i$  in each slot. The base station transmits at a constant power  $P_m$  in each slot.  $U_n^i$  denotes the maximum rate at which the base station can transmit ‘reliably’ to user  $i$  in slot  $n$ . Moreover,  $U_n^i \leq Q_n^i$ . The queue transition equation for user  $i$  is expressed as<sup>1</sup>:

$$Q_{n+1}^i = Q_n^i + A_{n+1}^i - I_n^i U_n^i, \quad (5.1)$$

where if a user  $i$  is scheduled in a slot  $n$ ,  $I_n^i = 1$  and for the rest  $I_n^j = 0, j \neq i$ . Let the vector  $\mathbf{I}_n = [I_n^1, \dots, I_n^N]^T, \mathbf{I} \in \mathbb{I}$  in a slot  $n$  be a vector such that only one element of the vector is 1 and the rest are 0.

## 5.2 Problem Formulation

In this section, we first formulate the problem as a constrained optimization problem. We then cast the problem within the CMDP framework. Subsequently, we indicate the difficulties in determining the optimal policy within the CMDP framework.

### 5.2.1 Formulation as a Constrained Optimization Problem

Recall that  $\mathbf{Q}_n \in \mathbb{Q}^N, \mathbf{X}_n \in \mathbb{X}^N, \bar{\mathbf{a}}, \bar{\mathbf{Q}}, \mathbf{U}_n$  and  $\bar{\boldsymbol{\delta}}$  denote the  $N$  dimensional vectors corresponding to the queue lengths of the users in slot  $n$ , channel states of the users in slot  $n$ , the average arrival rates of the users, the average queue lengths of the users over a long run, number of packets transmitted in slot  $n$  and the queue length constraints of the users respectively. The sum throughput over a long period of time can be expressed as:

$$\bar{T} = \liminf_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \mathbf{I}_n^T \mathbf{U}_n. \quad (5.2)$$

---

<sup>1</sup>Following the discussion in Chapter 3, we assume that, as compared to the packet arrival rate (and the average queue length/delay constraint),  $B$  is large enough so as to neglect the buffer overflow and hence packet drops.

and the queue length averaged over a long period of time can be expressed as:

$$\bar{\mathbf{Q}} = \liminf_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \mathbf{Q}_n \quad (5.3)$$

Intuitively, if the system adopts a *pure* opportunistic policy, i.e., schedule the user with the *best* channel state, the sum throughput will be maximized, but the delay constraints may not be satisfied. At times, the system has to schedule a user who does not have the best channel state in order to meet its delay requirements. Thus, there exists a tradeoff between maximizing the sum throughput and satisfying the user delay constraints. The scheduling problem can, therefore, be expressed as a constrained optimization problem:

$$\text{Maximize } \bar{T} \text{ subject to } \bar{\mathbf{Q}} \leq \bar{\boldsymbol{\delta}}. \quad (5.4)$$

### 5.2.2 Formulation within CMDP Framework

The state of the system in slot  $n$  is specified by the tuple  $\mathbf{S}_n \triangleq (\mathbf{Q}_n, \mathbf{X}_n)$ . The system state space  $\mathbb{S}^N = \mathbb{Q}^N \times \mathbb{X}^N$  is discrete and finite. In each slot, the scheduler selects a particular user  $i$  based on the system state, i.e., the control action is to select  $i$  such that  $I_n^i = 1$ ,  $I_n^j = 0$ ,  $\forall j \neq i$  in the vector  $\mathbf{I}_n$ . The state of the system in the next slot depends on its current state and the decision taken by the scheduler. Hence, the problem has the structure of a CMDP [59]. We define the following two quantities, throughput reward (throughput achieved by scheduling a user in a slot  $n$ ):

$$T(\mathbf{S}_n, \mathbf{I}_n) \triangleq \mathbf{I}_n^T \mathbf{U}_n, \quad (5.5)$$

and queuing cost:

$$c_q(\mathbf{S}_n, \mathbf{I}_n) \triangleq \mathbf{Q}_n. \quad (5.6)$$

The objective is to determine an optimal policy (i.e., a mapping from the history of the states and actions to the vector  $\mathbf{I}_n$ ) that achieves the maximum average throughput reward subject to the delay constraints of the users, i.e., achieves an average queuing cost below a prescribed bound.

Since the state space and action space are discrete and finite, it can be shown that an optimal stationary randomized policy exists [59]. For stationary randomized policies characterized by  $\mu(\cdot|\mathbf{s}) : \mathbf{s} \in \mathbb{S}^N \rightarrow$  probability measures on  $\mathbb{I}$ ,  $\mu(\cdot|\mathbf{s})$  for each state  $\mathbf{s}$  specifies the distribution with which the control in that state is applied. Under the assumption

of irreducibility of the Markov chain under such policies,  $\{\mathbf{S}_n\}$  is an ergodic Markov chain and thus has a unique stationary distribution  $\rho^\mu$ . Let  $\mathbf{E}^\mu$  denote the expectation with respect to (w.r.t.)  $\rho^\mu$ . Under a randomized policy  $\mu$ , the time averaged throughput in (5.2) can be expressed as:

$$\bar{T}^\mu = \mathbf{E}^\mu \left[ \sum_{i=1}^N \mu(\mathbf{S}_n)^T \mathbf{U}_n \right] = \sum_{\mathbf{I}, \mathbf{s}} \rho^\mu(\mathbf{s}) \mu(\mathbf{I}|\mathbf{s}) T(\mathbf{s}, \mu(\mathbf{s})). \quad (5.7)$$

and time averaged queue length in (5.3) can be expressed as:

$$\bar{\mathbf{Q}}^\mu \triangleq \mathbf{E}^\mu \left[ c_q(\mathbf{S}_n, \mu(\mathbf{S}_n)) \right] = \sum_{\mathbf{I}, \mathbf{s}} \rho^\mu(\mathbf{s}) \mu(\mathbf{I}|\mathbf{s}) c_q(\mathbf{s}, \mu(\mathbf{s})). \quad (5.8)$$

Then the scheduler objective can be stated as:

$$\text{Maximize } \bar{T}^\mu \text{ subject to } \bar{\mathbf{Q}}^\mu \leq \bar{\boldsymbol{\delta}}. \quad (5.9)$$

### 5.2.3 Issues in Determining the Optimal Policy

The traditional approaches based on Linear Programming (LP) [59] for determining the optimal policy cannot be used to solve (5.9) because of the following reasons:

1. *Large state space:* As discussed in Chapter 4, the state space for the multiuser system is large even for moderate number of users and the state space size increases exponentially with the number of users. The computational complexity of the traditional LP based approaches is proportional to the state space size [59] and hence the computational complexity also increases exponentially with users.
2. *Unknown system model:* We do not impose model related restrictions and hence assume that the system model is unknown.

Hence, we develop an indexing scheme which employs iterative update of a quantity termed as ‘weight’ associated with a user in order to resolve the issue of unknown system model. The proposed scheme is an indexing scheme that generates appropriate indices for the users in each slot and schedules the user having the highest index. Though the scheme is sub-optimal, yet it performs very well in practice. We present the details regarding the indexing scheme in the next section.

### 5.3 Indexing Scheduler

We seek to generate indices that exploit the tradeoff between maximizing throughput and satisfying delay constraints. The user having maximum index in a slot is scheduled in the slot. Note that maximizing the sum throughput requires that a user with the best channel state be scheduled in a slot. On the other hand, if the queue length of a user exceeds the queue length constraint, then the scheduler has to compromise on the objective of maximizing the sum throughput and possibly schedule a user not having the best channel state. These considerations can be precisely expressed as follows:

1. To fulfill the objective of maximizing the sum throughput, an index must be proportional to the channel state of the user. This ensures that a user with better channel state has a higher probability of being scheduled.
2. The index allocation must be cognizant of the user delay requirements. A user having tighter delay constraint must be given higher index and hence higher probability of being scheduled. If the slots allocated to a user are not sufficient to satisfy its delay constraint, its queue length would be greater than the desired queue length frequently. In order to satisfy the delay constraint of a user, its index must be proportional to the aggregate amount with which its queue length exceeds the desired queue length. This ensures that the user having a higher aggregate queue constraint violation has a greater probability of being scheduled.

Taking these requirements into consideration, we define the index  $\iota_n^i$  of a user  $i$  in a slot  $n$  as:

$$\iota_n^i = \lambda_n^i \times U_n^i. \quad (5.10)$$

$\lambda_n^i$  is the weight of user  $i$  in slot  $n$ . This weight is dynamically adjusted in each slot based on the deviation of the queue length of that user from its desired queue length. Once the indices are determined, the algorithm determines the user with the highest index with a non-empty queue and non-zero rate, and schedules this user. If there are multiple such users, one of them is scheduled randomly with uniform probability. We now describe an approach for determining the weight  $\lambda_n^i$  for a user  $i$  in slot  $n$ .

### 5.3.1 Determining Weights

As outlined above, if the aggregate queue constraint violation of a user is large, it must have a large weight. Hence, we dynamically update the weight in each slot by adding the deviation of the current queue length from the constraint to it. Consider sequence  $\{f_n\}$  that satisfies the following properties:

$$\lim_{n \rightarrow \infty} f_n = 0, \quad \lim_{n \rightarrow \infty} \sum_n (f_n)^2 < \infty, \quad \lim_{n \rightarrow \infty} \sum_n f_n = \infty. \quad (5.11)$$

The first two properties in (5.11) ensure that the sequence  $\{f_n\}$  converges to zero sufficiently rapidly, while the third property ensures that it does not converge to zero too rapidly. Let  $\lambda_0^i = 1 \forall i$ . The weight  $\lambda_n^i$  for a user  $i$  in slot  $n$  is then determined using the following iteration:

$$\lambda_{n+1}^i = \min(\Gamma, \max(0, \lambda_n^i + f_n \times (Q_n^i - \bar{\delta}^i))), \quad (5.12)$$

where  $\Gamma \gg 0$ , i.e., we project the  $\lambda^i$  iterates in the interval  $[0, \Gamma]$ . The properties of  $\{f_n\}$  ensure that the update rate of weight  $\lambda^i$  is neither too fast nor too slow (following arguments similar to those of [25]). The stable value of the weight determines the proportion of slots allocated to a user based on its delay requirement and thereby the relative priority between the users. The intuition behind (5.12) is to iteratively tune the weight of user  $i$  so as to satisfy its delay constraint. If  $Q_n^i$  continues to be less than  $\bar{\delta}^i$  then it progressively reduces the weight  $\lambda_n^i$  in the subsequent slots thereby reducing the probability of user  $i$  being scheduled. On the other hand, if  $Q_n^i$  continues to be more than  $\bar{\delta}^i$ , then it progressively increases the weight  $\lambda_n^i$  thereby increasing the probability of user  $i$  being scheduled and hence increasing the proportion of slots that would be allocated to user  $i$ . The weight update results in a redistribution of the proportion of slots allocated to users. If the delays are feasible, the scheme determines an allocation such that the delay constraints of all the users are satisfied.

**Theorem 5.1.**  $\lambda_n$  iterates converge to a stable value  $\lambda_*$ .

*Proof.* Let  $\lambda_n = [\lambda_n^1, \dots, \lambda_n^N]^T$  denote the weight vector in slot  $n$ . (5.12) can be expressed in the vector form as:

$$\lambda_{n+1} = \min(\mathbf{\Gamma}, \max(\mathbf{0}, \lambda_n + f_n \times (\mathbf{Q}_n - \bar{\delta}))). \quad (5.13)$$

We consider the  $\boldsymbol{\lambda}$  and  $\mathbf{Q}$  values after  $\tau$  slots for large  $\tau$ . Let  $b_l$  denote the value of  $b$  at the  $(\tau \times l)$  th slot. Note that if the weight  $\lambda^i$  of user  $i$  is increased, over a period of time, its queue length  $Q^i$  reduces, thus increasing  $\bar{\delta}^i - Q^i$ . We model this effect using the following equation:

$$\mathbf{Q}_{l+1} - \bar{\boldsymbol{\delta}} = G(\boldsymbol{\lambda}_l) \implies \mathbf{Q}_{l+1} = \bar{\boldsymbol{\delta}} + G(\boldsymbol{\lambda}_l), \quad (5.14)$$

where  $G(\cdot)$  is a monotonically non-increasing continuous function of  $\boldsymbol{\lambda}$ . Moreover, if  $Q^i - \bar{\delta}^i$  increases,  $\lambda^i$  increases. We model this effect using the following equation:

$$\boldsymbol{\lambda}_{l+1} = F(\mathbf{Q}_l - \bar{\boldsymbol{\delta}}), \quad (5.15)$$

where  $F(\cdot)$  is a monotonically non-decreasing continuous function of  $\mathbf{Q}_l - \bar{\boldsymbol{\delta}}$ . (5.14) and (5.15) form the following fixed point iteration:

$$\boldsymbol{\lambda} = F(G(\boldsymbol{\lambda})). \quad (5.16)$$

Since  $F(G(\cdot))$  is a composition of continuous functions, it is continuous. Thus we have a continuous mapping from  $[0, \boldsymbol{\Gamma}]$  to  $[0, \boldsymbol{\Gamma}]$ . Hence, by Brouwer's fixed point theorem [99], there exists a fixed point in  $[0, \boldsymbol{\Gamma}]$ .  $G(\boldsymbol{\lambda})$  being non-increasing in  $\boldsymbol{\lambda}$ ,  $F(G(\boldsymbol{\lambda}))$  is non-increasing in  $\boldsymbol{\lambda}$ . Hence, the fixed point is unique, say  $\boldsymbol{\lambda}_*$  which is denoted as the stable value.

□

*Remark 5.1.* Convergence of the weights to stable values indicates that the scheme allocates slots to the users in such a proportion that their delay constraints are satisfied. Hence, the scheme can be considered to be a delay satisfying sub-optimal algorithm.

### 5.3.2 Implementation Details

Based on (5.10), and (5.12), the base station scheduler implements the scheme as explained in Algorithm 4. We assume that the scheduler is aware of the value of channel state  $X^i$  in each time slot. In each time slot, it determines the number of packet arrivals, channel state and current queue length for each user. The number of packets  $U_n^i$  to be transmitted to each user is then determined based on maximum transmission power and the channel state for the user. The scheduler then determines the index for each user and schedules the

user with the highest index. The weights are then appropriately adjusted. The algorithm thus continues in each slot.

```

1: Initialize  $\iota_0^i \leftarrow 0$   $\lambda_0^i \leftarrow 0$ ,  $n \leftarrow 1$ , queue length  $Q_0^i \leftarrow 0$ , channel state  $X_0^i \leftarrow 0$  for all  $i$ .
2: while TRUE do
3:   Determine number of arrivals  $A_{n+1}^i$  and channel state  $X_n^i$  in the current slot for all  $i$ .
4:   Determine  $U_n^i$ ,  $\forall i$ .
5:   Determine index  $\iota_n^i$ ,  $\forall i$  using (5.10).
6:   Schedule the user  $j$  having the largest index.
7:    $I_n^j \leftarrow 1$ .
8:    $I_n^i \leftarrow 0$ ,  $\forall i \neq j$ .
9:   Update queue length using (5.1).
10:  Update the weight  $\lambda_n^i$ ,  $\forall i$  using (5.12).
11:   $n \leftarrow n + 1$ .
12: end while

```

**Algorithm 4:** Indexing Scheduler (IS)

## 5.4 Experimental Evaluation

In this section, we demonstrate the following through the simulations within the IEEE 802.16 framework [5]:

1. The algorithm satisfies the delay constraints of all users.
2. The algorithm is efficient in terms of the achieved sum throughput by its comparison with the M-LWDF scheduler [16].

M-LWDF scheme considers the probability with which a users queue length is allowed to exceed a certain target queue length. We assume that this probability is the same for all users and ignore it in the present simulations. Specifically, the adapted M-LWDF schedules a user  $i$  in each slot such that,

$$i = \arg \max_j \tau_n^j \times U_n^j, \quad (5.17)$$

where  $\tau_n^j$  is the delay experienced by the head of the line packet for user  $j$ . M-LWDF scheme transmits at a constant power in each time slot. In Scenario 5.2 considered below, we first determine the average delays experienced by the users under the M-LWDF scheme for various average arrival rates. The values of these delays are then considered to be the delay constraints for the indexing scheduler. We determine the average delays experienced by the users under the IS and also the sum throughput achieved under it.

We perform the simulations within the framework of an IEEE 802.16 system, details of which are provided next.

### 5.4.1 The IEEE 802.16 System

We consider the DL transmissions in the *residential* scenario as in [97] where the BS provides Internet access to the subscribers. IEEE 802.16 nrtPS can be extended for providing average delay requirements on the downlink to non-real time applications. The unicast polling service of nrtPS can be used to determine the channel state perceived by the users. On the downlink, the base station has knowledge of the queue lengths of all users. The scheduling algorithm can thus be implemented as a part of nrtPS.

We assume the FDD mode of operation where all SSs have full-duplex capability. We consider a single carrier system with a frame duration of 1 msec and bandwidth of 10 MHz. The SSs employ the following modulations: 64-QAM, 16-QAM, QPSK and QPSK with a rate 1/2 code; along with a filter roll-off factor of 0.22. These provide us with the following 4 rates of transmission: 24 Mbps, 16 Mbps, 8 Mbps and 4 Mbps respectively. We consider 20 connections on the DL and assume that the number of connections does not change over the duration of the simulations. We measure the sum of queuing and transmission delays of the packets and ignore the propagation delays.

### 5.4.2 Simulation Results

Internet traffic is modeled as a web traffic source. As discussed in Chapter 4, we assume that the application layer packets are of variable length and have a truncated Pareto distribution with shape factor  $\xi = 1.2$ , mode  $v = 2000$  bits, cutoff threshold  $c = 10000$  bits, which provides us with an average packet size of 3860 bits. In each time frame, we generate the arrivals for all users using Poisson distribution. Arrivals are generated in an

Simulation Parameter	Value
Slot duration	1 msec
Number of rates	4
Bandwidth	10 MHz
Transmit power	4 Watts
Transmission rates	24, 16, 8, 4 Mbits/sec
Simulation time	100000 slots
$f_n$	$\frac{A}{n}, A > 0$
$\xi$	1.2
$v$	2000
$c$	10000 bits
$N$	20
$\Gamma$	10000

Table 5.1: Summary of parameters common for all scenarios

i.i.d. manner across frames. At the MAC layer, the packets are divided into fragments of size 2000 bits. Smaller fragments are padded with extra bits so that all the fragments are of size 2000 bits.

We simulate a Rayleigh channel for each user. As pointed out in Section 2.1.2, for a Rayleigh model, channel state  $X^i$  is an exponentially distributed random variable with mean  $\alpha^i$  (probability density function expressed in (2.7)). We discretize the channel into eight equal probability bins using the discretization procedure specified in Section 3.6. We fix the transmission power at 4 Watts. Based on this transmission power, we determine the number of fragments to be transmitted for all the users in each slot. We choose  $f_n = \frac{A}{n}$ ,  $A > 0$ . Users are divided into two groups (Group 1 and Group 2) of 10 users each. In all the experiments, each simulation run consists of simulating the algorithms for 100000 frames. Results are presented after averaging over 20 simulation runs. A summary of parameters common to all the simulation scenarios has been provided in Table 5.1.

*Scenario 5.1.* In this scenario, we demonstrate that the algorithm satisfies the various user specified delay constraints. We consider two cases: symmetric case and asymmetric case.

Simulation Parameter	Value
Delay constraint (Group 1&2) symmetric case	25 – 175 msec in steps of 25 msec
Delay constraint (Group 1) asymmetric case	100 msec
Delay constraint (Group 2) asymmetric case	25 – 175 msec in steps of 25 msec
Mean arrival rate (Group 1&2) both cases	0.6562 Mbps/user
Mean channel state (Group 1&2) both cases	0.4698 (–3.28 dB)

Table 5.2: Summary of parameters for Scenario 5.1

In both cases, in each frame, arrivals are generated with a Poisson distribution with mean 170 packets/sec/user. This results in an average arrival rate of 0.6562 Mbits/sec/user. We choose  $\alpha^i = 0.4698$  (–3.28 dB)  $\forall i$ . In each frame, we generate  $X^i$  using exponential distribution with mean  $\alpha^i$ . We determine the channel state based on the bin that contains  $X^i$  as explained above. In the symmetric case, we measure the average delay experienced by each user with different values of delay constraint such as 25, 50, 75, 100, 125, 150, 175 msec. The values of the delays for a particular user (chosen at random) are plotted in Figure 5.2. In the asymmetric case, the delay constraint of the users in Group 1 is fixed at 100 msec, while the delay constraint of the users in Group 2 is varied as 25, 50, 75, 100, 125, 150, 175 msec in successive experiments. The values of the parameters for this scenario have been summarized in Table 5.2. The average delays experienced by two specific users (each selected at random from Group 1 and Group 2) for the IS are plotted in Figure 5.3. It can be seen from Figures 5.2 and 5.3 that the delay constraints are satisfied in both cases for the IS. The sum throughput achieved for both cases is plotted in Figures 5.4 and 5.5 respectively.

*Scenario 5.2.* In this scenario, we demonstrate that the system achieves a high sum throughput. In each frame, arrivals are generated with a Poisson distribution. In successive experiments, the mean arrival rate is fixed at 10, 40, 70, 100, 130, 160, 190 packets/sec/user respectively. This results in an average arrival rate of 0.0386 to 0.7334 Mbps/user in successive experiments. Rest of the parameters are same as in Scenario 5.1. The values of the parameters for this scenario have been summarized in Table 5.3. We first determine the delays experienced by the users and the sum throughput achieved under the adapted M-LWDF scheme. The delays experienced by the users in the M-LWDF scheme

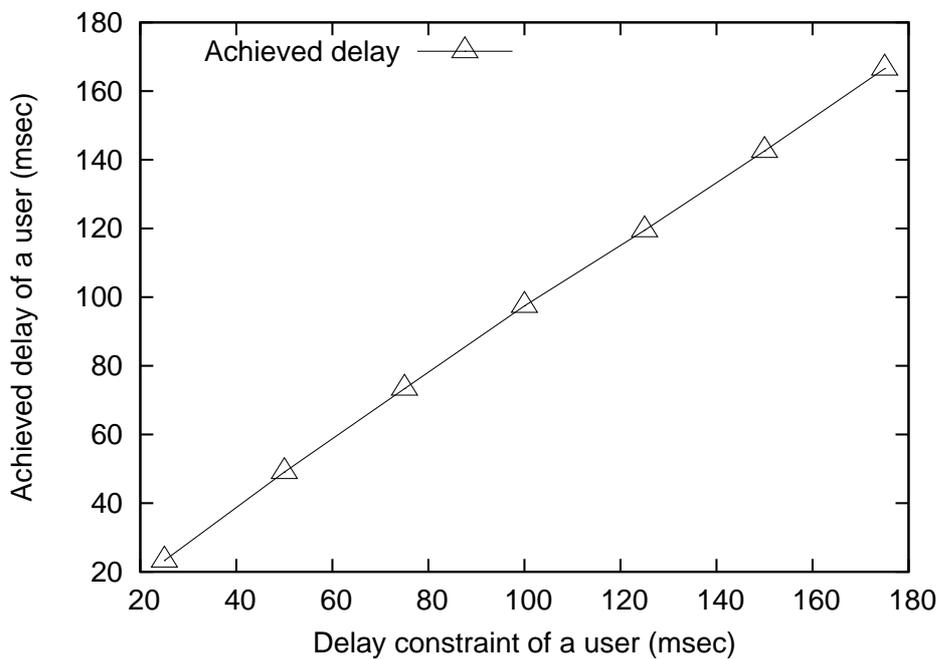


Figure 5.2: Delay experienced by a user selected at random for various average delay constraints - symmetric case

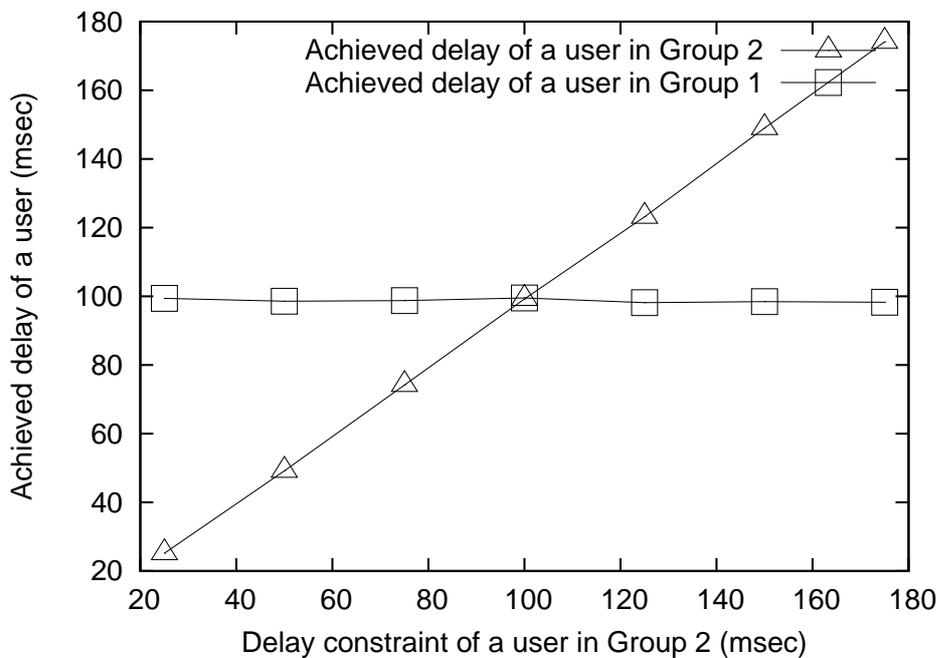


Figure 5.3: Delay experienced by two users selected at random from Group 1 and Group 2 for various average delay constraints - asymmetric case

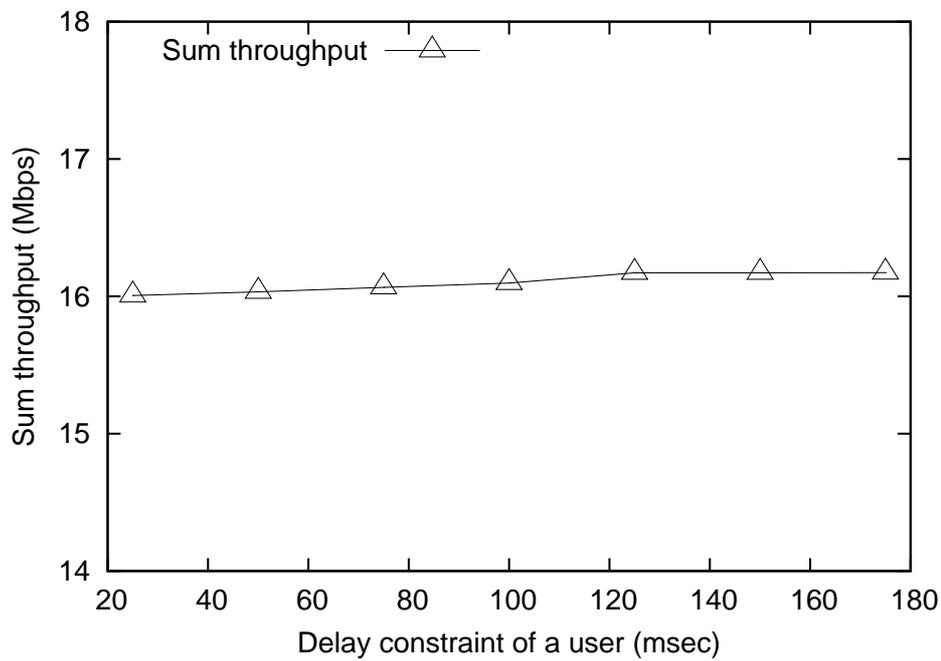


Figure 5.4: Sum throughput for various average delay constraints - symmetric case

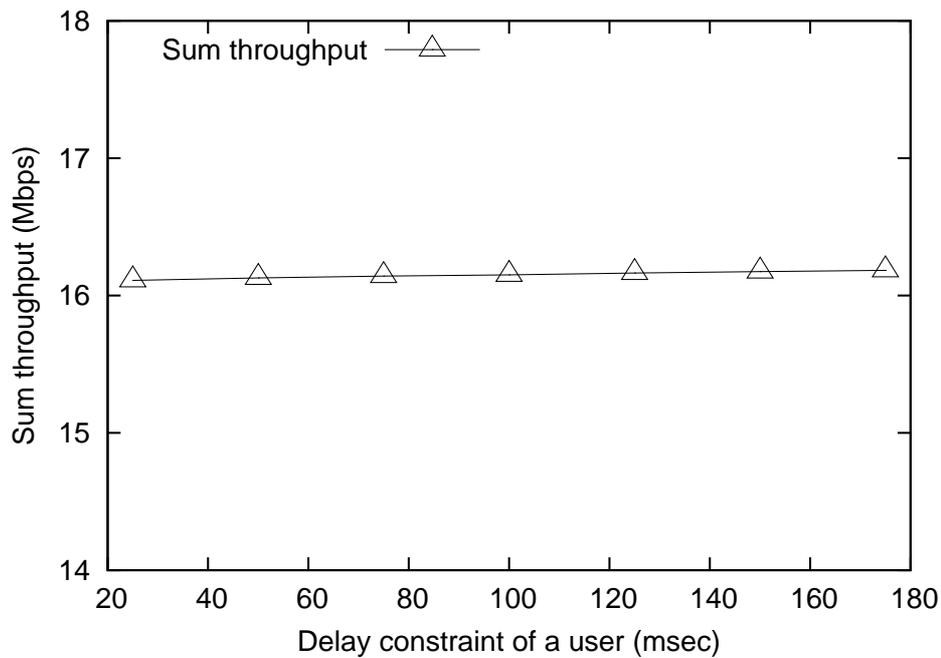


Figure 5.5: Sum throughput for various average delay constraints - asymmetric case

Simulation Parameter	Value
Mean arrival rate (Group 1&2)	0.0386 to 0.7334 Mbps/user in 7 steps
Mean channel state (Group 1&2)	0.4698 (−3.28 dB)

Table 5.3: Summary of parameters for Scenario 5.2

serve as delay constraints for the users in the IS. We determine the delays experienced by a particular user selected at random and the sum throughput achieved under the IS and compare these with those of the M-LWDF scheme in Figures 5.6 and 5.7 respectively. From Figure 5.6 it can be seen that the delays experienced by a user under the IS are less than or equal to those under the M-LWDF scheme implying that the delay constraints are satisfied. Moreover, from Figure 5.7, it can be seen that the sum throughput achieved by the IS is very close to that achieved by the M-LWDF scheme. The IS caters to the delay constraints, and while doing it, we have demonstrated through simulations, that it also achieves a high sum throughput.

*Remark 5.2.* It should be noted that the M-LWDF algorithm does not cater to the objective of satisfying the delay constraints. Its objective is to maintain the delay below a certain prescribed limit with a high probability. Hence, we first simulate the M-LWDF algorithm in order to determine the delay under it and then specify this delay as the constraint for the IS in Scenario 5.2. In situations where the objective is to maintain the average delay below a prescribed limit, the IS would prove to be a better choice than the M-LWDF scheme.

*Scenario 5.3.* In this scenario, we demonstrate that the algorithm satisfies the user specified delay constraint of 100 msec for varying channel conditions. We consider two cases: symmetric case and asymmetric case. In both cases, in each frame, arrivals are generated with a Poisson distribution with mean 170 packets/sec/user. This results in an average arrival rate of 0.6562 Mbps/user. We fix the delay constraint at 100 msec for both cases for all the users. In each frame, we generate  $X^i$  using exponential distribution with mean  $\alpha^i$ . We determine the channel state based on the channel bin that contains  $X^i$  as explained above. In the symmetric case,  $\alpha^i = \alpha, \forall i$ . We measure the average delay experienced by each user with different values of  $\alpha$  such as −13 dB, −8.47 dB, −5.41 dB, −3.28 dB, −1.59

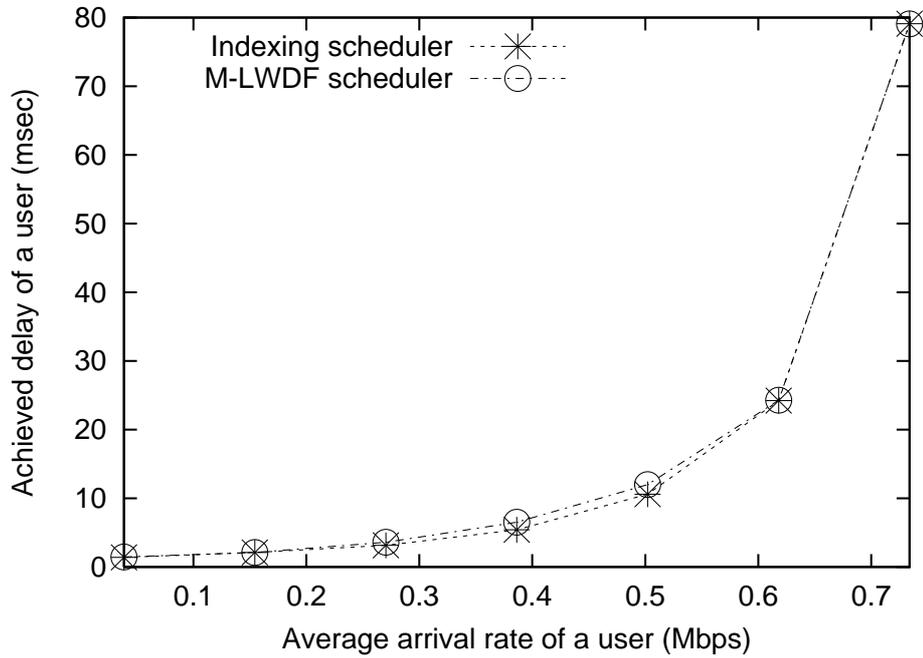


Figure 5.6: Delays experienced under IS with those under M-LWDF scheduler as constraints

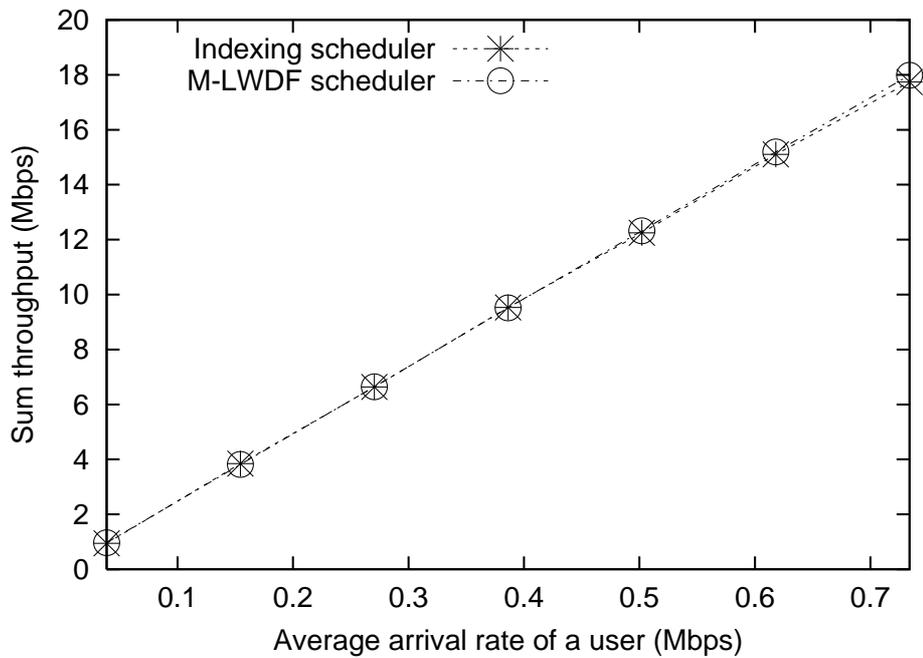


Figure 5.7: Comparison of the sum throughput under M-LWDF and IS

Simulation Parameter	Value
Mean channel state symmetric case	-13 to 3.18 dB in 8 steps
Mean channel state, Group 1, asymmetric case	-3.28 dB
Mean channel state, Group 2, asymmetric case	-13 to 3.18 dB in 8 steps
Mean arrival rate (Group 1&2) both cases	0.6562 Mbps/user
Delay constraint (Group 1&2) both cases	100 msec

Table 5.4: Summary of parameters for Scenario 5.3

dB, -0.08 dB, 1.42 dB, 3.18 dB. The value of the delay for a particular user (chosen at random) are plotted in Figure 5.8. In the asymmetric case, the  $\alpha$  for the users in Group 1 is fixed at -3.28 dB, while  $\alpha$  for the users in Group 2 is varied as -13 dB, -8.47 dB, -5.41 dB, -3.28 dB, -1.59 dB, -0.08 dB, 1.42 dB, 3.18 dB in successive experiments. The parameters for this scenario have been summarized in Table 5.4. The average delay experienced by two specific users (each selected at random from Group 1 and Group 2) for the IS are plotted in Figure 5.9. It can be seen from Figures 5.8 and 5.9 that the delay constraints are satisfied in both cases for the IS for even low values of  $\alpha$ . When the mean channel state is extremely low, (less than -5 dB) however, the delay constraints are not satisfied since the average throughput is very low in this case. The sum throughput achieved for both cases is plotted in Figures 5.10 and 5.11 respectively.

*Scenario 5.4.* In this scenario, we demonstrate that the algorithm satisfies the user specified delay constraint of 100 msec for varying average arrival rates. We consider two cases: symmetric case and asymmetric case. We choose  $\alpha^i = 0.4698$  (-3.28 dB)  $\forall i$  for both cases. In each frame, we generate  $X^i$  using exponential distribution with mean  $\alpha^i$ . We determine the channel state based on the channel bin that contains  $X^i$ . In the symmetric case, we measure the average delay experienced by each user with different arrival rates such as 110, 130, 150, 170, 190, 210, 230, 250 packets/sec/user, i.e., 0.4246 to 0.9650 Mbps/user. The values of the delays for a particular user (chosen at random) are plotted in Figure 5.12. In the asymmetric case, the average arrival rate for the users in Group 1 is fixed at 170 packets/msec/user, i.e., 0.6562 Mbps/user while the average arrival rate of the users in Group 2 is varied as 110, 130, 150, 170, 190, 210, 230, 250 packets/sec/user, i.e., 0.4246 to 0.9650 Mbps/user in successive experiments. The values of the parameters for

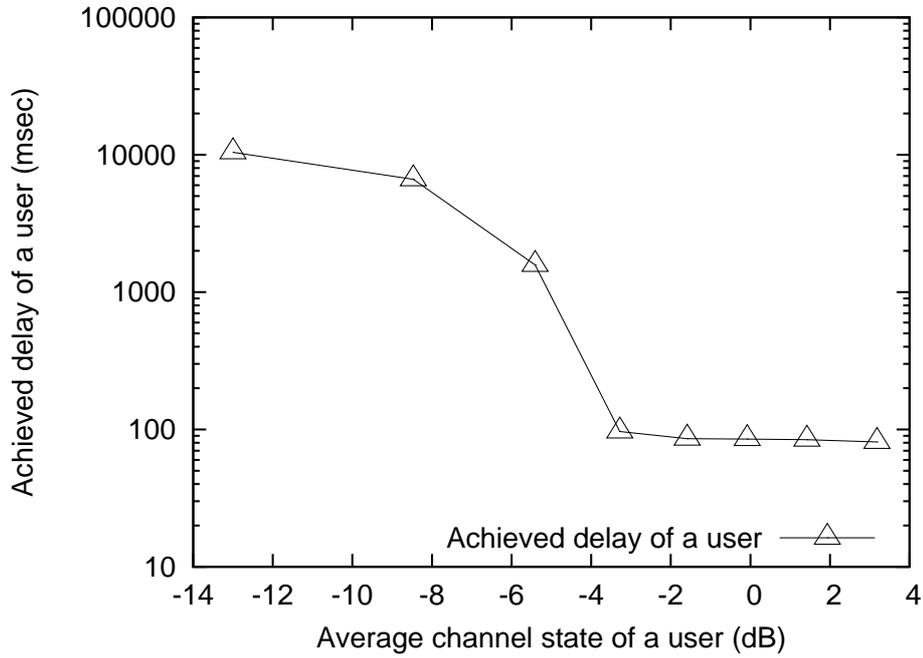


Figure 5.8: Delay experienced by a user selected at random for various average channel states - symmetric case

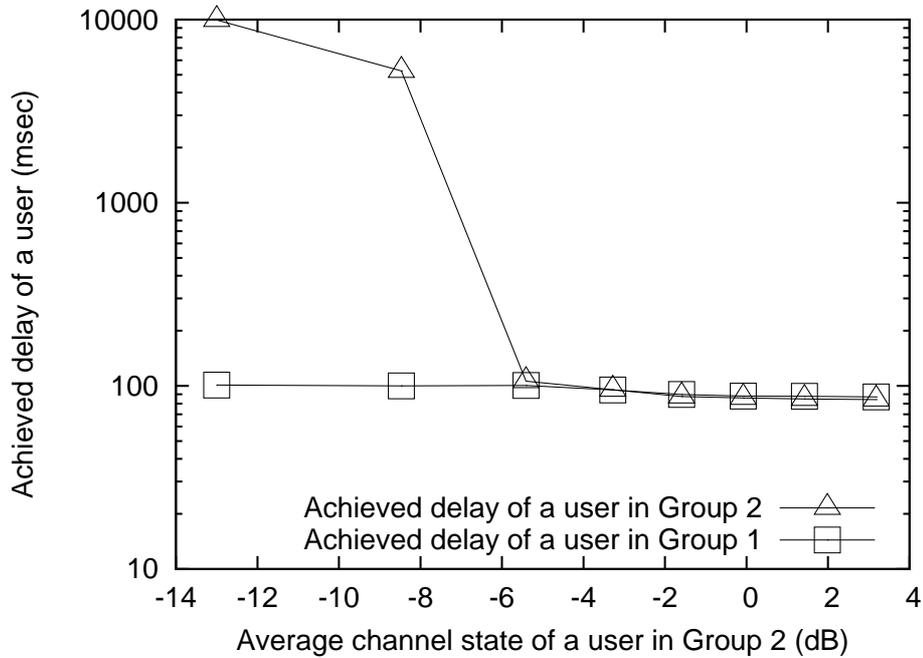


Figure 5.9: Delay experienced by two users selected at random from Group 1 and Group 2 for various average channel states - asymmetric case

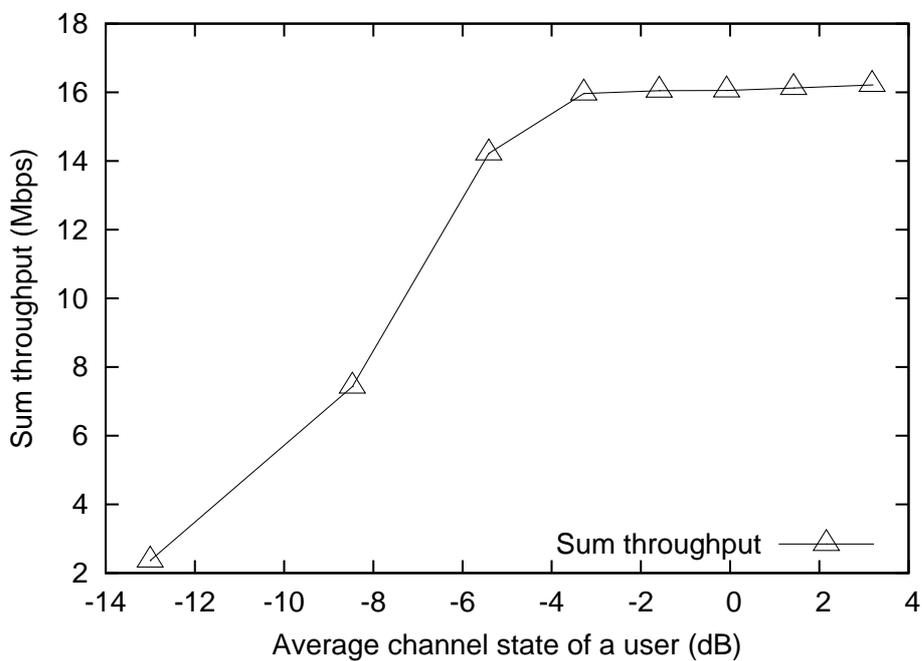


Figure 5.10: Sum throughput for various average channel states - symmetric case

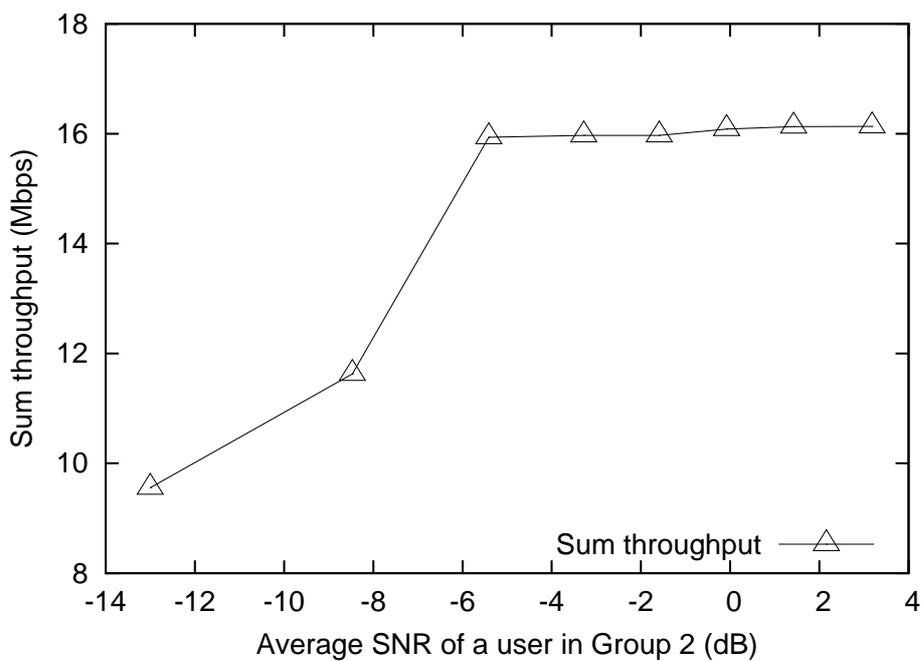


Figure 5.11: Sum throughput for various average channel states - asymmetric case

Simulation Parameter	Value
Mean arrival rate (Group 1&2) symmetric case	0.4246 to 0.9650 Mbps/user in 8 steps
Mean arrival rate (Group 1) asymmetric case	0.6562 Mbps/user
Mean arrival rate (Group 2) asymmetric case	0.4246 to 0.9650 Mbps/user in 8 steps
Mean channel state (Group 1&2) both cases	-3.28 dB
Delay constraint (Group 1&2) both cases	100 msec

Table 5.5: Summary of parameters for Scenario 5.4

this scenario have been summarized in Table 5.5. The average delays experienced by two specific users (each selected at random from Group 1 and Group 2) for the IS are plotted in Figure 5.13. It can be seen from Figures 5.12 and 5.13 that the delay constraints are satisfied till the arrival rate of about 0.75 Mbps/user for the symmetric case and about 0.9 Mbps/user for the asymmetric case. The sum throughput achieved for both cases is plotted in Figures 5.14 and 5.15 respectively.

## 5.5 Conclusions

In this chapter, we have investigated the problem of scheduling users on the downlink of a TDM system with constraints on the average packet delays over a fading wireless channel. We have suggested an indexing scheme, IS, that is easy to implement in practice. The IS generates indices in each time slot and the user with the maximum index is scheduled. Our simulations for the IEEE 802.16 system have indicated that the delay constraints of the users are satisfied. Since this problem has not been studied previously, for comparison purposes, we have adapted the M-LWDF scheme to our scenario. The IS is more suited to the problem studied in this chapter (where the QoS requirements mandate maintaining the average delay less than or equal to a prescribed bound) as compared to the M-LWDF scheme. Moreover, the comparisons indicate that the IS is highly throughput efficient.

We have investigated the centralized scheduling issues on the uplink and downlink in Chapters 4 and 5 respectively. In the next chapter, we focus on distributed scheduling aspects. In this case, scheduling takes the form of distributed channel access.

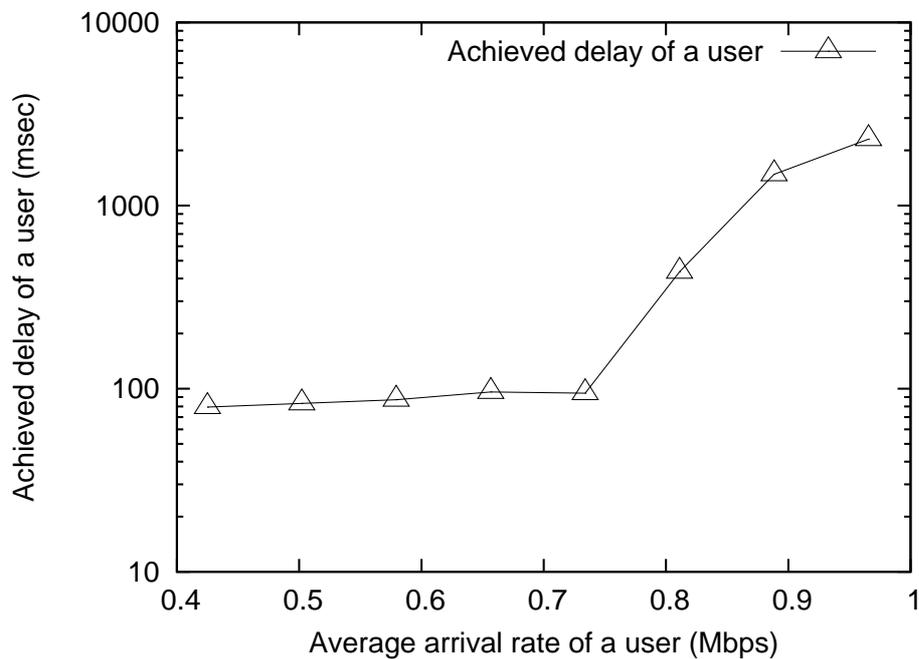


Figure 5.12: Delay experienced by a user selected at random for various average arrival rates - symmetric case

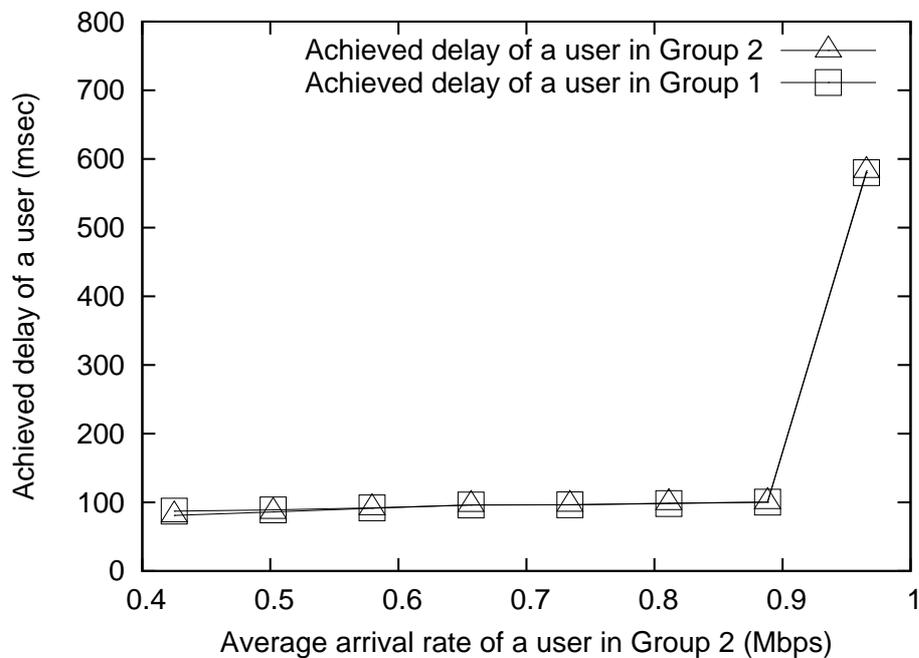


Figure 5.13: Delay experienced by two users selected at random from Group 1 and Group 2 for various average arrival rates - asymmetric case

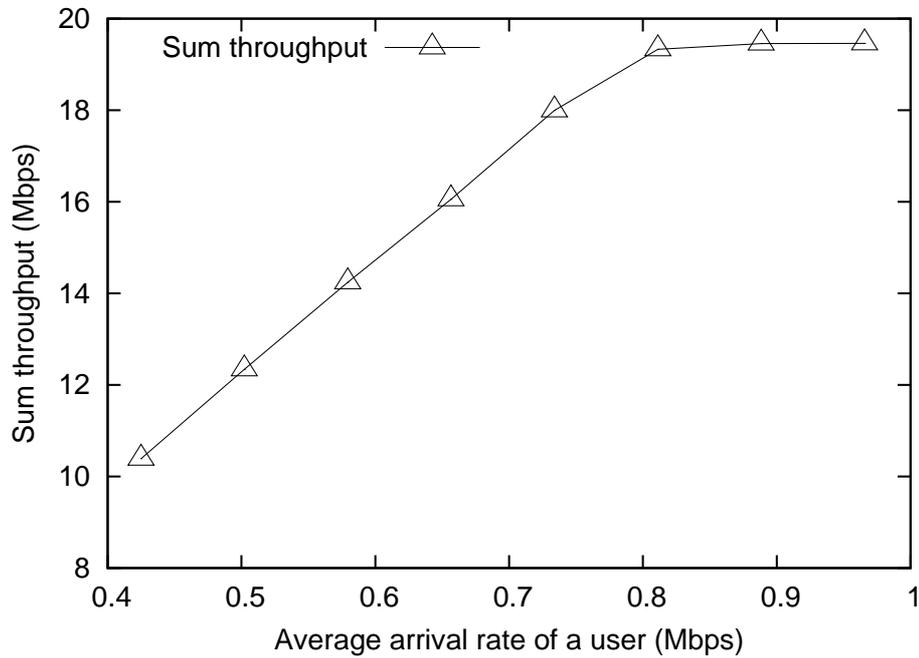


Figure 5.14: Sum throughput for various average arrival rates - symmetric case

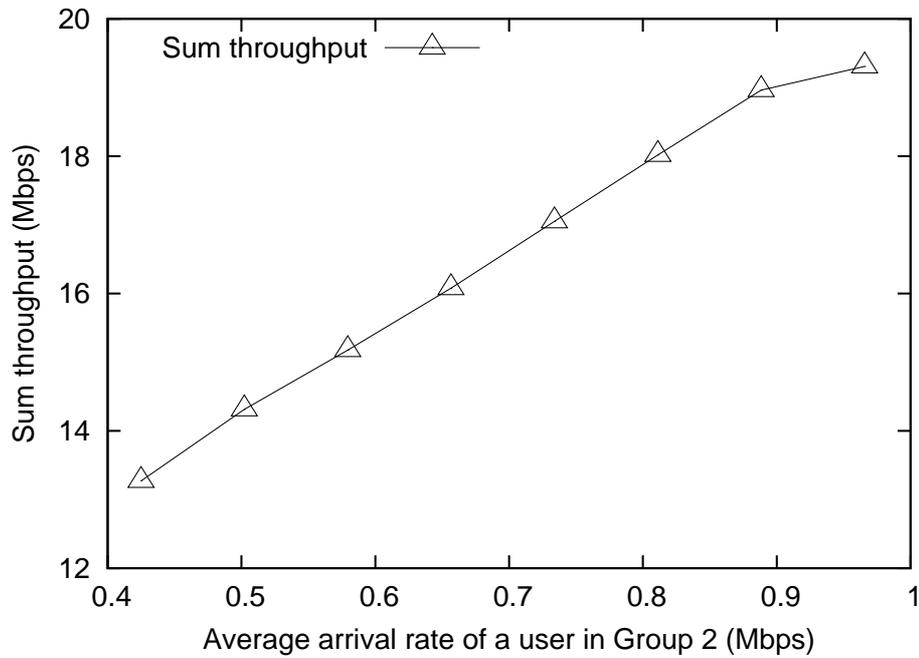


Figure 5.15: Sum throughput for various average arrival rates - asymmetric case

# Chapter 6

## Energy Efficient Scheduling for Multiuser Distributed Channel Access

### 6.1 Introduction

This chapter focuses on the problem of distributed multipoint-to-multipoint scheduling or access control in a single hop. The general problem of providing end-to-end QoS in a multihop wireless network involves other issues such as routing. However, to focus on the scheduling or access control aspect, we consider a somewhat restricted scenario where users located in a vicinity are divided into source-destination pairs. All nodes are in the transmission range of each other, hence only one source can transmit in a time slot. Each source transmits at a constant power. Rate adaptation is performed based on the channel state perceived by the associated destination. Since each source transmits at a constant power, the average power consumption is synonymous with the average channel access rate or transmission probability. The sources attempt to regulate their transmissions in such a fashion that the specified delay constraints are satisfied. Since each channel access consumes power, the problem is to determine the channel access rate or steady state transmission probability that is just sufficient for satisfying the specified delay constraints. Our objective is to design distributed scheduling with no information exchange between the sources by employing random access communication.

Random access communication is a well studied topic [10, 100]. In recent years, the area of random access communication has attracted renewed research interest because of ad hoc networks [65]. A well studied problem in this area is that of determining efficient channel access schemes that attain a high saturation throughput. Our objective, in this chapter, is to study an orthogonal problem where the focus is to determine the channel access rate (and thereby throughput) for a source such that it is just sufficient for satisfying the QoS (delay) constraint. Moreover, we consider variable rate transmission at the source, i.e., we assume that a source transmits at a rate depending on the channel condition perceived by the associated destination. Furthermore, we assume that the probability distributions of the arrival and channel fading processes corresponding to each source and destination are not known.

We formulate the problem as a multistage constrained optimization problem. Our solution approach is to iteratively tune the source channel access rate or transmission probability such that it is just sufficient to satisfy the delay constraints. Since it is difficult to obtain an optimal solution, we suggest two sub-optimal approaches. The first is based on the stochastic gradient approach, and is referred to as Stochastic Gradient Algorithm (SGA) in the chapter. In this approach, each source treats the problem of determining the steady state transmission probability as an independent optimization problem. We suggest an iterative primal dual algorithm that determines a locally optimal transmission probability for each source. However, this algorithm is a three timescale stochastic approximation and hence the convergence of this algorithm is expected to be slow. In order to speed up the convergence, we then motivate a single time scale stochastic approximation algorithm (STSAA). We prove that STSAA converges to an equilibrium and that it satisfies the delay constraints. This is also validated through simulations.

The rest of the chapter is organized as follows. Section 6.2 provides details regarding the system model. We formulate the problem as an optimization problem in Section 6.3. In Section 6.4, we propose SGA. In Section 6.5, we propose STSAA and prove that the algorithm converges to an equilibrium. We present the simulation results in Section 6.6 and conclude in Section 6.7.

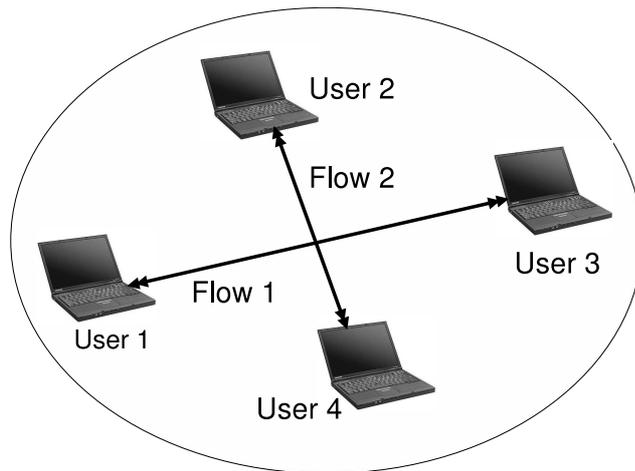


Figure 6.1: Distributed transmission scenario

## 6.2 System Model

We consider a scenario depicted in Figure 6.1 where  $N$  nodes are located in a geographical area. We number the nodes as  $1, \dots, N$ . All nodes are in the transmission range of each other and hence only one source can transmit at any instant of time. We consider a time slotted system, i.e., time is divided into slots of equal duration normalized to unity. There can be multiple flows between a source and destination. Moreover, multiple sources can communicate with a single destination. However, for the sake of notational simplicity, we assume that only one source communicates with a specific destination and a single flow between a source and destination pair. The analysis can be easily extended to more general cases. User equipment is assumed to have a simplex structure with carrier sensing, i.e., it can either transmit or receive at an instant of time and when it is not transmitting, it can detect whether there is a transmission in the slot.

We assume that the system operates in a *distributed* fashion, i.e., there does not exist a central entity that coordinates the transmissions of the sources. In each slot  $n$ , the source  $i$  transmits with a certain probability  $\theta_n^i$  to its associated destination  $j$ . If more than one source transmits in a slot, then all transmissions are unsuccessful, i.e., there is a *collision*.

We assume that each source receives a  $(0, 1, e)$  feedback in each slot, where 0 denotes that there is no transmission in the slot, 1 denotes successful transmission and  $e$  denotes collision or unsuccessful transmission. We assume that this feedback is immediate and

error free.

We assume a wireless channel with block fading [27]. We have already introduced the block fading model in Chapter 2, here we revisit it for the distributed communication scenario. Under this model, if  $\chi_n^i$  is the transmitted signal by source  $i$  in slot  $n$ , then the signal  $Y_n^j$  received by destination  $j$  in slot  $n$  can be expressed as:

$$Y_n^j = H_n^j \chi_n^i + Z_n^j, \quad (6.1)$$

where  $Z_n^j$  is the complex Additive White Gaussian Noise (AWGN) at destination  $j$ .  $H_n^j$  is the channel gain and  $X_n^j = |H_n^j|^2 \in \mathbb{X}$  is the channel state for destination  $j$  in slot  $n$ . We assume that the destination estimates the channel state  $X_n^j$  and informs this to the source instantaneously. Moreover, the distribution of  $X_n^j$  is not known to source  $i$ .

Let  $A_n^i$  be the number of packets arriving into the queue of source  $i$  in slot  $n$ . All packets are of equal size  $\ell$  bits. We assume that  $\{A_n^i\}$  is an i.i.d. process  $\forall i$ . Moreover, the distribution of the random variable  $A_n^i$  is not known to the source  $i$ . Packets are stored in the source queue until they are transmitted. Let  $Q_n^i$  be the number of packets in the source  $i$  queue in slot  $n$ . We assume that the each source has a finite but large buffer of size  $B$  packets. Hence, the random variable  $Q_n^i$  takes values from a discrete and finite set  $\mathbb{Q} \triangleq \{0, \dots, B\}$ . We assume that in each slot, the sources transmit with a constant maximum power  $P_m$ . Let  $U_n^i$  denote the number of packets that source  $i$  can transmit *reliably* to destination  $j$  in slot  $n$ . In practice, this can be determined based on the modulation or coding scheme employed at the physical layer. Moreover,  $U_n^i \leq Q_n^i$ .

The queue evolution equation for source  $i$  can be expressed as<sup>1</sup>:

$$Q_{n+1}^i = Q_n^i + A_{n+1}^i - I_n^i U_n^i, \quad (6.2)$$

where  $I_n^i$  is an indicator variable that is set to 1 if source  $i$  successfully transmits to destination  $j$  in slot  $n$ . This implies that no other source transmits in that slot. By Little's law (2.36), one can treat the average delay as synonymous with the average queue length. Hence, as is considered throughout the thesis, we consider the average queue length constraints instead of the average delay constraints.

---

<sup>1</sup>Following the discussion in Chapter 3, we assume that, as compared to the packet arrival rate (and the average queue length/delay constraint),  $B$  is large enough so as to neglect the buffer overflow and hence packet drops.

## 6.3 Problem Formulation

Since the sources transmit at constant power, the average power consumed is synonymous with the average access rate, i.e., ratio of the total number of transmission attempts to the total number of slots, or the transmission probability in each slot. Hence, our attempt is to design a random access mechanism under which a source transmits with as low a transmission probability as is just sufficient for satisfying the delay constraint.

The long term transmission probability for source  $i$  can be expressed as:

$$\bar{\theta}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \theta_n^i. \quad (6.3)$$

The average queue length at source  $i$  over a long period of time can be expressed as:

$$\bar{Q}^i = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M Q_n^i. \quad (6.4)$$

It is required that this average queue length be below a certain desired queue length, say,  $\bar{\delta}^i$ . Hence, the optimization problem can be expressed as:

$$\text{Minimize } \bar{\theta}^i \text{ subject to } \bar{Q}^i \leq \bar{\delta}^i. \quad (6.5)$$

Note that any transmission probability that satisfies the delay constraints with inequality is sub-optimal because by reducing the transmission probability, the delay constraints can be satisfied with equality, thereby saving power. We denote the vector  $\boldsymbol{\theta}^* = [\theta^{1,*}, \dots, \theta^{N,*}]^T$  as the equilibrium or steady state transmission probability vector. This vector is the minimum transmission probability vector in the following sense; there exists no vector  $\boldsymbol{\theta}^\#$  such that  $\boldsymbol{\theta}^\# \leq \boldsymbol{\theta}^* \wedge \exists i$  such that  $\theta^{i,\#} < \theta^{i,*}$  and if the sources transmit with transmission probabilities from  $\boldsymbol{\theta}^\#$ , the delay constraints of all users are satisfied.

## 6.4 An Algorithm based on the Stochastic Gradient Approach

In this method, each source treats the problem of determining the steady state transmission probability as an independent optimization problem. We present an algorithm based on the stochastic gradient approach for solving this optimization problem at a given

source  $i$ . We first convert the constrained problem in (6.5) into an unconstrained problem using the Lagrangian approach. We then describe an iterative primal dual algorithm to determine the locally optimal primal (transmission probability) and dual.

### 6.4.1 Lagrangian Approach

Let  $\lambda^i > 0$  be a real number termed as the Lagrange Multiplier (LM). The Lagrangian can be expressed as:

$$L^i(\theta^i, \lambda^i) = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M (\theta_n^i + \lambda^i(Q_n^i - \bar{\delta}^i)). \quad (6.6)$$

The objective is to determine the saddle point of the Lagrangian, i.e., to determine  $\theta^{i,*}$  and  $\lambda^{i,*}$  such that the following saddle point optimality conditions are satisfied:

$$L^i(\theta^{i,*}, \lambda^i) \leq L^i(\theta^{i,*}, \lambda^{i,*}) \leq L^i(\theta^i, \lambda^{i,*}). \quad (6.7)$$

Let  $\nabla^{\theta^i} L^i(\cdot, \cdot)$  and  $\nabla^{\lambda^i} L^i(\cdot, \cdot)$  denote the partial gradient of  $L^i(\cdot, \cdot)$  w.r.t  $\theta^i$  and  $\lambda^i$  respectively. At the saddle point,  $\theta^{i,*}$  and  $\lambda^{i,*}$  satisfy the following conditions:

$$\nabla^{\theta^i} L^i(\theta^i, \lambda^i) \Big|_{\theta^i = \theta^{i,*}} = 0, \quad (6.8)$$

$$\nabla^{\lambda^i} L^i(\theta^i, \lambda) \Big|_{\lambda^i = \lambda^{i,*}} = 0, \quad (6.9)$$

and the complementary slackness condition,

$$\lambda^{i,*}(\bar{Q}^i - \bar{\delta}^i) = 0. \quad (6.10)$$

Note that the Lagrangian in (6.6) is a time averaged function that can not be determined a priori in an online implementation setup, hence, we can not determine its gradient. If the optimal LM  $\lambda^{i,*}$  is known, we can use an iterative method that improve its estimate of *only* the optimal  $\theta^i$ . Since the optimal LM  $\lambda^{i,*}$  is also not known, we resort to a primal-dual method that determines *both*  $\theta^{i,*}$  and  $\lambda^{i,*}$  iteratively [40]. In order to ensure convergence of  $\theta^i$  and  $\lambda^i$  iterates to the optimal  $\theta^{i,*}$  and  $\lambda^{i,*}$ , the iterations proceed at different *timescales*, i.e., the  $\theta^i$  and  $\lambda^i$  values are updated at different rates [91]. Let  $\{b_n\}$  and  $\{c_n\}$  be two positive sequences. The coupled iterative equations can be expressed as:

$$\theta_{n+1}^i = \pi_1 \left[ \theta_n^i - b_n \nabla^{\theta^i} L^i(\theta_n^i, \lambda_n^i) \right], \quad (6.11)$$

and

$$\lambda_{n+1}^i = \pi_2 \left[ \lambda_n^i + c_n (Q_n^i - \bar{\delta}^i) \right], \quad (6.12)$$

where  $\pi_1(\cdot)$  is a projection function that projects the  $\theta_n^i$  iterates in the interval  $[0, \omega]$ ,  $\omega \leq 1$ .  $\omega$  is a configuration parameter that limits the transmission probabilities of the sources. This ensures that the system does not become unstable because of too many collisions.  $\pi_2(\cdot)$  is a projection function that projects the  $\lambda_n^i$  iterates in the interval  $[0, \Gamma]$ ,  $\Gamma \gg 0$  in order to ensure their boundedness.

In a time slot  $n$ , define the *immediate* cost function  $g^i(\theta_n^i, \lambda_n^i)$  as:

$$g^i(\theta_n^i, \lambda_n^i) \triangleq \theta_n^i + \lambda_n^i (Q_n^i - \bar{\delta}^i). \quad (6.13)$$

It should be noted that the Lagrangian in (6.6) can be expressed in terms of the immediate cost function as:

$$L^i(\theta^i, \lambda^i) = \limsup_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M g^i(\theta_n^i, \lambda_n^i). \quad (6.14)$$

The partial gradient of the Lagrangian w.r.t  $\theta^i$  is not known. Hence, we use the stochastic gradient approach [26] in order to approximate the gradient of  $L^i(\theta^i, \lambda^i)$ .

### 6.4.2 Stochastic Gradient Approach

From (6.14), we know that  $L^i(\cdot, \cdot)$  is a time average of the instantaneous cost function  $g^i(\cdot, \cdot)$ . Hence, in order to determine the gradient  $\nabla^{\theta^i} L^i(\cdot, \cdot)$  of  $L^i(\cdot, \cdot)$  we perform time averaging of  $\nabla^{\theta^i} g^i(\cdot, \cdot)$ . Note that  $\theta^i$  and  $\lambda^i$  are not independent. Increase in  $\theta^i$  leads to a decrease in the queue length and hence leads to a decrease in  $\lambda^i$ , while decrease in  $\theta^i$  leads to an increase in  $\lambda^i$ . Exact relationship between  $\theta^i$  and  $\lambda^i$  is difficult to determine. We assume that a perturbation of  $\varepsilon_1 > 0$  in  $\theta^i$  leads to a perturbation of  $-\varepsilon_2$  ( $\varepsilon_2 > 0$ ) in  $\lambda^i$ . Define  $j_{n+1}^{i,1}(\theta_n^i, \lambda_n^i)$  and  $j_n^{i,2}(\theta_n^i, \lambda_n^i)$  to be the two averaged perturbations of  $g^i(\theta_n^i, \lambda_n^i)$ . These can be expressed as:

$$j_{n+1}^{i,1}(\theta_n^i, \lambda_n^i) = (1 - e_n) j_n^{i,1}(\theta_n^i, \lambda_n^i) + e_n \left[ g^i(\theta_n^i + \varepsilon_1, \lambda_n^i - \varepsilon_2) \right], \quad (6.15)$$

and

$$j_{n+1}^{i,2}(\theta_n^i, \lambda_n^i) = (1 - e_n) j_n^{i,2}(\theta_n^i, \lambda_n^i) + e_n \left[ g^i(\theta_n^i - \varepsilon_1, \lambda_n^i + \varepsilon_2) \right], \quad (6.16)$$

where  $\{e_n\}$  is a positive sequence and  $\varepsilon_1, \varepsilon_2 \in (0, 1]$ . Let  $\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i)$  denote the estimated gradient of the Lagrangian evaluated at  $(\theta_n^i, \lambda_n^i)$  in a slot  $n$ . It can be expressed as:

$$\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) = \frac{j_n^{i,1}(\theta_n^i, \lambda_n^i) - j_n^{i,2}(\theta_n^i, \lambda_n^i)}{2\varepsilon_2}. \quad (6.17)$$

We impose the following properties on the sequences  $\{e_n\}$ ,  $\{b_n\}$  and  $\{c_n\}$ .

$$e_n \rightarrow 0, b_n \rightarrow 0, c_n \rightarrow 0; \quad \sum_n e_n = \sum_n b_n = \sum_n c_n = \infty; \quad (6.18)$$

$$\sum_n (e_n^2 + b_n^2 + c_n^2) < \infty; \quad \frac{b_n}{e_n}, \frac{c_n}{b_n} \rightarrow 0. \quad (6.19)$$

These properties ensure that the sequences  $\{e_n\}$ ,  $\{b_n\}$  and  $\{c_n\}$  converge to 0 neither too rapidly nor too slowly. Moreover, the last property in (6.19) ensures that the update rates of  $j_n^{i,1}(\cdot, \cdot)$ ,  $j_n^{i,2}(\cdot, \cdot)$ ,  $\theta_n^i$  and  $\lambda_n^i$  iterates are different. We update  $j^{i,1}$  and  $j^{i,2}$  at the fastest timescale. The  $\theta^i$  iterations are at a slower timescale. This ensures that the  $\theta^i$  iterations see a converged gradient, and hence the probabilities are properly updated. We update  $\lambda^i$  at the slowest timescale. From (6.11), the primal iteration can be represented as:

$$\theta_{n+1}^i = \pi_1[\theta_n^i - b_n(\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i))]. \quad (6.20)$$

From (6.12), (6.15), (6.16) and (6.20), SGA can be expressed as:

$$j_{n+1}^{i,1}(\theta_n^i, \lambda_n^i) = (1 - e_n)j_n^{i,1}(\theta_n^i, \lambda_n^i) + e_n \left[ g^i(\theta_n^i + \varepsilon_1, \lambda_n^i - \varepsilon_2) \right], \quad (6.21)$$

$$j_{n+1}^{i,2}(\theta_n^i, \lambda_n^i) = (1 - e_n)j_n^{i,2}(\theta_n^i, \lambda_n^i) + e_n \left[ g^i(\theta_n^i - \varepsilon_1, \lambda_n^i + \varepsilon_2) \right], \quad (6.22)$$

$$\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) = \frac{j_n^{i,1}(\theta_n^i, \lambda_n^i) - j_n^{i,2}(\theta_n^i, \lambda_n^i)}{2\varepsilon_2}, \quad (6.23)$$

$$\theta_{n+1}^i = \pi_1[\theta_n^i - b_n(\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i))], \quad (6.24)$$

$$\lambda_{n+1}^i = \pi_2 \left[ \lambda_n^i + c_n(Q_n^i - \bar{\delta}^i) \right]. \quad (6.25)$$

### 6.4.3 Implementation Details

Based on online primal-dual computations in (6.21), (6.22), (6.23) (6.24) and (6.25), source  $i$  implements the scheduling scheme. We assume that the source is aware of the value of channel state  $X^j$  in each slot. In practice, this may be achieved by the destination first estimating the channel state and then informing this to the transmitter through a feedback mechanism. In each slot, the transmitter has the information regarding the number of

packet arrivals and current queue length. The number of packets to be transmitted is then determined depending on the channel state. The partial gradients  $j^{i,1}(\cdot, \cdot)$  and  $j^{i,2}(\cdot, \cdot)$  are first updated. Then the estimated gradient  $\hat{\nabla}^{\theta^i}(\cdot, \cdot)$  is updated. This is then used to determine the transmission probability  $\theta^i$  and the source transmits with this probability. If the transmission is successful,  $U^i$  packets are transmitted and the LM  $\lambda^i$  is appropriately updated. The algorithm thus continues in each slot. The complete scheme is explained in Algorithm 5.

- 1: Initialize the partial gradients  $j_0^{i,1}(\cdot, \cdot) \leftarrow 0$ ,  $j_0^{i,2}(\cdot, \cdot) \leftarrow 0$ , the LM  $\lambda_0^i \leftarrow 0$ , estimated gradient  $\hat{\nabla}_0^{\theta^i}(\theta_0^i, \lambda_0^i) \leftarrow 0$ ,  $\theta_0^i \leftarrow \theta_0$ ,  $n \leftarrow 1$ , queue length  $Q_0^i \leftarrow 0$ , channel state  $X_0^i \leftarrow 0$ .
- 2: **while** TRUE **do**
- 3:   Determine number of arrivals  $A_{n+1}^i$  and channel state  $X_n^i$  in the current slot.
- 4:   Determine  $U_n^i$ .
- 5:   Update  $j_n^{i,1}(\cdot, \cdot)$ ,  $j_n^{i,2}(\cdot, \cdot)$  and estimated gradient  $\hat{\nabla}_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i)$  using (6.21), (6.22), (6.23) respectively.
- 6:   Update transmission probability  $\theta_n^i$  using (6.24).
- 7:   Transmit with probability  $\theta_n^i$  with rate  $U_n^i$ .
- 8:   **if** Transmission is successful **then**
- 9:      $I_n^i \leftarrow 1$ .
- 10:   **else**
- 11:      $I_n^i \leftarrow 0$ .
- 12:   **end if**
- 13:   Update the LM  $\lambda_n^i$  using (6.25).
- 14:   Update queue length using (6.2).
- 15:    $n \leftarrow n + 1$ .
- 16: **end while**

**Algorithm 5:** Stochastic Gradient Algorithm (SGA)

#### 6.4.4 Limitations of SGA

SGA is a three timescale algorithm. Due to this, the convergence of the transmission probability  $\theta_n^i$  and the LM  $\lambda_n^i$  to stable values is expected to be slow. Moreover, the algo-

rithm is difficult to analyze. Hence, we propose another sub-optimal algorithm, STSAA in the next section. The simple structure of STSAA allows us to analyze it and prove certain properties regarding its asymptotic behavior. Moreover, it has computational advantages over SGA.

## 6.5 A Single Timescale Stochastic Approximation Algorithm (STSAA)

Consider the immediate cost function  $g^i(\cdot, \cdot)$ . Using the gradient  $\nabla_n^{\theta^i} g^i(\theta_n^i, \lambda_n^i)$  of  $g^i(\cdot, \cdot)$  as an approximation to the gradient  $\nabla_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i)$  of  $L^i(\cdot, \cdot)$ , we can write the  $\theta^i$  iteration (6.24) as:

$$\theta_{n+1}^i = \pi_1 \left[ \theta_n^i - b_n \left( \nabla_n^{\theta^i} g^i(\theta_n^i, \lambda_n^i) \right) \right]. \quad (6.26)$$

Note that (6.26) can be expressed as:

$$\theta_{n+1}^i = \pi_1 \left[ \theta_n^i - b_n \left( \nabla_n^{\theta^i} g^i(\theta_n^i, \lambda_n^i) - \nabla_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) + \nabla_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) \right) \right]. \quad (6.27)$$

Now, the quantity  $\left( \nabla_n^{\theta^i} g^i(\theta_n^i, \lambda_n^i) - \nabla_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) \right)$  is a zero mean random variable. Hence asymptotically, (6.26) can be assumed to have a behavior corresponding to the algorithm,

$$\theta_{n+1}^i = \pi_1 \left[ \theta_n^i - b_n \left( \varphi^i + \nabla_n^{\theta^i} L^i(\theta_n^i, \lambda_n^i) \right) \right], \quad (6.28)$$

where  $\varphi^i$  is a zero mean random variable. (6.28) is the well known stochastic gradient algorithm where one attempts to iteratively tune a parameter based on noisy measurements of the gradient of its function [26]. Let  $\varepsilon = \varepsilon_1 \varepsilon_2$ . Then,  $\frac{\partial \lambda_n^i}{\partial \theta_n^i} = -\varepsilon$ , by arguments of Section 6.4.2. Substituting  $\nabla_n^{\theta^i} g^i(\theta_n^i, \lambda_n^i) = (1 - \varepsilon(Q_n^i - \bar{\delta}^i))$ , (6.26) can be expressed as:

$$\theta_{n+1}^i = \pi_1 \left[ \theta_n^i - b_n (1 - \varepsilon(Q_n^i - \bar{\delta}^i)) \right]. \quad (6.29)$$

Taking cue from (6.29), we propose the following stochastic ‘gradient like’ iterative algorithm (STSAA) for determining the equilibrium transmission probability for source  $i$ :

$$\theta_{n+1}^i = \pi \left[ \theta_n^i + f_n \times (Q_n^i - \bar{\delta}^i) \right], \quad (6.30)$$

where  $\pi(\cdot)$  projects the  $\theta_n$  iterates in the interval  $[0, \omega]$ ,  $\omega \in [0, 1]$  being a configurable parameter as discussed previously. The step size sequence  $\{f_n\}$  has the following properties:

$$\sum_n f_n = \infty, \quad \sum_n (f_n)^2 < \infty. \quad (6.31)$$

As discussed previously, these properties of  $f_n$  govern the convergence of the  $\theta_n$  iterates to their steady state values and ensure that  $f_n$  converges to zero neither too rapidly nor too slowly. Note that one can view (6.30) as iteratively tuning the probability of transmission such that it is just sufficient to satisfy the delay constraints. If  $Q_n^i > \bar{\delta}^i$ , the probability of transmission is increased, on the other hand, if  $Q_n^i < \bar{\delta}^i$ , the probability of transmission is reduced. Moreover, this algorithm does not require any information regarding the probability distribution of the arrival process or the channel fading process. Furthermore, there is no information exchange between the sources regarding their transmission policies. We assume that the delay constraints are feasible, i.e., there exists a transmission probability vector  $\theta'$  such that if the transmission probabilities are fixed at  $\theta'$ , the delay constraints are satisfied.

### 6.5.1 Implementation Details

Based on (6.30), source  $i$  implements STSAA. We assume that the source is aware of the value of channel state  $X^j$  in each slot. In each slot, the source determines the number of packet arrivals and current queue length. The channel state is then used to determine the number of packets to be transmitted. This is then used to determine the transmission probability  $\theta^i$ , and the source transmits with this probability. If the transmission is successful,  $U^i$  packets are transmitted. The algorithm thus continues in each slot. The complete algorithm is explained in Algorithm 6.

### 6.5.2 Convergence Analysis

We now prove that the algorithm (6.30) indeed determines the equilibrium transmission probability  $\theta^{i,*}$ . Let  $\theta_n = [\theta_n^1, \dots, \theta_n^N]^T$  denote the transmission probability vector in slot  $n$ . Let  $\bar{\mathbf{Q}} = [\bar{Q}^1, \dots, \bar{Q}^N]^T$  denote the vector of average queue lengths achieved by nodes

```

1: Initialize  $\theta_0^i \leftarrow \theta_0$ ,  $n \leftarrow 1$ , queue length  $Q_0^i \leftarrow 0$ , channel state  $X_0^i \leftarrow 0$ .
2: while TRUE do
3:   Determine number of arrivals  $A_{n+1}^i$  and channel state  $X_n^i$  in the current slot.
4:   Determine  $U_n^i$ .
5:   Update transmission probability  $\theta_n^i$  using (6.30).
6:   Transmit with probability  $\theta_n^i$  with rate  $U_n^i$ .
7:   if Transmission is successful then
8:      $I_n^i \leftarrow 1$ .
9:   else
10:     $I_n^i \leftarrow 0$ .
11:  end if
12:  Update queue length using (6.2).
13:   $n \leftarrow n + 1$ .
14: end while

```

**Algorithm 6:** Single Timescale Stochastic Approximation Algorithm (STSAA)

$1, \dots, N$ . (6.30) can be represented as:

$$\theta_{n+1}^i = \theta_n^i + f_n \times (Q_n^i - \bar{\delta}^i) + \eta_n^i, \quad (6.32)$$

where  $\eta_n^i$  is a correction term introduced to account for the projection operation  $\pi(\cdot)$ . We first analyze the following iterative equation:

$$\theta_{n+1}^i = \theta_n^i + f_n \times (Q_n^i - \bar{\delta}^i), \quad (6.33)$$

From (6.33), note that the probability of transmission for source  $i$  in slot  $n+1$  is dependent on the probability of transmission in slot  $n$  and the deviation of the current queue length from the queue length constraint. The deviation of the current queue length from the constraint can be considered as a ‘noise’ using the theory of stochastic approximation [26, 25]. Note that the average queue length  $\bar{Q}^i$  achieved by source  $i$  is dependent on the transmission probability of source  $i$  as well as the transmission probabilities of all other sources  $m$ ,  $m \neq i, j$ ,  $m = 1, \dots, N$ , i.e., the entire  $\boldsymbol{\theta}$  vector. By the o.d.e. method of analysis for the stochastic approximation algorithm, we can consider the limiting o.d.e. [25, 90],

$$\dot{\theta}^i(t) = \left( \bar{Q}^i(\boldsymbol{\theta}(t)) - \bar{\delta}^i \right), \quad 1 \leq i \leq N, \quad (6.34)$$

where  $\bar{Q}^i(\boldsymbol{\theta})$  is the stationary average queue length achieved by source  $i$  if the transmission probabilities are fixed at  $\boldsymbol{\theta} = [\theta^1, \dots, \theta^N]^T$ . Note that  $Q_n^i$  takes values from the finite set  $\mathbb{Q}$ , ensuring that  $\bar{Q}^i(\cdot)$  is bounded.

We make the following assumption:

*Assumption 6.1.*  $\bar{Q}^i(\boldsymbol{\theta})$  is a continuously differentiable function of  $\boldsymbol{\theta}$ .

Note that  $\bar{Q}^i(\boldsymbol{\theta})$  is increasing in  $\theta^k$ ,  $k \neq i, j$ . This is because, increase in the transmission probability of source  $k$  leads to a higher collision probability for source  $i$ , and hence for a constant transmission probability  $\theta^i$  for source  $i$ , its achieved average queue length  $\bar{Q}^i(\cdot)$  increases. With this consideration and under Assumption 6.1, (6.34) is a cooperative o.d.e. in the sense of [101], a special case of [102]. From results of [101], we have,

**Theorem 6.1.** *For initial conditions belonging to an open dense set,  $\boldsymbol{\theta}(\cdot)$  converges to the set  $H$  of equilibria of (6.34).*

Now consider the term  $\eta_n^i$  in (6.30). If the iterates converge to the set  $[0, \omega)$ , the correction term becomes asymptotically negligible and hence can be neglected.

*Remark 6.1.* Note that, (6.34) is stable only if  $\bar{Q}^i(\boldsymbol{\theta}(t)) = \bar{\delta}^i$ . This implies that the convergence of the o.d.e. to equilibrium results in the delay constraints being satisfied with equality. Hence, transmitting at the stable transmission probability  $\theta^{i,*}$  ensures that the delay constraints are satisfied with equality. Note that the equilibrium transmission probability is dependent on the ‘resource requirement’ by a source-destination pair. This implies that larger arrival rate, lower delay constraints and poorer channel conditions result in higher transmission probability. If the system does not have enough capacity to support the requirements of the users, the collision rate will become quite high and result in instability. To prevent instability, we introduce the parameter  $\omega$  and project the  $\theta_n^i$  iterates in the interval  $[0, \omega]$ . Moreover, admission control can ensure that the system admits users only if there is sufficient capacity for satisfying the user requirements and thereby avoid unstable behavior.

In the next section, we simulate SGA and STSAA. Our objective is to validate the analytical results obtained in this section using simulation studies.

## 6.6 Simulation Results

In this section, we demonstrate that SGA and STSAA satisfy the delay constraints. We consider a time slotted system with a slot duration of 1 msec. The arrival process for each source is Poisson. Each packet is of constant length equal to  $\ell = 2000$  bits. We assume that the system has a bandwidth  $W$  of 10 MHz. Each source transmits at a constant power of 1 Watt. We assume  $\omega = 0.1$ . We simulate a Rayleigh channel for each user. For a Rayleigh model, channel state  $X^i$  is an exponentially distributed random variable with mean  $\alpha^i$  (probability density function expressed in (2.7)). We discretize the channel into eight equal probability bins using the discretization procedure specified in Section 3.6. For the sake of simplicity, we assume that the rate of transmission for source  $i$  in slot  $n$  can be determined using the following capacity relation:

$$U_n^i = W \times \log_2\left(1 + \frac{P_m X_n^j}{N_0 W}\right). \quad (6.35)$$

We consider a system with  $N = 20$  source destination pairs. We divide the source destination pairs into 2 groups (Group 1 and Group 2) of 10 pairs each. We consider three scenarios: delay variation, arrival variation and channel variation. We present the results after averaging over 20 simulation runs each consisting of simulating the algorithm for 100,000 slots. The parameters common for all scenarios have been summarized in Table 6.1.

*Scenario 6.1. Delay Variation:* In this scenario, we vary the average delay constraints for the pairs in Group 2 in successive experiments while keeping the average delay constraints for the pairs in Group 1 constant in all the experiments. For Group 2, the delay constraints are varied as 100 – 160 msec in steps of 10 msec in successive experiments, while the delay constraints for Group 1 are kept at 100 msec in all experiments. The mean channel state  $\alpha$  for all the destinations is kept at  $-3.28$ dB (0.4698) for all the experiments. In each slot, we generate the channel state using the exponential distribution with mean  $\alpha$  and subsequently discretize it using the probability bins as mentioned above. Moreover, we generate the arrivals with the Poisson distribution such that the average arrival rate is 20 kbps at each source. The parameters have been summarized in Table 6.2. Each source makes the scheduling decision using its transmission probability  $\theta_n^i$  in each slot. Based on the feedback received, it then transitions its queue and determines the new value of the

Simulation Parameter	Value
Slot duration	1 msec
Bandwidth	10 MHz
Transmit power	1 watt
Simulation time	100000 slots
$\omega$	0.1
Packet size $\ell$	2000 bits
$N$	20
$e_n$	$\frac{1}{n^{0.6}}$
$b_n$	$\frac{1}{n^{0.9}}$
$c_n$	$\frac{1}{n}$
$f_n$	$\frac{0.05}{n}$

Table 6.1: Summary of parameters common for all scenarios

Simulation Parameter	Value
Delay constraint (Group 1)	100 msec
Delay constraint (Group 2)	100 – 160 msec in steps of 10 msec
Mean arrival rate (Group 1&2)	20 kbps/source
Mean channel state (Group 1&2)	0.4698 (−3.28 dB)

Table 6.2: Summary of parameters for Scenario 6.1

transmission probability. We select two source destination pairs,  $i, j$  and  $k, l$  at random from Group 1 and Group 2. For these pairs, we determine the delays experienced and stable transmission probabilities within each experiment and plot these in Figures 6.2 and 6.3 respectively. From Figure 6.2, it can be seen that all the delay constraints are satisfied. From Figure 6.3, it can be seen that higher delay constraints require lower transmission probabilities and consequently lower power consumption.

*Scenario 6.2. Arrival Variation:* In this scenario, we vary the average arrival rate for the pairs in Group 2 in successive experiments while keeping the average arrival rate for the pairs in Group 1 constant in all the experiments. We generate the arrivals with Poisson

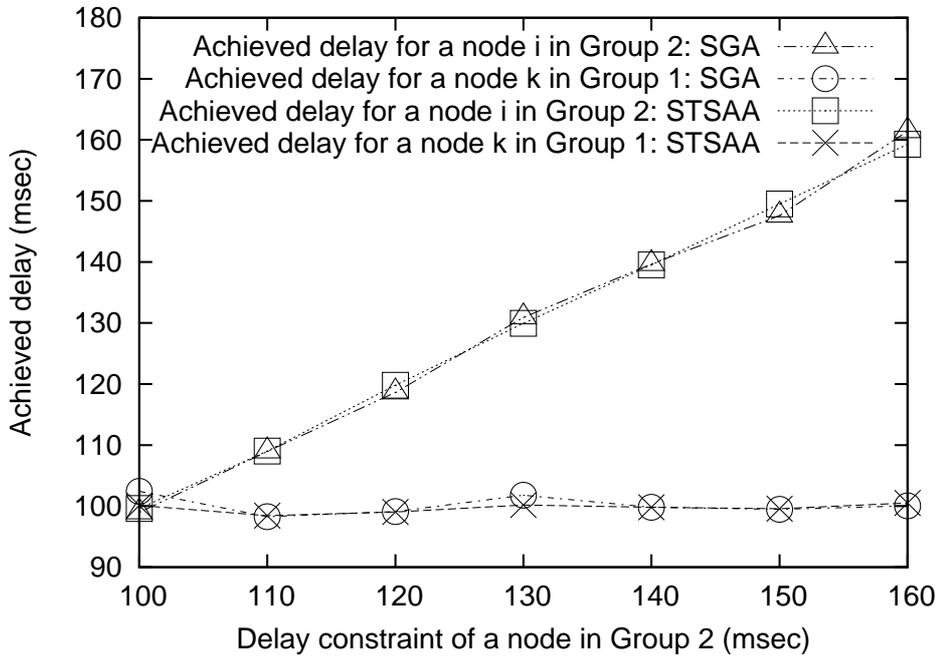


Figure 6.2: Achieved delay for various average delay constraints

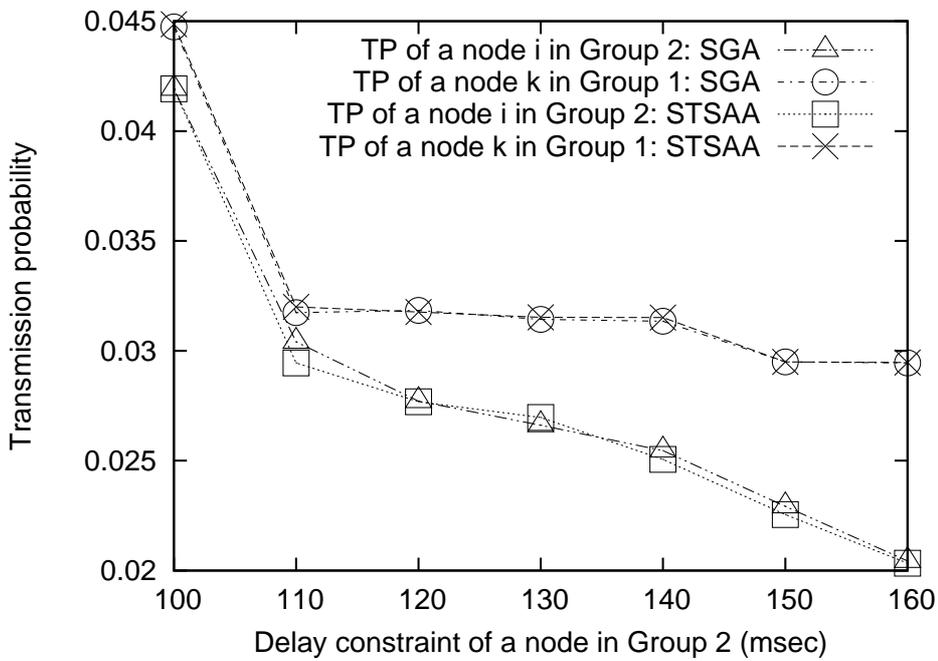


Figure 6.3: Transmission probability (TP) for various average delay constraints

Simulation Parameter	Value
Delay constraint (Group 1&2)	100 msec
Mean arrival rate (Group 1)	20 kbps/source
Mean arrival rate (Group 2)	12 – 36 kbps/source in steps of 4 kbps
Mean channel state (Group 1&2)	0.4698 (–3.28 dB)

Table 6.3: Summary of parameters for Scenario 6.2

distribution. For the pairs in Group 2, the average arrival rate is varied as 12 – 36 kbps in steps of 4 kbps in successive experiments, while the arrival rate for the pairs in Group 1 is kept constant at 20 kbps in all experiments. The average delay constraint is kept constant at 100 msec for all pairs. The mean channel state  $\alpha$  is kept at –3.28dB (0.4698) for all experiments. In each slot, we generate the channel state using exponential distribution with mean  $\alpha$  and subsequently discretize it using the probability bins as mentioned above. The parameters have been summarized in Table 6.3. We select two pairs  $i, j$  and  $k, l$  at random from Group 1 and Group 2. For these pairs, we determine the delays experienced and the stable transmission probabilities within each experiment and plot these in Figures 6.4 and 6.5 respectively. Figure 6.4 depicts the range of arrival rates for which the delay constraints are satisfied. From Figure 6.5, it can be seen that higher arrival rates require higher transmission probabilities and consequently higher power consumption.

*Scenario 6.3. Channel Variation:* In this scenario, we vary the average channel state for the pairs in Group 2 in successive experiments while keeping the average channel state for the pairs in Group 1 constant in all experiments. For the pairs in Group 2, the average channel state  $\alpha$  is varied as 0.05 (–13 dB), 0.1422 (–8.47 dB), 0.2877 (–5.41 dB), 0.4698 (–3.28 dB), 0.6934 (–1.59 dB), 0.9817 (–0.08 dB), 1.3867 (1.42 dB) in successive experiments, while the average channel state for the pairs in Group 1 is kept at 0.4698 (–3.28 dB) in all experiments. The average delay constraint is kept constant at 100 msec for all the pairs for all experiments. We generate the arrivals with Poisson distribution. The mean arrival rate for all the pairs is kept at 20 kbps for all experiments. In each time slot, we generate the channel state using the exponential distribution with mean  $\alpha$  and subsequently discretize it using the probability bins as mentioned above. The parameters have been summarized in Table 6.4. We select two pairs  $i, j$  and  $k, l$  at

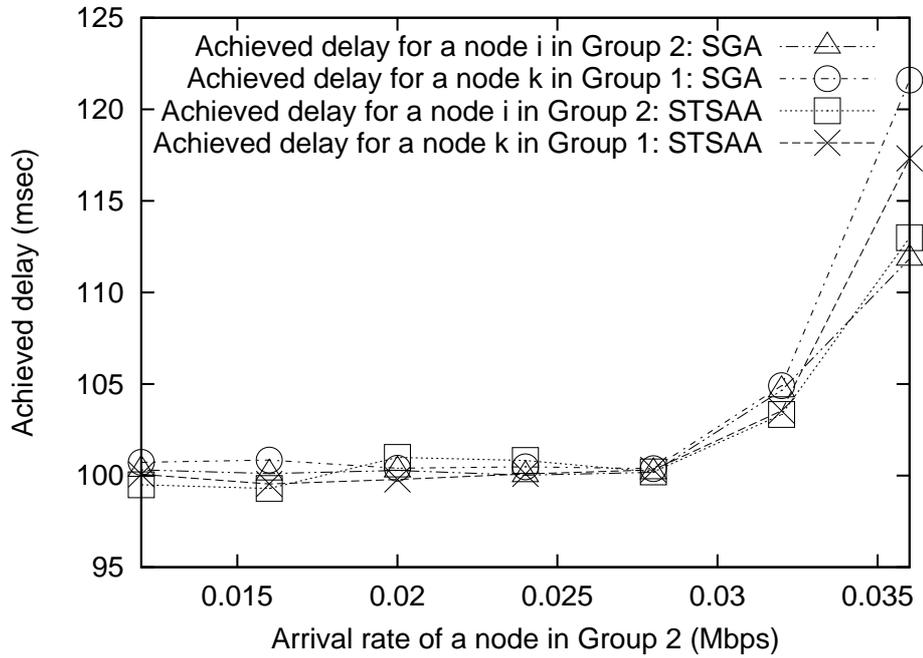


Figure 6.4: Achieved delay for various average arrival rates

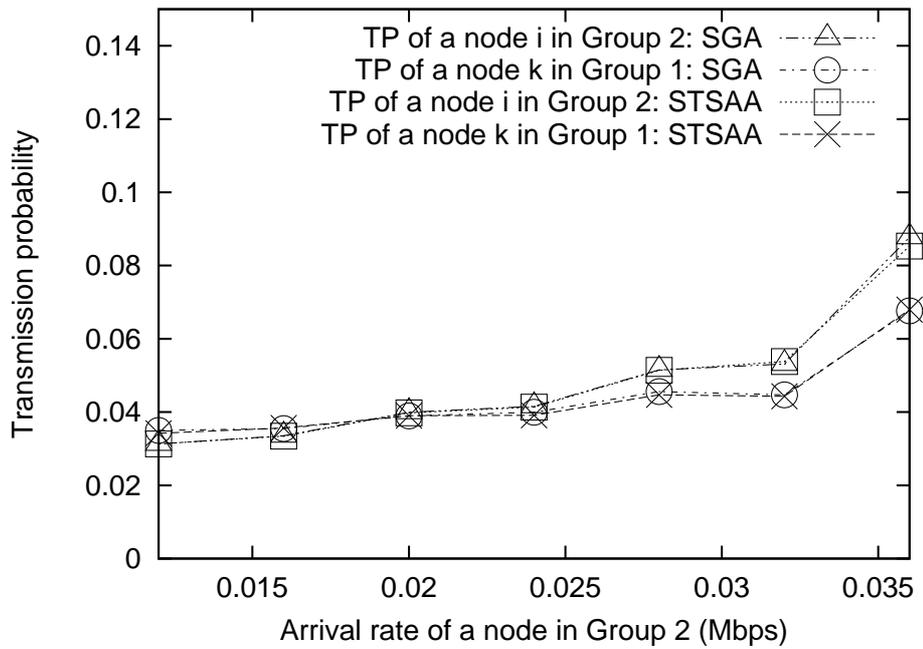


Figure 6.5: Transmission probability (TP) for various average arrival rates

Simulation Parameter	Value
Delay constraint (Group 1&2)	100 msec
Mean arrival rate (Group 1&2)	20 kbps/source
Mean channel state (Group 1)	0.4698 (−3.28 dB)
Mean channel state (Group 2)	−13 dB to 1.42 dB) in 7 steps

Table 6.4: Summary of parameters for Scenario 6.3

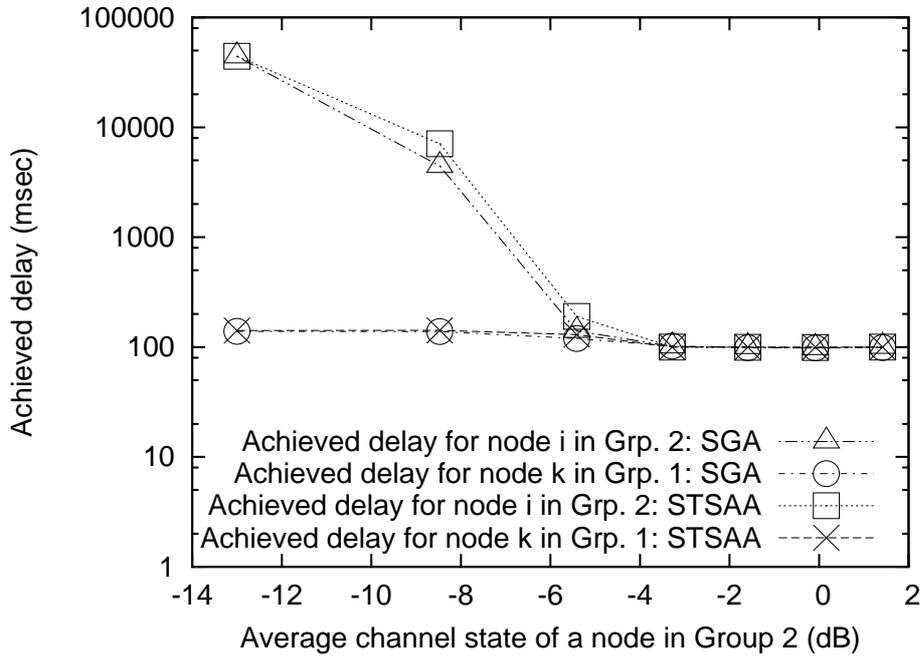


Figure 6.6: Achieved delay for various average channel states

random from Group 1 and Group 2. For these pairs, we determine the delays experienced and stable transmission probabilities within each experiment and plot these in Figures 6.6 and 6.7 respectively. Figure 6.6 depicts the range of channel states for which the delay constraints are satisfied. From Figure 6.7, it can be seen that better channel conditions result in lower transmission probabilities and consequently lower power consumption.

x

## 6.7 Conclusions

In this chapter, we have proposed a novel approach for a distributed multiaccess system, where each source iteratively adjusts its transmission probability with an objective of

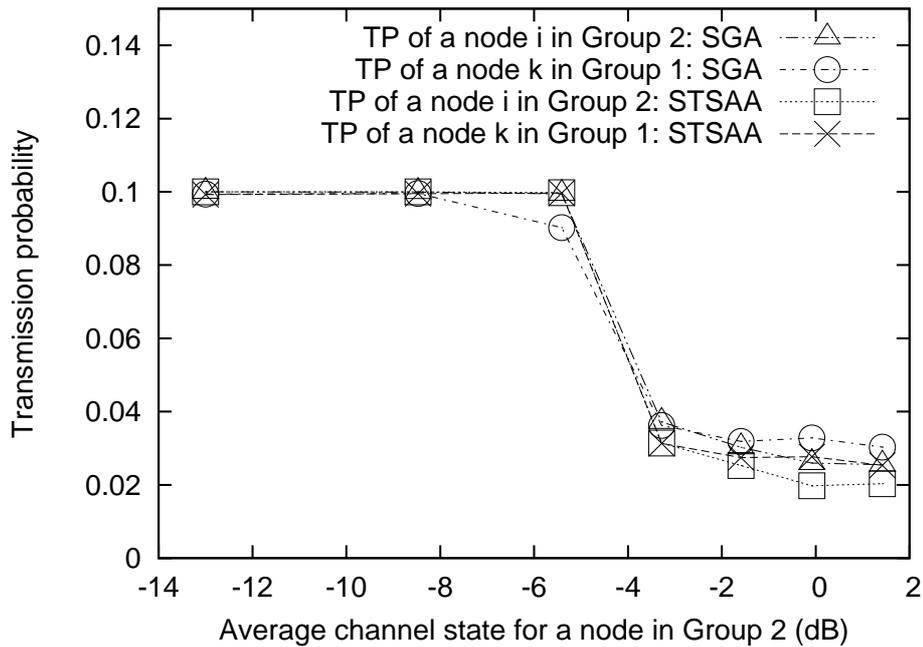


Figure 6.7: Transmission probability (TP) for various average channel states

determining the transmission probability just sufficient to satisfy the delay constraint. Moreover, we have considered channel state dependent variable rate transmission. While random access communication is a well studied problem, the delay constrained random channel access has not received much attention. We have formulated the problem as a constrained multistage optimization problem and provided two efficient solution techniques (SGA and STSAA) based on stochastic approximation. We have proved that STSAA converges to an equilibrium and satisfies the delay constraints. The proposed algorithms can be easily implemented in practice. Moreover, the simulation results have demonstrated that the specified delay constraints are, indeed, satisfied.

# Chapter 7

## Conclusions and Future Work

### 7.1 Conclusions

In this thesis, we have addressed cross layer scheduling over a fading wireless channel. As discussed in Chapter 2, a large number of scheduling algorithms catering to different objectives such as minimizing power, delay or packet loss, maximizing throughput, subject to some resource allocation or QoS constraints have been proposed in the literature. These problems can be formulated as constrained optimization problems. They are essentially control problems where the scheduler can be viewed as a controller operating in an uncertain environment. The objective of the scheduler is to determine an optimal scheduling policy such that the desired objective function is optimized while satisfying the required QoS.

Most of these scheduling policies can be determined by formulating the problem as a CMDP and then using the CMDP solution techniques such as LP or other iterative techniques [59]. However, these techniques face the curse of dimensionality, i.e., for a large state space, the techniques are computationally infeasible. Moreover, these techniques require knowledge of the transition probability mechanism of the underlying Markov chain [22]. This probability mechanism depends on the system model, i.e., probability distributions of the arrivals and channel state processes of the users. This knowledge, being difficult to possess in practice, further renders the traditional approaches infeasible for determining the optimal policy within the CMDP framework. To address this issue, one usually makes some assumptions regarding the system model. However, the performance

of schemes developed under the assumed system model is limited by the accuracy of the model. Hence, our approach has been to develop model unaware computationally efficient schemes in this thesis.

In summary, we have designed computationally efficient algorithms that determine optimal, or sub-optimal but highly efficient scheduling policies and do not require knowledge of the system model. Furthermore, we have proved analytically that the schemes satisfy the specified average delay guarantees. The proposed schemes can be easily adopted for implementation within practical systems such as IEEE 802.16 [5] or IEEE 802.11 [6]. We now summarize some of the significant contributions of the thesis in further detail.

We began our investigations with point to point or single user communication scenario in Chapter 3. Although this scenario is somewhat restrictive, it, nevertheless, provides an opportunity for studying various aspects related to efficient communication over the fading channel under a QoS constraint. For this scenario, we have considered the problem of providing an average delay guaranty while minimizing the long term average power expenditure. This problem has been previously formulated as a CMDP [27, 30, 29, 28]. However, an important issue of computing the optimal packet scheduling policy was not addressed. Moreover, as pointed out above, our arguments related to assumptions on the system model rendered the traditional CMDP solution techniques infeasible.

We have adopted model unaware techniques. Specifically, we have employed the Reinforcement Learning (RL) approach [23, 24]. Our strategy has been to learn the Dynamic Programming (DP) value function [22, 31] and thereby the optimal policy in an online manner. For this, we have reformulated the Relative Value Iteration Algorithm (RVIA) by introducing the concept of a post decision state. The resulting equation has a nice structure amenable to online implementation. Thus, we have obtained an online algorithm for determining the optimal value function. The constraints are naturally handled using the Lagrangian approach. However, the issue of determining the optimal Lagrange Multiplier (LM) remains. To resolve this issue, we have introduced coupled LM iterations having a slower update rate than the value function iterations.

The o.d.e. approach for proving the convergence of Stochastic Approximation (SA) algorithms [90, 25] has been employed to prove the convergence of the value function to the optimal value function for a fixed value of LM. Two timescale arguments [91]

along with the Envelope Theorem of mathematical economics [103] has enabled us to prove that the coupled iterations converge to respective optimal values. While theoretical analysis is important, we have also investigated the practical utility of the proposed algorithm through our simulations for different arrival rates, average channel states and delay constraints. These investigations have demonstrated the practical utility of the algorithm. Not only does it satisfy the specified constraints but converges to a ‘close’ vicinity of the optimal solution in ‘reasonable’ number of iterations for the algorithm to be practically useful.

In Chapter 4, we have considered an extension of the problem considered in Chapter 3 for multiuser uplink communication. The specific problem addressed is that of minimizing the power expenditure of each user on the uplink subject to satisfying individual user delay constraints. We believe that the said problem accurately captures the user requirements as compared to a well studied related problem [93] where the objective is to minimize the sum power on the uplink subject to individual user QoS constraint. The problem under consideration is a multi-objective optimization problem with a large state space, since a state comprises of the queue length and channel state information of all the users. An algorithm that explores the state space for determining the optimal policy is clearly computationally infeasible. Moreover, system model related arguments are applicable in this scenario also.

To address these issues, we have proposed a novel approach based on a modified version of the single user algorithm of Chapter 3. Our approach has considered an extension of the single user algorithm under a hypothetical scenario where the transmitter and scheduler are assumed to be two separate entities. The scheduler determines the optimal rate in each slot, however, the transmitter fails to transmit with a certain probability. We have proposed an optimal algorithm for this scenario. The algorithm is then used in the multiuser scenario, using which a user determines a transmission rate in each slot and informs it to the base station. The base station then schedules the user with the highest rate. Note that the users do not have an incentive to inform unnecessarily high rates since this can lead to higher power consumption than that is sufficient for satisfying the QoS requirements. The analogy between this scheme and the extended single user scheme is apparent. However, while in the single user scenario, the probability of transmitter failure

is independent of the scheduler action, in the multiuser scenario it is not. The problem is essentially a multiagent learning problem. Note that each user attempts to learn its own optimal scheduling policy by exploring on the single user state space. This resolves the state space explosion problem. The algorithm has an interesting interpretation as an auction. In each slot, the base station can be assumed to auction the slot. The user bidding the highest rate wins the auction. If a user does not win the auction for a long enough period of time, its QoS constraints may be violated, thus forcing it to bid at a higher rate.

We have argued that the scheme has a stabilizing structure and satisfies the delay constraints of all users. We believe that the scheme can be easily adopted for implementation in any orthogonal multiaccess system. The simulations within an IEEE 802.16 system demonstrated that the scheme satisfies the delay constraints of all the users.

In Chapter 5, we have considered the downlink scheduling problem where the objective is to maximize the sum throughput subject to delay constraints of each user. While downlink scheduling has been well studied in general, with various other scheduling objectives and constraints, the specific problem considered in Chapter 5 has not received sufficient attention. This problem can be formulated as a CMDP. The primary issue in determining the optimal policy is that of the large state space. This issue is compounded by the issue of unknown system model. Reinforcement learning algorithms as employed previously, could be employed in this case also to solve the later issue. However, with such a large state space, the reinforcement learning algorithms would take a prohibitive amount of time to converge to the optimal policy. This motivates the design of sub-optimal schemes.

We have suggested a sub-optimal indexing scheme based on generating appropriate indices for all users in each slot and then scheduling the user with the highest index. The indexing scheme has been designed to exploit the throughput-delay tradeoff. An index is a product of two terms: a term proportional to the channel state of the user and another proportional to the aggregate queue constraint violation. Iteratively adjusting the later term based on the queue length of the users results in the scheme satisfying the delay constraint of the users. The weights are analogous to LMs that ensure that the delay constraints are satisfied. To demonstrate that the scheme achieves a high sum

throughput, we have performed several simulations within the IEEE 802.16 framework. Our comparisons with an adapted version of the throughput optimal M-LWDF scheme [16] have indicated that the throughput achieved by the proposed scheme is very close to that achieved by the M-LWDF scheme. The delay constraints of all the users are also satisfied.

Departing from centralized scheduling aspects, in Chapter 6, we have focused on the distributed scheduling problem, which can be more appropriately described as a distributed access control problem, since in this case, there is no explicit scheduling entity as such. There is copious literature dealing with this problem. Random access communication is a well studied topic [10, 100]. Moreover, recently ad hoc networks [65] have attracted considerable research interest. While a more general problem of providing end-to-end QoS in the context of multihop wireless network involves issues such as routing apart from the access control aspect, we have not investigated these issues. Rather, in order to concentrate on the scheduling or access control aspect, we have considered a somewhat restricted scenario where all nodes are in close proximity and, therefore, are in the transmission range of each other. This implies that in order to avoid collisions, only one node can transmit in any time slot. Moreover, nodes are divided into source and destination pairs, and the destinations require QoS in the form of average delay guarantees from the sources. Each source transmits at a constant power and accesses the channel randomly. Since each access consumes some amount of energy, the objective is to determine the minimum steady state access rate or transmission probability for each source so that the delay constraints are satisfied.

We have proposed two iterative algorithms for determining the transmission probability. The first algorithm is referred to as SGA in the chapter, and is based on the stochastic gradient algorithm. It involves update of three quantities on three timescales. The idea is to update the transmission probability of a node using the gradient of the Lagrangian. Though we have not performed the convergence analysis of the algorithm, it being a three timescale algorithm, we expected the convergence of the algorithm to be slow. Moreover, it is difficult to analyze its properties analytically. The second scheme referred to as STSAA is a single timescale stochastic approximation algorithm. We have proved that STSAA satisfies the delay constraints of the users. We have also proved that

it converges to an equilibrium.

The central theme of this thesis has been to develop computationally efficient algorithms that determine optimal or sub-optimal but highly efficient scheduling policies in the absence of model knowledge. While we have addressed the scheduling problem for various scenarios by proposing efficient algorithms, several extensions of these algorithms are possible. Moreover, the formulations developed in this thesis can be extended to more general scenarios. In the next section, we discuss some of these extensions.

## 7.2 Future Work

For the algorithms proposed in Chapters 3-6, we have performed an asymptotic convergence analysis. Further investigations into the practical utility of the algorithms could be performed by analyzing the rate of convergence of these algorithms. This would quantify the proximity of the solution determined by these algorithms to the optimal or steady state value after a certain number of iterations. Moreover, the impact of various quantities such as variance of the arrival and channel state processes as well as the number of users in the multiuser scenario on the convergence rate needs to be investigated. In cases where the convergence rate is not sufficient, techniques for speeding up the convergence need to be employed. One such approach could be to exploit the structural properties of the optimal policy.

In Chapter 4, we have argued that the Auction Algorithm satisfies the delay constraints and has a stabilizing behavior. Providing a more rigorous proof for analyzing the convergence properties of the algorithm is an interesting future work.

While we have performed simulation studies to demonstrate the practical utility of the algorithms, evaluating these algorithms in more practical scenarios such as IEEE 802.16 or IEEE 802.11 test-beds would be interesting. This may require extending the existing protocols in order to accommodate these algorithms. We believe that the multiuser uplink algorithm proposed in Chapter 4 and the Indexing Scheduler (IS) proposed in Chapter 5 could be easily accommodated within the IEEE 802.16 framework with minor modifications to the existing protocol. The SGA and SA (algorithms proposed in Chapter 6) can be easily adapted for access control and QoS provisioning within any

practical system. In the IEEE 802.11 system, e.g., the algorithms can be a replacement for the exponential backoff mechanism. The Request To Send (RTS) packet could be transmitted using a probability determined by the algorithms. Studying the performance of these algorithms with the proposed modifications within the IEEE 802.11 framework would be worth investigating.

An interesting aspect that needs investigation is an interplay of power control as considered in Chapter 4, with access control considered in Chapter 6. The problem would then take the form of power minimization with constraint on the average delay under the distributed access control scenario. The resulting scheme would be a generalization of the Power Controlled Multiple Access (PCMA) protocol [104].

As discussed in Chapter 1, typically in a network, user applications have diverse QoS requirements. In this thesis, we have focused on providing average delay guarantees to the users. However, we believe that the algorithms developed in this thesis could be easily modified for providing other QoS guarantees such as average rate guarantees or average packet drop guarantees. These modifications form an interesting future work.

This thesis has only concentrated on providing QoS guarantees in orthogonal multiple access systems (primarily in TDMA systems, although we believe that the extension to other orthogonal systems such as Frequency Division Multiple Access (FDMA) is rather straightforward) with single transmit and receive antenna. Extensions of the algorithms developed in this thesis to multiple antenna systems [20] remain to be investigated. Multiple antenna considerations further increase the state space size, since now information regarding the transmit or receive antenna has to be included in the state.

Eventually, the goal of wireless communication is to provide end-to-end QoS in a challenging uncertain environment (wireless channel) by optimal, computationally efficient and practically implementable scheduling algorithms. We believe that the algorithms developed in this thesis form a useful first step towards this goal by providing MAC layer QoS. The proposed algorithms are simple to implement and have been analyzed by employing the theory of stochastic approximation. However, much remains to be done for providing end-to-end QoS!



# Appendix A

## Markov Decision Process

In this appendix, we review fundamentals of Markov Decision Process (MDP). Since discrete (time and state) MDP with average cost has been used extensively in the thesis, our focus is on reviewing results related to such MDP. Moreover, we also review some results regarding Constrained MDP (CMDP). The treatment in this Appendix is standard and follows texts [31, 59].

### A.1 Markov Decision Process

Consider dynamical systems that evolve in a stochastic fashion. MDP constitutes a basic framework for controlling such systems where decisions taken by an entity called the *controller* influence the system evolution. *State* of the system is a minimal set of parameters that capture the information regarding the system. System evolution thus refers to state transition. Each decision or *action* taken by the controller incurs an *immediate* cost and influences the future state transitions. The current state and control action fully determine the probability of transition to any given state. The state transition is probabilistic because of the action of the environment on the system through a *noise* or *disturbance* process. An MDP is a generalization of a non-controlled Markov chain. A key Markovian property: conditioned on the state and action at a time  $n$ , the past states and the next one are independent; applies to MDP also.

For a discrete time MDP, the system state is observed at discrete instants of time  $n = 1, 2, \dots, M$ .  $M$  is called the *horizon* of the system. It can be either finite or infinite.

We focus on an MDP over an infinite horizon. Let us denote the time intervals between successive observation instances as slots. Let  $S_n$  denote the state of the system in slot  $n$ . The state evolution can be described by the stochastic process  $\{S_n\}$ . For a discrete state MDP, the state variable  $S_n$  draws values from a discrete state space  $\mathbb{S}$ . We focus on an MDP with finite state space. Let  $U_n \in \mathbb{U}$  denote the action taken by the controller in slot  $n$ . Each state has a set of feasible actions  $\mathbb{F}_n(S_n) \subset \mathbb{U}$ ;  $U_n \in \mathbb{F}_n(S_n)$ . The actions taken by the controller can be described using the stochastic process  $\{U_n\}$ . We consider an MDP with discrete and finite action space. The random noise is characterized by a transition probability mechanism  $p_n(S_n, U_n, \cdot)$ , where  $p_n(S_n, U_n, S'_n)$  denotes the probability of taking an action  $U_n$  in state  $S_n$  and transitioning to state  $S'_n$  at time  $n$ . Let  $c_n : \mathbb{S} \times \mathbb{U} \rightarrow \mathbb{R}$  be the immediate cost function.  $c_n(S_n, U_n)$  denotes the immediate cost incurred in taking an action  $U_n$  at state  $S_n$  at time  $n$ . We assume stationarity, thereby, the feasible action set, the transition probabilities and the immediate cost function do not change with time and hence we drop the subscript  $n$  in the notation; these quantities are then denoted as  $\mathbb{F}(S_n)$ ,  $p(S_n, U_n, S'_n)$  and  $c(S_n, U_n)$  respectively.

A decision function  $\mu_n : \mathbb{S} \rightarrow \mathbb{F}(S_n)$  at time  $n$  is a mapping from the state space to the feasible action space. A policy is a sequence of decision functions over the problem horizon. For an infinite horizon problem, a policy  $\pi$  is represented as  $\{\mu_1, \mu_2, \dots\}$ . A *stationary* policy is a sequence of decision functions of the form  $\{\mu, \mu, \dots\}$ . Thus a stationary policy depends only on the current state of the system and is independent of the time at which the decision is taken.

The control problem in an MDP consists of minimizing the expected cumulative cost (expectation taken with respect to the random noise) over a specified horizon. For infinite horizon problems, average cost is a better notion of cost instead of cumulative cost. Hence the average cost problem is specified as:

$$\min \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \left\{ \sum_{n=0}^{M-1} c(S_n, U_n) \right\}. \quad (\text{A.1})$$

Let  $V^\pi(S_0)$  denote the cost incurred with initial state  $S_0$  under a policy  $\pi = \{\mu_0, \mu_1, \dots\}$ . Then  $V^\pi(S_0)$  can be expressed as:

$$V^\pi(S_0) = \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \left\{ \sum_{n=0}^{M-1} c(S_n, \pi(S_n)) \right\}. \quad (\text{A.2})$$

The *optimal value function*  $V(s)$  at a state  $s$  denotes the minimum cost incurred with  $s$  as the initial state under any policy, i.e.,

$$V(s) = \min_{\pi} V^{\pi}(s), \quad \pi \in \Pi, \quad (\text{A.3})$$

where  $\Pi$  is the set of all policies.

The following equation called the average cost Bellman's equation is the cornerstone of all dynamic programming (DP) formulation of average cost problems;

$$V(s) = \min_{u \in \mathbb{F}(s)} \left\{ c(s, u) + \sum_{s'} p(s, u, s') V(s') \right\} - \beta, \quad (\text{A.4})$$

where  $\beta$  is the optimal average cost. Note that above equation is actually a set of equations, one for each state and provides an optimality criterion. In this equation  $V(s)$  denotes the average cost incurred from a state  $s$  over and above the optimal average cost  $\beta$ .

A stationary *randomized* policy is a sequence of decision functions  $\{\mu, \mu, \dots\}$  such that  $\mu$  specifies a probability  $\rho(s, u)$  for taking each action  $u \in \mathbb{F}(s)$ , and there exists at least one state where the optimal decision is a randomization between two or more actions. On the other hand, a stationary *deterministic* policy is a sequence of decision functions  $\{\mu, \mu, \dots\}$  such that  $\rho(s, u)$ , the probability of taking action  $u \in \mathbb{F}(s)$  in each state  $s$ , is 1 for exactly one action and is 0 for all other actions.

**Definition A.1.** An MDP is said to be *unichain* if under any stationary deterministic policy, the corresponding Markov chain contains a single aperiodic ergodic class.

The Relative Value Iteration Algorithm (RVIA) is an algorithm for determining the optimal value function  $V(\cdot)$  and the optimal average cost  $\beta$  in an iterative fashion. RVIA can be expressed as:

$$V_{n+1}(s) = \min_{u \in \mathbb{F}(s)} \left\{ c(s, u) + \sum_{s'} p(s, u, s') V_n(s') \right\} - V_n(s^0), \quad (\text{A.5})$$

where  $s^0$  is a pre-determined state and  $V_n(\cdot)$  is the estimate of the optimal value function after  $n$  iterations of the algorithm. Note that  $V_n(\cdot) \rightarrow V(\cdot)$  and  $V_n(s^0) \rightarrow \beta$  for a unichain MDP.

## A.2 Constrained Markov Decision Process (CMDP)

A CMDP corresponds to an MDP where the controller attempts to minimize one objective cost subject to constraints on other costs. Let us consider  $K$  constraints. Let  $\mathbf{d} : \mathbb{S} \times \mathbb{U} \rightarrow \mathbb{R}^K$  denote the  $K$  dimensional vector of immediate costs related to the  $K$  constraints. An average cost problem formulated within the CMDP framework can be expressed as:

$$\begin{aligned} \min \quad & \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=0}^{M-1} c(S_n, U_n), \\ & \text{subject to,} \\ & \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=0}^{M-1} \mathbf{d}(S_n, U_n) \leq \boldsymbol{\delta}, \end{aligned} \quad (\text{A.6})$$

where  $\boldsymbol{\delta} = [\delta^1, \dots, \delta^K]$  is the  $K$  dimensional constraint vector.

Let  $\boldsymbol{\lambda} \in \mathbb{R}_+^K$  be a  $K$  dimensional non-negative vector. We convert the constrained problem defined in (A.6) into an unconstrained problem using the Lagrangian approach. Define  $L(\boldsymbol{\lambda}, s, u)$  as:

$$L(\boldsymbol{\lambda}, s, u) = c(s, u) + \boldsymbol{\lambda}^T (\mathbf{d}(s, u) - \boldsymbol{\delta}). \quad (\text{A.7})$$

The objective now is to solve an unconstrained problem with immediate cost  $L(\boldsymbol{\lambda}, s, u)$  for a fixed value of the  $\boldsymbol{\lambda}$  vector termed as the Lagrange Multiplier (LM) vector. This unconstrained problem can be expressed as:

$$\min \quad \limsup_{M \rightarrow \infty} \frac{1}{M} \mathbf{E} \sum_{n=0}^{M-1} L(\boldsymbol{\lambda}, S_n, U_n). \quad (\text{A.8})$$

This problem can be solved using the following linear program:

$$\begin{aligned} \max_V \quad & \sum_s V(s), \\ & \text{subject to,} \\ & L(\boldsymbol{\lambda}, s, u) + \sum_{s'} p(s, u, s') V(s') - \beta \geq V(s), \quad \forall s \in \mathbb{S}, \forall u \in \mathbb{F}(S). \end{aligned} \quad (\text{A.9})$$

A dual LP to the one specified in (A.9) can be specified as follows:

$$\begin{aligned}
& \min_{\rho} \sum_s \sum_u \rho(s, u) L(\boldsymbol{\lambda}, s, u), \\
& \text{subject to,} \\
& \sum_u \rho(s, u) = \sum_{s'} \sum_u \rho(s', u) p(s', u, s), \\
& \sum_s \sum_u \rho(s, u) = 1, \\
& \rho(s, u) \geq 0, \quad \forall s, u.
\end{aligned} \tag{A.10}$$

The following theorem provides a bound on the number of randomizations under a stationary randomized policy [59].

**Theorem A.1.** *Under an average cost unichain CMDP with  $K$  constraints, if it is feasible to satisfy the constraints, then there exists an optimal stationary policy  $\mu$  such that the total number  $n(\mu)$  of randomizations is at most  $K$ , i.e., it randomizes between at most  $K + 1$  actions.*



# Appendix B

## Reinforcement Learning

In this appendix, we introduce Reinforcement Learning (RL) and review an important RL algorithm: the Q-learning algorithm. We follow texts [24, 23].

### B.1 Reinforcement Learning

The reinforcement learning problem concerns itself with *learning from interaction* to achieve a goal. A *learning agent* interacts with the environment and takes certain actions based on this interaction in order to maximize a certain reward or to minimize a certain cost. One can easily see the analogy between a learning agent and a controller operating in an uncertain environment as considered in Appendix 1. However, the DP algorithms like value iteration considered in Appendix 1 assume a perfect statistical knowledge of the environment, i.e., the environment model is known. To make this statement precise, consider the RVIA studied in Appendix 1:

$$V_{n+1}(s) = \min_{u \in \mathbb{F}(s)} \left\{ c(s, u) + \sum_{s'} p(s, u, s') V_n(s') \right\} - V_n(s^0). \quad (\text{B.1})$$

Note that this algorithm assumes a knowledge of the transition probability mechanism  $p(s, u, s')$ , which in turn depends on the knowledge of the environment model. The RL algorithms do not assume any such knowledge. Moreover, the DP algorithms are faced by the *curse of dimensionality*, i.e., the computational complexity of these algorithms is prohibitively high for a large state space. RL algorithms are computationally more efficient and operate without perfect model knowledge.

We now study a well known RL algorithm: the Q-learning algorithm and show how Q-learning overcomes the curse of dimensionality and lack of model knowledge.

## B.2 Q-learning

The Q-learning algorithm introduces what are called Q-factors:  $Q : \mathbb{S} \times \mathbb{U} \rightarrow \mathbb{R}$ .  $Q(s, u)$  is the expected cost for taking an action  $u$  in a state  $s$  and thereafter following an optimal policy. Let  $C(s, u, s')$  be the immediate cost incurred in taking an action  $u$  in state  $s$  and moving to a state  $s'$ .  $c(s, u)$  can be expressed in terms of  $C(s, u, s')$  as:

$$c(s, u) = \sum_{s'} p(s, u, s') C(s, u, s'). \quad (\text{B.2})$$

For an average cost problem,  $Q(s, u)$  can be expressed as:

$$Q(s, u) = \sum_{s'} p(s, u, s') \left( C(s, u, s') + \min_{v \in \mathbb{F}(s')} Q(s', v) \right) - \beta, \quad (\text{B.3})$$

where  $\beta$  is the optimal average cost per stage. (B.3) is the Bellman's equation expressed in terms of the Q-factors. One can write an iterative equation for determining the optimal Q-factors. This iterative equation is called the Q-learning algorithm and can be expressed as:

$$Q_{n+1}(s, u) = C(s, u, S') + \min_{v \in \mathbb{F}(S')} Q_n(S', v) - \min_{v \in \mathbb{F}(s^0)} Q_n(s^0, v), \quad (\text{B.4})$$

where  $s^0$  is a fixed state.  $S'$  is generated from  $(s, u)$  according to the transition probability  $p(s, u, S')$ .

Note that (B.4) does not require the knowledge of the transition probability mechanism  $p(s, u, s')$ . Moreover, since the Q-factors are learnt one state at a time using simulation (via  $p(s, u, s')$ ), the algorithm is computationally efficient. Learning the optimal Q-factors requires that each state-action pair be visited *substantial* number of times. If the state space is large, Q-learning can take a long time to converge to the optimal Q-factors.

# Appendix C

## Stochastic Approximation

In this appendix, we introduce Stochastic Approximation (SA) and review the ordinary differential equation (o.d.e.) approach for proving the convergence of the SA algorithms. The material in this appendix is based on texts [24, 26, 105].

### C.1 Stochastic Approximation and Stochastic Iterative Algorithm (SIA)

We have seen in Appendix 1, that we can employ the value iteration algorithm in order to obtain a solution to the Bellman's equations. Iterative algorithms are often employed in order to obtain a solution to a system of equations. However, in certain cases, only noise corrupted information regarding the variables involved in an iteration is available. SA algorithms or SIA's are variants of deterministic iterative algorithms that can work in the presence of noise. These algorithms can be used to solve optimization problems as well as for solving systems of equations. To illustrate this point, assume that we are interested in solving a system of equation of the form:

$$X = h(X), \tag{C.1}$$

where  $X$  is a variable in  $d$  dimensions and  $h(\cdot)$  is a function from  $\mathbb{R}^d$  onto itself. Now consider the case where  $h(\cdot)$  takes the form  $h(X) = \nabla f(X) - X$ , for some cost function  $f(\cdot)$ . In that case (C.1) takes the form:

$$\nabla f(X) = 0, \tag{C.2}$$

which is related to the problem of finding a minimum of the function  $f(\cdot)$ .

One possible algorithm for solving (C.1) can be expressed as:

$$X_{n+1} = X_n + ah(X_n), \quad (\text{C.3})$$

where  $a$  is a small step size. Now assume that the exact functional form of  $h(\cdot)$  is not known or that evaluation of  $h(X_n)$  is difficult, but we have an access to  $h(X_n) + M_{n+1}$ ,  $M_{n+1}$  being a random noise term, i.e., we have access to a noise corrupted version of  $h(X_n)$ . In that case, we replace (C.3) with the following iterative algorithm:

$$X_{n+1} = X_n + a(h(X_n) + M_{n+1}), \quad (\text{C.4})$$

The resulting algorithm is called as a SA algorithm or a SIA. Using a smaller step size  $a$  reduces the sensitivity to the noise  $M_{n+1}$  but makes the algorithm slow. To resolve this conflict, one can use a gradually decreasing step size  $a_n$ . The resulting algorithm can be expressed as:

$$X_{n+1} = X_n + a_n(h(X_n) + M_{n+1}), \quad (\text{C.5})$$

where the following properties are imposed on the step size sequence  $\{a_n\}$ :

$$\lim_n a_n = 0, \quad \sum_n (a_n)^2 < \infty, \quad \sum_n a_n = \infty. \quad (\text{C.6})$$

The second property ensures that the sequence  $\{a_n\}$  converges to 0 sufficiently rapidly, while the third property ensures that it does not converge to 0 too rapidly. With this brief introduction to SA algorithms, we proceed to describe an important method for analyzing the asymptotic behavior of these algorithms.

## C.2 Convergence Analysis

In this section, we describe the popular o.d.e. method [90, 25] for analyzing the asymptotic properties of SA algorithms. We assume that the noise sequence  $\{M_n\}$  satisfies:

$$\mathbf{E}[M_{n+1} | M_m, X_m, m \leq n] = 0 \quad \forall n. \quad (\text{C.7})$$

Thus,

$$Z_n = \sum_{i=0}^{n-1} a_i M_{i+1}, \quad n \geq 1, \quad (\text{C.8})$$

satisfies:

$$\mathbf{E}[Z_{n+1}|M_m, X_m, m \leq n] = Z_n \quad \forall n, \quad (\text{C.9})$$

i.e.,  $Z_n$  is a martingale process. If we assume that  $\sup_n \mathbf{E}[|M_n|] < \infty$ , the martingale convergence theorem ensures that  $Z_n$  converges with probability one (w.p.1). Moreover, we assume the stability of the iterates, i.e.,  $\sup_n \|X_n\| < \infty$ . Consider a well-posed o.d.e. in  $\mathbb{R}^d$ :

$$\dot{x}(t) = h(x(t)), \quad (\text{C.10})$$

which is assumed to have a globally asymptotically stable attractor set  $J$ . Let  $J^\epsilon$  denote an  $\epsilon$ -neighborhood of  $J$  for some  $\epsilon > 0$ . A bounded function  $y(\cdot) : [0, \infty] \rightarrow \mathbb{R}^d$  is said to be a  $(T, \delta)$ -perturbation of (C.10) for some  $T, \delta > 0$  if one can determine  $T_n \uparrow \infty$  such that  $T_{n+1} - T_n \geq T$ ,  $\forall n$  and solutions  $x^n(t)$ ,  $t \in I_n \triangleq [T_n, T_{n+1}]$  of (C.10) such that:

$$\sup_{t \in I_n} \|x^n(t) - y(n)\| < \delta. \quad (\text{C.11})$$

The following theorem is useful in analyzing the asymptotic behavior of the SA algorithm (C.5).

**Theorem C.1.** *Given  $\epsilon > 0$  and  $T > 0$ , there exists  $\delta_0 > 0$  such that for every  $\delta < \delta_0$  a  $(T, \delta)$ -perturbation  $y(\cdot)$  of (C.10) converges to  $J^\epsilon$ .*

The way to use this theorem is to define an interpolated trajectory of the algorithm (C.5) and interpret it as a  $(T, \delta)$ -perturbation of (C.10). Since this interpolated trajectory converges to  $J^\epsilon$  by Theorem C.1 w.p.1,  $X_n \rightarrow J^\epsilon$  w.p.1.

### C.3 Stochastic Approximation on Two Timescales

In many situations, one needs to estimate two variables (solve two systems of equations) that are dependent on each other (coupled), simultaneously. These equations can be of the form:

$$\begin{aligned} X &= h_1(X, Y), \\ Y &= h_2(X, Y). \end{aligned} \quad (\text{C.12})$$

As before, assume that only noisy measurements of  $h_1(\cdot, \cdot)$  and  $h_2(\cdot, \cdot)$  are available. In such cases, one can write SIA (C.5) for each of the variables as:

$$\begin{aligned} X_{n+1} &= X_n + a_n (h_1(X_n, Y_n) + M_{n+1}^1), \\ Y_{n+1} &= Y_n + b_n (h_2(X_n, Y_n) + M_{n+1}^2), \end{aligned} \tag{C.13}$$

where  $\{M_n^1\}, \{M_n^2\}$  are noise sequences that have properties specified in (C.7). One can view these iterations as two loops; one needs to wait for a near convergence of the variable in one loop in order to iterate on the other variable. The same effect can be achieved by updating both the variables, but with different timescales. This can be done by choosing different update rates or step sizes for the two variables. The variable that is updated on a faster timescale is considered to form the ‘inner’ loop, while the one on the slower timescale forms the ‘outer’ loop. Assume that the sequences  $\{a_n\}$  and  $\{b_n\}$  have properties specified in (C.6), with  $\frac{a_n}{b_n} \rightarrow 0$ . This ensures that  $Y_n$  is updated on a faster timescale. The fast component (inner loop) then sees the slow component as almost static, while the slow component (outer loop) sees the fast one as almost equilibrated. One can then show that both variables  $X_n$  and  $Y_n$  converge to the appropriate stable attractor sets [91].

# Appendix D

## Properties of Auction Algorithm (4.19)

In this Appendix, we prove some properties of the Auction Algorithm (4.19).

**Lemma D.1.** *The rate which a user  $i$  informs to the base station in a state  $s^i$ ,  $r^i(s^i)$ , is an increasing function of  $\lambda^i$ .*

*Proof.* Since  $\lambda^i$  is a part of the immediate cost  $c(\cdot, \cdot, \cdot, \cdot, \cdot)$ , the value function depends on  $\lambda^i$ . We make this dependence explicit by introducing  $\lambda^i$  in the notation. Let the post decision state at time  $n$  be  $\tilde{s}^i = (q^i, x^i)$ . The value function for user  $i$  after  $n$  iterations of RVIA is now represented using the notation  $\tilde{V}_n^i(\lambda^i, (q^i, x^i))$ . In order to prove the monotonicity of  $r^i(s^i)$  in  $\lambda^i$ , we prove the supermodularity of the value function in  $(r^i, q^i, \lambda^i)$ , i.e., we show that  $\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) + \tilde{V}_n^i(\lambda^i; q^i, x^i) \geq \tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i, x^i) + \tilde{V}_n^i(\lambda^i; q^i + \Delta q^i, x^i)$ .

We prove the result by induction. Let  $\tilde{V}_0^i(\lambda^i; q^i, x^i) = 0, \quad \forall \lambda^i, q^i, x^i$ . It can be verified that  $\tilde{V}_0^i(\cdot; \cdot, \cdot)$  is supermodular. Assume that  $\tilde{V}_n^i(\cdot; \cdot, \cdot)$  is supermodular, i.e.,  $\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) + \tilde{V}_n^i(\lambda^i; q^i, x^i) \geq \tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i, x^i) + \tilde{V}_n^i(\lambda^i; q^i + \Delta q^i, x^i)$ .

We now prove that  $\tilde{V}_{n+1}^i(\cdot; \cdot, \cdot)$  is supermodular. Let  $r_1$  be optimal at  $(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i)$  and  $r_2$  be optimal at  $(\lambda^i; q^i, x^i)$ . Let  $\zeta(a)$  be the unknown law of arrivals. Let  $\kappa(x'|x)$  be the unknown law for the channel fading.

- Case 1:  $r_1 - r_2 \geq \Delta q^i$ :

$$\begin{aligned}
& \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i + \Delta q^i, x^i) = \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i) \zeta(a^i) \\
& \left\{ (\lambda^i + \Delta\lambda^i)(q^i + \Delta q^i) - \lambda(q^i + \Delta q^i) + \theta_n^i [P(x^i, r_1) + \right. \\
& \tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i - r_1, x^i) - P(x^i, r_1) - \tilde{V}_n^i(\lambda^i; q^i + \Delta q^i + a^i - r_1, x^i)] \\
& \left. + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i, x^i) - \tilde{V}_n^i(\lambda^i; q^i + \Delta q^i + a^i, x^i)] \right\}. \quad (D.1)
\end{aligned}$$

Substitute  $q_1 = q^i + a^i - r_2$  and  $\Delta q_1 = r_2 - r_1 + \Delta q^i$ . Hence  $q_1 + \Delta q_1 = q^i + a^i - r_1 + \Delta q^i$ .

$$\begin{aligned}
& \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i + \Delta q^i, x^i) = \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i) \zeta(a^i) \left\{ \right. \\
& \Delta\lambda^i(q^i + \Delta q^i) + \theta_n^i [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q_1 + \Delta q_1, x^i) - \tilde{V}_n^i(\lambda^i; q_1 + \Delta q_1, x^i)] \\
& \left. + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i, x^i) - \tilde{V}_n^i(\lambda^i; q^i + \Delta q^i + a^i, x^i)] \right\}. \quad (D.2)
\end{aligned}$$

Now consider:

$$\begin{aligned}
& \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i, x^i) = \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i) \zeta(a^i) \left\{ \Delta\lambda(q^i) \right. \\
& + \theta_n^i [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + a^i - r_2, x^i) - \tilde{V}_n^i(\lambda; q^i + a^i - r_2, x^i)] \\
& \left. + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + a^i, x^i) - \tilde{V}_n^i(\lambda^i; q^i + a^i, x^i)] \right\}. \quad (D.3)
\end{aligned}$$

Again substitute  $q_1 = q^i + a^i - r_2$  and  $\Delta q_1 = r_2 - r_1 + \Delta q^i$ . Hence  $q_1 + \Delta q_1 = q^i + a^i - r_1 + \Delta q^i$ .

$$\begin{aligned}
& \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i, x^i) = \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i) \zeta(a^i) \left\{ \Delta\lambda(q^i) \right. \\
& + \theta_n^i [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q_1, x^i) - \tilde{V}_n^i(\lambda^i; q_1, x^i)] + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + a^i, x^i) \\
& \left. - \tilde{V}_n^i(\lambda^i; q^i + a^i, x^i)] \right\}. \quad (D.4)
\end{aligned}$$

From (D.2) and (D.4) using supermodularity (hence increasing differences property) of  $V_n^i(\cdot, \cdot, \cdot)$  and the fact that value function is an increasing function of the queue length [27], we obtain,

$$\begin{aligned}
& \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i + \Delta q^i, x^i) \\
& \geq \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i, x^i) - \tilde{V}_{n+1}^i(\lambda^i; q^i, x^i). \quad (D.5)
\end{aligned}$$

After rearranging the terms, we get the required result.

- $r_1 - r_2 \leq \Delta q^i$ :

Note that  $P(x^i, r)$  is an increasing convex function of  $r$  and hence we can write,

$$P(x^i, r_1) + P(x^i, r_2) \geq P(x^i, r_1 - \Delta q^i) + P(x^i, r_2 + \Delta q^i). \quad (\text{D.6})$$

Now,

$$\begin{aligned} & \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i, x^i) + \tilde{V}_{n+1}^i(\lambda^i; q^i, x^i) = \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i)\zeta(a^i) \left\{ (\lambda^i + \Delta\lambda^i) \right. \\ & (q^i + \Delta q^i) + \lambda q^i + \theta_n^i [P(x^i, r_1) + P(x^i, r_2) + \tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i - r_1, x^i) \\ & + \tilde{V}_n^i(\lambda^i; q^i + a^i - r_2, x^i)] + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i, x^i) \\ & \left. + \tilde{V}_n^i(\lambda^i; q^i + a^i, x^i)] \right\} \\ & \geq \sum_{x^{i'}} \sum_{a^i} \kappa(x^{i'}|x^i)\zeta(a^i) \left\{ (\lambda^i + \Delta\lambda^i)(q^i) + \lambda^i(q^i + \Delta q^i) + \theta_n^i [P(x^i, r_1 - \Delta q^i) \right. \\ & + P(x^i, r_2 + \Delta q^i) + \tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + a^i - (r_1 + \Delta q^i), x^i) \\ & + \tilde{V}_n^i(\lambda^i; q^i + a^i - (r_2 + \Delta q^i), x^i)] \\ & \left. + (1 - \theta_n^i) [\tilde{V}_n^i(\lambda^i + \Delta\lambda^i; q^i + \Delta q^i + a^i, x^i) + \tilde{V}_n^i(\lambda^i; q^i + a^i, x^i)] \right\} \\ & \geq h_{n+1}(r_1 - \Delta q^i, \lambda^i + \Delta\lambda^i, q^i, x^i) + h_{n+1}(r_2 + \Delta q^i, \lambda^i, q^i + \Delta q^i, x^i) \\ & \geq \tilde{V}_{n+1}^i(\lambda^i + \Delta\lambda^i; q^i, x^i) + \tilde{V}_{n+1}^i(\lambda; q^i + \Delta q^i, x^i), \end{aligned}$$

where  $h_n(r, \lambda^i, s^i)$  denotes the cost of taking action  $r$  when the LM equals  $\lambda^i$  in state  $s^i$ .

Supermodularity of the value function implies monotonicity of optimal action  $r^i(s^i)$  in the parameter  $\lambda$  by Theorem 10.2 of [99].  $\square$

**Lemma D.2.**  $\lambda^i$  is a decreasing function of  $\theta^i$ .

*Proof.* Increase in  $\theta^i$  for a user  $i$  implies that user  $i$  obtains a higher fraction of the time slots, i.e., more opportunities to transmit. This in turn implies that its average queue length would be smaller on an average, thus being above the constraint fewer times on an average. This leads to a lower value of  $\lambda^i$ .  $\square$

**Lemma D.3.**  $\lambda^j$  is an increasing function of  $\theta^i$  for  $j \neq i$ .

*Proof.* Increase in  $\theta^i$  for a user  $i$  implies that user  $i$  obtains a higher fraction of the time slots, i.e., more opportunities to transmit. This in turn implies that user  $j$  obtains a lower fraction of time slots, i.e.,  $\theta^j$  reduces thereby increasing  $\lambda^j$  by Lemma D.2.  $\square$

**Lemma D.4.**  $\lambda^j$  is an increasing function of  $\lambda^i$  for  $j \neq i$ , i.e.,  $\frac{\partial \lambda^j}{\partial \lambda^i} > 0$ .

*Proof.* Increase in  $\lambda^i$  results in an increase in the rate  $r^i$  informed by user  $i$  to the base station, thus leading to an increase in  $\theta^i$  for a user  $i$ . This implies that user  $i$  obtains a higher fraction of the time slots, i.e., more opportunities to transmit. This in turn implies that user  $j$  obtains a lower fraction of time slots, i.e.,  $\theta^j$  reduces thereby increasing  $\lambda^j$  by Lemma D.2. Thus  $\frac{\partial \lambda^j}{\partial \lambda^i} > 0$ .  $\square$

We now argue that if there is sufficient capacity to meet the delay constraints of all the users, the LMs exhibit stabilizing behavior. First note that the  $\{\lambda_n\}$  iterates are bounded since they are constrained to remain in the interval  $[0, \Gamma]$ . Now, the system capacity is finite since each user has a limit on the maximum power with which it can transmit in any time slot. Consider the LM update equation for user  $i$ :

$$\lambda_{n+1}^i = \Lambda[\lambda_n^i + e_n (Q_n^i - \bar{\delta}^i)]. \quad (\text{D.7})$$

For the purposes of analysis, we ignore the projection operator. The projection operation introduces some error, which becomes asymptotically negligible if the LMs converge in the interval  $[0, \Gamma]$ . Hence we analyze the following LM iteration:

$$\lambda_{n+1}^i = \lambda_n^i + e_n (Q_n^i - \bar{\delta}^i). \quad (\text{D.8})$$

Note that (D.8) can be written in the form:

$$\lambda_{n+1}^i = \lambda_n^i + e_n (Q_n^i - \bar{Q}_n^i + \bar{Q}_n^i - \bar{\delta}^i), \quad (\text{D.9})$$

where  $\bar{Q}_n^i$  is the running average of the queue length of user  $i$  upto slot  $n$ . The term  $Q_n^i - \bar{Q}_n^i$  is a zero mean random variable.

Consider a situation where there exists at least one user whose delay constraint is not being satisfied and on an average all the users are not transmitting at maximum power. This means that the system has residual capacity. Since the delay constraint of say a user  $i$  is not being satisfied,  $\bar{Q}_n^i > \bar{\delta}^i$ . This means that the LM  $\lambda^i$  increases. This leads to an

increase in the rate that the user  $i$  informs to the base station in any state  $s$ , by Lemma D.1. Since the base station schedules the user informing the highest rate, the fraction of slots allocated to user  $i$  increases thus increasing  $\theta^i$ . Let  $p^i(s)$  be the probability of user  $i$  being in a state  $s$ . The average throughput achieved by user  $i$  when it is scheduled with a probability  $\theta^i$  can be expressed as:

$$T_{av}^i = \sum_r \sum_s p^i(s) p^i(r|s) \theta^i, \quad (\text{D.10})$$

where and  $p^i(r|s)$  is the probability that user  $i$  informs a rate  $r$  to the base station when it is in state  $s$ . Since the rate  $r(s)$  and  $\theta^i$  increase, there is an increase in the throughput. Note that in order to increase the throughput, the user  $i$  has to transmit at a higher average rate, thus increasing the average power consumption. Moreover, note that an increase in the fraction of slots allocated to a user  $i$  reduces the fraction of slots allocated to user  $j, j \neq i$ , i.e.,  $\theta^j, j \neq i$  reduces. This reduces the throughput of user  $j$ , thus increasing its average queue length  $\bar{Q}^j$ . Thus an increase in  $\lambda^i$  leads to an increase in the average queue length  $\bar{Q}^j, j \neq i$  implying that  $\frac{\partial \bar{Q}^j}{\partial \lambda^i} > 0$ .

When the throughput of user  $j$  reduces, its average queue length becomes larger than the constraint, thus increasing its LM  $\lambda^j$ . This further leads to an increase in the average rate at which user  $j$  transmits, thus increasing the average throughput of user  $j$ . If the average throughput of all the users is such that their delay constraint is satisfied, the LMs stabilize.

From (D.9) and by the o.d.e. method of analyzing the asymptotic behavior of stochastic approximation algorithms [25, 90], it can be easily argued that the LM iterates track the o.d.e.:

$$\dot{\lambda}^i(t) = \bar{Q}^i(t) - \bar{\delta}^i. \quad (\text{D.11})$$

We have already observed that:

$$\frac{\partial \bar{Q}^i}{\partial \lambda^j} > 0, j \neq i. \quad (\text{D.12})$$

Note that the average queue length  $\bar{Q}_n^i$  for user  $i$  is a function of the LMs of all the users. Let  $F^i(\cdot)$  represent that function. Hence (D.11) can be expressed as:

$$\dot{\lambda}^i(t) = F^i(\boldsymbol{\lambda}(t)) - \bar{\delta}^i. \quad (\text{D.13})$$

Moreover, if the system has enough capacity to satisfy the constraints of all the users, the average queue lengths of all the users remain bounded, implying that the LMs remain

bounded. (D.12), (D.13) and boundedness of the LMs allows us to make use of arguments similar to that of the ‘cooperative o.d.e.’ concept from [101] to claim that the  $\lambda$  iterates stabilize.

Increase in transmission power of a user  $i$  leads to an increase in its transmission rate. This leads to a higher fraction of slots being allocated to user  $i$ , leading to a decrease in the fraction of slots allocated to a user  $j$ . If the delay constraint of the user  $j$  is not satisfied under this new allocation of slots, user  $j$  increases its transmission power. Following on similar lines as above, the value function iterates which govern this power adjustment, have a similar stabilizing structure. A more rigorous analytical proof, however, needs to be worked out. This provides the required justification for the convergence of the value function and LM iterates to equilibrium values which in turn implies that the delay constraints of the users are satisfied.

# Bibliography

- [1] V. Gray, "ICT Market Trends," Feb. 2008. [Online] Available: [http://www.itu.int/ITU-D/ict/papers/2008/ITU\\_Gray\\_WT0.pdf](http://www.itu.int/ITU-D/ict/papers/2008/ITU_Gray_WT0.pdf).
- [2] "Cellular Operators Association of India." [Online] Available: [http://www.coai.in/archives\\_statistics\\_2008\\_q1.htm](http://www.coai.in/archives_statistics_2008_q1.htm).
- [3] "Third Generation Partnership Project." [Online] Available: <http://www.3gpp.org/specs/specs.htm>.
- [4] "Third Generation Partnership Project 2." [Online] Available: [http://www.3gpp2.org/Public\\_html/specs/index.cfm](http://www.3gpp2.org/Public_html/specs/index.cfm).
- [5] LAN/MAN Committee, *IEEE 802.16-2004: IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems*. IEEE Computer Society, May 2004.
- [6] LAN/MAN Committee, *IEEE 802.11-2007: IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, March 2007.
- [7] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann, 2001.
- [8] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems. I. Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, 1997.

- 
- [9] J. Proakis, *Digital Communications*. McGraw-Hill, 4 ed., 1991.
- [10] D. P. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1987.
- [11] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.
- [12] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead," *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, 2005.
- [13] X. Lin, N. B. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," *IEEE Journal in Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.
- [14] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, 1997.
- [15] S. Kunniyur and R. Srikant, "End-to-End Congestion Control: Utility Functions, Random Losses and ECN Marks," in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1323–1332, Mar. 2000.
- [16] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing Quality of Service over a Shared Wireless Link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, 1996.
- [17] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [18] B. Prabhakar, E. U. Biyikoglu, and A. E. Gamal, "Energy-Efficient Transmission over a Wireless Link via Lazy Packet Scheduling," in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 386–394, Apr. 2001.
- [19] E. Biglieri, J. Proakis, and S. Shamai, "Fading Channels: Information-Theoretic and Communications Aspects," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2619–2692, 1998.
- [20] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

- [21] R. Knopp and P. A. Humblet, "Information Capacity and Power Control in Single-Cell Multiuser Communications," in *Proceedings of IEEE ICC*, vol. 1, pp. 331–335, June 1995.
- [22] M. Puterman, *Markov Decision Processes*. Wiley, 1994.
- [23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [24] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [25] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.
- [26] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2003.
- [27] R. A. Berry and R. G. Gallager, "Communication over Fading Channels with Delay Constraints," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.
- [28] M. Agarwal, A. Karandikar, and V. S. Borkar, "Structural Properties of Optimal Transmission Policies over a Randomly Varying Channel," *IEEE Transactions on Automatic Control*, July 2008. To appear. [Online] Available: [http://www.ee.iitb.ac.in/~karandi/pubs\\_dir/mukul.ps.gz](http://www.ee.iitb.ac.in/~karandi/pubs_dir/mukul.ps.gz).
- [29] M. Goyal, A. Kumar, and V. Sharma, "Power Constrained and Delay Optimal Policies for Scheduling Transmissions over a Fading Channel," in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 311–320, Mar. 2003.
- [30] D. V. Djonin and V. Krishnamurthy, "Structural Results on Optimal Transmission Scheduling over Dynamical Fading Channels:: A Constrained Markov Decision Process Approach," *The IMA Volumes in Mathematics and its Applications*, vol. 143, pp. 75–98, 2007.
- [31] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Athena Scientific, 1995.

- [32] A. Goldsmith and P. Varaiya, "Capacity of Fading Channels with Channel Side Information," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1986–1992, 1997.
- [33] D. Tse and S. Hanly, "Multi-access Fading Channels - Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, 1998.
- [34] T. S. Rappaport, *Wireless Communications Principles and Practice*. Prentice Hall, 2002.
- [35] H. S. Wang and N. Moayeri, "Finite-State Markov Channel – A Useful Model for Radio Communication Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.
- [36] H. Wang and N. Mandayam, "A Simple Packet Transmission Scheme for Wireless Data over Fading Channels," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1055–1059, 2004.
- [37] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [38] D. Tse and S. Hanly, "Multi-access Fading Channels - Part II: Delay-Limited Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2816–2831, 1998.
- [39] X. Liu, E. Chong, and N. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE Journal on Selected Areas in Communication*, vol. 19, no. 10, pp. 2053–2064, 2001.
- [40] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [41] D. Tse, "Optimal Power Allocation over Parallel Gaussian Broadcast Channels," in *Proceedings of IEEE ISIT*, p. 27, 1997.
- [42] L. Li and A. Goldsmith, "Capacity and Optimal Resource Allocation for Fading Broadcast Channels: Part I: Ergodic Capacity," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1083–1102, 2001.

- [43] P. Bender, P. Black, M. Grob, P. Padovani, N. Sindhushyana, and A. Viterbi, "CDMA/HDR : A Bandwidth Efficient High Speed Wireless Data Service for Nomadic Users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, 2000.
- [44] V. K. N. Lau and Y.-K. R. Kwok, *Channel-Adaptive Technologies and Cross-Layer Designs for Wireless Systems with Multiple Antennas*. Wiley-Interscience, 2006.
- [45] E. F. Chaponniere, P. Black, J. M. Holtzman, and D. Tse, "Transmitter directed Multiple Receiver System using Path Diversity to Equitably Maximize Throughput," Sept. 2002. U.S. Patent No. 6449490.
- [46] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.
- [47] S. Borst and P. Whiting, "Dynamic Channel-Sensitive Scheduling Algorithms for Wireless Data Throughput Optimization," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 3, pp. 569–586, 2002.
- [48] F. Berggren and R. Jantti, "Asymptotically Fair Transmission Scheduling over Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 326–336, 2004.
- [49] Y. Liu and E. Knightly, "Opportunistic Fair Scheduling over Multiple Wireless Channels," in *Proceedings of IEEE INFOCOM*, vol. 2, pp. 1106–1115, Mar. 2003.
- [50] S. Kulkarni and C. Rosenberg, "Opportunistic Scheduling Policies for Wireless Systems with Short Term Fairness Constraints," in *Proceedings of IEEE GLOBECOM*, vol. 1, pp. 533–537, Dec. 2003.
- [51] E. Yeh, "An Inter-Layer view of Multiaccess Communications," in *Proceedings of IEEE ISIT*, p. 112, 2002.
- [52] M. J. Neely, E. Modiano, and C. Rohrs, "Power and Server Allocation in Multi-beam Satellite with Time Varying Channels," in *Proceedings of IEEE INFOCOM*, pp. 138–152, 2002.

- [53] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity," *IEEE Transactions on Information Theory*, no. 2, pp. 466–478, 1993.
- [54] S. Shakkottai and A. Stolyar, "Scheduling for Multiple Flows Sharing a Time-varying Channel: The Exponential Rule," *AMS Translations Series 2*, vol. 207, pp. 185–202, 2002.
- [55] E. Yeh and A. Cohen, "Delay Optimal Rate Allocation in Multiaccess Fading Communications," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pp. 404–407, Oct. 2002.
- [56] R. Berry and E. M. Yeh, "Cross-Layer Wireless Resource Allocation," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 59–68, 2004.
- [57] C. A. C. Coello, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, vol. 3, no. 1, pp. 269–308, 1999.
- [58] Y. Sawargi, *Theory of Multiobjective Optimization*. Academic Press, 1985.
- [59] E. Altman, *Constrained Markov Decision Processes*. Chapman and Hall/CRC Press, 1999.
- [60] R. Berry, "Power and Delay Trade-offs in Fading Channels," June 2000. PhD Thesis, Massachusetts Institute of Technology.
- [61] D. Rajan, A. Sabharwal, and B. Aazhang, "Delay-Bounded Packet Scheduling of Bursty Traffic over Wireless Channels," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 125–144, 2004.
- [62] G. Rajadhyaksha and V. S. Borkar, "Transmission Rate Control over Randomly Varying Channels," *Probability in the Engineering and Informational Sciences*, vol. 19, no. 1, pp. 73–82, 2005.
- [63] D. Farias and B. V. Roy, "The Linear Programming Approach to Approximate Dynamic Programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

- [64] M. J. Neely, "Optimal Energy and Delay Tradeoffs for Multi-user Wireless Downlinks," in *Proceedings of IEEE INFOCOM*, pp. 1–13, 2006.
- [65] J. Schiller, *Mobile Communications*. Addison-Wesley, 2000.
- [66] N. Abramson, "THE ALOHA SYSTEM - Another Alternative for Computer Communications," in *Proceedings of 1970 Fall Joint Computer Conference*, vol. 37, pp. 281–285, 1970.
- [67] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1995.
- [68] E. Altman, R. El-Azouzi, and T. Jimenez, "Slotted Aloha as a Game with Partial Information," *Computer Networks*, no. 45, pp. 701–713, 2004.
- [69] A. B. MacKenzie and S. B. Wicker, "Selfish Users in Aloha: A Game-Theoretic Approach," in *Proceedings of IEEE VTC, Fall*, vol. 3, pp. 1354–1357, Oct. 2001.
- [70] V. S. Borkar and A. A. Kherani, "Random Access in Wireless Ad Hoc Networks as a Distributed Game," in *Proceedings of WiOpt*, Mar. 2004.
- [71] E. Altman, D. Barman, R. El-Azouzi, and T. Jimenez, "A Game Theoretic Approach for Delay Minimization in Slotted Aloha," in *Proceedings of IEEE ICC*, vol. 7, pp. 3999–4003, June 2004.
- [72] S. Lu, V. Bharghavan, and R. Srikant, "Optimal Power and Retransmission Control Policies for Random Access Systems," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 1156–1166, 2004.
- [73] A. B. MacKenzie and S. B. Wicker, "Game Theory in Communications: Motivation, Explanation, and Application to Power Control," in *Proceedings of IEEE GLOBECOM*, vol. 2, pp. 821–826, Nov. 2001.
- [74] T. Heikkinen, "A Potential Game Approach to Distributed Power Control and Scheduling," *Computer Networks*, vol. 50, no. 13, pp. 2295–2311, 2006.
- [75] E. Altman, T. Boulogne, R. El-Azouzi, T. Jimenez, and L. Wynter, "A Survey on Networking Games in Telecommunications," *Computers and Operations Research*, vol. 33, no. 2, pp. 286–311, 2004.

- [76] A. B. MacKenzie and L. A. DaSilva, *Game Theory for Wireless Engineers*. Morgan and Claypool, 2006.
- [77] E. Altman, K. Avrachenkov, N. Bonneau, M. Debbah, R. El-Azouzi, and D. Menasche, “Constrained Stochastic Games in Wireless Networks,” in *Proceedings of IEEE GLOBECOM*, pp. 315–320, Nov. 2007.
- [78] X. Qin and R. A. Berry, “Exploiting Multiuser Diversity for Medium Access Control in Wireless Networks,” in *Proceedings of IEEE INFOCOM*, vol. 2, pp. 1084–1094, Mar. 2003.
- [79] Y. Yu and G. B. Giannakis, “Opportunistic Medium Access for Wireless Networking Adapted to Decentralized CSI,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1445–1455, 2006.
- [80] X. Qin and R. A. Berry, “Opportunistic Splitting Algorithms for Wireless Networks,” in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1662–1672, Mar. 2004.
- [81] S. Adireddy and L. Tong, “Exploiting Decentralized Channel State Information for Medium Access Control in Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 537–561, 2005.
- [82] X. Qin and R. A. Berry, “Distributed Approaches for Exploiting Multiuser Diversity in Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 392–413, 2006.
- [83] D. Zheng, W. Ge, and J. Zhang, “Distributed Opportunistic Scheduling for Ad-Hoc Communications: an Optimal Stopping Approach,” in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Sept. 2007.
- [84] V. S. Borkar, “Convex Analytic Methods in Markov Decision Processes,” in *E. A. Feinberg, A. Schwartz (Eds.) Handbook of Markov Decision Processes*, pp. 347–375, Kluwer Academic Publishers, 2001.
- [85] V. F. Magirou, “Stockpiling under Price Uncertainty and Storage Capacity Constraints,” *European Journal of Operational Research*, no. 3, pp. 233–246, 1982.

- [86] B. V. Roy, D. P. Bertsekas, Y. Lee, and J. N. Tsitsiklis, "A Neuro-Dynamic Programming Approach to Retailer Inventory Management," in *Proceedings of IEEE CDC*, vol. 4, pp. 4052–4057, 1997.
- [87] V. S. Borkar, "On the Lock-in Probability of Stochastic Approximation," *Combinatorics, Probability and Computing*, vol. 11, no. 1, pp. 11–20, 2002.
- [88] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning Algorithms for Markov Decision Processes with Average Cost," *SIAM Journal on Control and Optimization*, vol. 40, pp. 681–698, 2001.
- [89] V. S. Borkar and K. Soumyanath, "A New Analog Parallel Scheme for Fixed Point Computation, part 1: Theory," *IEEE Transactions on Circuits and Systems I*, vol. 44, pp. 351–355, 1997.
- [90] V. S. Borkar and S. P. Meyn, "The ODE Method for Convergence of Stochastic Approximation and Reinforcement Learning," *SIAM Journal on Control and Optimization*, vol. 38, pp. 447–469, 2000.
- [91] V. S. Borkar, "Stochastic Approximation with Two Time Scales," *Systems and Control Letters*, vol. 29, pp. 291–294, 1996.
- [92] V. S. Borkar, "An Actor-Critic Algorithm for Constrained Markov Decision Processes," *Systems and Control Letters*, vol. 54, pp. 207–213, 2005.
- [93] A. Bhorkar, A. Karandikar, and V. S. Borkar, "Power Optimal Opportunistic Scheduling," in *Proceedings of IEEE GLOBECOM*, pp. 1–5, Nov. 2006.
- [94] E. Yeh, "Multiaccess and Fading in Communication Networks," Sept. 2001. PhD Thesis, Massachusetts Institute of Technology.
- [95] H. P. Young, *Strategic Learning and its Limits*. Oxford University Press, 2004.
- [96] H. P. Young, "The Possible and the Impossible in Multi-Agent Learning," *Artificial Intelligence*, vol. 171, no. 7, pp. 429–433, 2007.
- [97] C. Cicconeti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network*, vol. 20, no. 2, pp. 50–55, 2006.

- 
- [98] T. G. Neame, M. Zukerman, and R. G. Addie, “Modeling Broadband Traffic Streams,” in *Proceedings of IEEE GLOBECOM*, pp. 1048–1052, Dec. 1999.
- [99] R. K. Sundaram, *A First Course in Optimization Theory*. Cambridge University Press, 1996.
- [100] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*. Springer-Verlag, 1989.
- [101] M. W. Hirsch, “Systems of Differential Equations that are Competitive or Cooperative II: Convergence almost Everywhere,” *SIAM Journal on Mathematical Analysis*, vol. 16, pp. 423–439, 1985.
- [102] H. L. Smith, *Monotone Dynamical Systems*. American Mathematical Society, 1995.
- [103] A. Mas-Collel, M. Whinston, and J. Green, *Microeconomic Theory*. Oxford University Press, 1995.
- [104] J. P. Monks, V. Bharghavan, and W. Hwu, “A Power Controlled Multiple Access Protocol for Wireless Packet Networks,” in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 219–228, Apr. 2001.
- [105] B. Bharath and V. S. Borkar, “Stochastic Approximation Algorithms: Overview and Recent Trends,” *Sadhana*, vol. 24, no. 4, 5, pp. 425–452, 1999.